Independent Doubly Adaptive Rejection Metropolis Sampling Within Gibbs Sampling

Luca Martino, Jesse Read, and David Luengo

Abstract*—Bayesian methods have become very popular in signal processing lately, even though performing exact Bayesian inference is often unfeasible due to the lack of analytical expressions for optimal Bayesian estimators. In order to overcome this problem, Monte Carlo (MC) techniques are frequently used. Several classes of MC schemes have been developed, including Markov Chain Monte Carlo (MCMC) methods, particle filters and population Monte Carlo approaches. In this paper, we concentrate on the Gibbs-type approach, where automatic and fast samplers are needed to draw from univariate (full-conditional) densities. The Adaptive Rejection Metropolis Sampling (ARMS) technique is widely used within Gibbs sampling, but suffers from an important drawback: an incomplete adaptation of the proposal in some cases. In this work, we propose an alternative adaptive MCMC algorithm ($\mathbf{IA^2RMS}$) that overcomes this limitation, speeding up the convergence of the chain to the target, allowing us to simplify the construction of the sequence of proposals, and thus reducing the computational cost of the entire algorithm. Note that, although $\mathbf{IA^2RMS}$ has been developed as an extremely efficient MCMC-within-Gibbs sampler, it also provides an excellent performance as a stand-alone algorithm when sampling from univariate distributions. In this case, the convergence of the proposal to the target is proved and a bound on the complexity of the proposal is provided. Numerical results, both for univariate ($stand-alone\ IA^2RMS$) and multivariate ($IA^2RMS-within-Gibbs$) distributions, show that $\mathbf{IA^2RMS}$ outperforms ARMS and other classical techniques, providing a correlation among samples close to zero.

***Index Terms*—Adaptive MCMC, adaptive rejection Metropolis sampling, bayesian inference, Gibbs sampler, metropolis-hastings within Gibbs sampling, monte carlo methods.**

Manuscript received March 23, 2014; revised October 16, 2014 and March 18, 2015; accepted March 27, 2015. Date of publication April 06, 2015; date of current version May 13, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ruixin Niu. This work was supported by the Spanish Government through projects COMONSENS (CSD2008-00010), ALCIT (TEC2012-38800-C03-01), COMPREHENSION (TEC2012-38883-C02-01), and DISSECT (TEC2012-38058-C03-01). The work of L. Martino was supported by the ERC under Grant 239784 and by the AoF under Grant 251170. The work of D. Luengo was supported by the BBVA Foundation through project MG-FIAR ("I Convocatoria de Ayudas Fundación BBVA a Investigadores, Innovadores y Creadores Culturales").

L. Martino is with the Department of Mathematics and Statistics, University of Helsinki, Helsinki 00014, Finland (e-mail: lukatotal@gmail.com).

J. Read is with the Department of Computer Science, Aalto University, Espoo 02150, Finland (e-mail: jesse.read@gmail.com).

D. Luengo is with the Department of Signal Theory and Communications, Universidad Politécnica de Madrid, Madrid 28040, Spain (e-mail: david.luengo@upm.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2420537

I. Introduction

BAYESIAN methods and their implementations by means of sophisticated Monte Carlo (MC) techniques [1], [2] have become very popular in signal processing over the last two decades [3]–[7]. Indeed, in many problems of practical interest these techniques demand procedures for sampling from probability distributions with non-standard forms, such as Markov chain Monte Carlo (MCMC) methods [4], [8], particle filters [6], [9] or population Monte Carlo approaches [10]. In particular, MCMC techniques generate samples from a target probability density function (pdf) by drawing from a simpler proposal pdf [1], [11]. The two best known MCMC approaches are the Metropolis-Hastings (MH) and the Gibbs sampling algorithms [2]. The Gibbs sampler is extensively used in signal processing [3], [7] and machine learning [12], [13] to generate samples from multi-dimensional target densities, drawing each component from the corresponding univariate full-conditional density.

The key point for the successful application of Gibbs sampling is being able to draw efficiently from these univariate pdfs. Otherwise, generic universal sampling techniques, like rejection sampling (RS) or MH-type algorithms, are used *within* the Gibbs sampler to draw from complicated full-conditionals. In the first case, samples generated from the RS algorithm are independent, but the acceptance rate can be very low. In order to increase the acceptance rate, automatic and self-tuning samplers, such as *adaptive rejection sampling* (ARS) [14], [15], have been developed to draw efficiently from univariate target densities. The samples generated by ARS are independent and the proposal always converges to the true shape of the target, but ARS can only be applied for log-concave (i.e., unimodal) targets. Several generalizations of ARS have been proposed [16]–[18], but they are still only able to handle very specific classes of pdfs. In the second case, we have an MCMC-inside-another-MCMC approach. In the so called *MH-within-Gibbs* approach, *only* one MH step is often performed within each Gibbs iteration [2], [11]. This *hybrid* approach preserves the ergodicity of the Gibbs sampler [2], [19], and achieves a satisfactory performance in many cases. For this reason, it has been extensively used in signal processing applications [20]–[22]. On the other hand, several authors have noticed that using a single MH step for the *internal* MCMC is not always the best solution (cf. [23]).

Fig. 1 provides a graphical illustration of the previous discussion. The best scenario occurs when efficient and direct samplers for each of the full-conditionals are available, whereas the worst scenario corresponds to the highly correlated samples (as

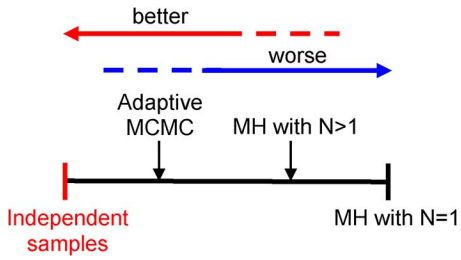

Fig. 1. Performance of a Gibbs sampler depending on the technique used for drawing from the full-conditionals.

typically generated by the MH algorithm using only $N = 1$ steps per Gibbs iteration unless the proposal is specifically tailored to the target).[1] Hence, we depict "MH with $N > 1$" to the right of "Adaptive MCMC".[2] In the middle, we have the adaptive MCMC algorithms and the MH algorithm with $N > 1$ steps per Gibbs iteration. On the one hand, note that there are several types of adaptive algorithms with different levels of performance. The better the adaptation provided by an adaptive MCMC algorithm (in terms of converging to the target), the more independent the samples generated, and thus the better its performance and the closer to the left (i.e., the ideal situation). On the other hand, using of a larger value of N for the MH algorithm, there is more probability of avoiding the "burn-in" period so that the last sample be distributed as the full-conditional. Thus, the case $N > 1$ is closer to the ideal situation. However, unless the proposal is very well tailored to the target, a properly designed adaptive MCMC algorithm should provide less correlated samples than a standard MH algorithm.

Therefore, some authors have increased the number of MH transitions ($N > 1$) per iteration of the Gibbs sampler [25]–[27], whereas other authors have suggested alternative approaches for sampling from the full conditionals [28]–[32]. Indeed, this was one of the main motivations for the development of the *Adaptive Rejection Metropolis Sampling* (ARMS) algorithm by Gilks *et al.* [28], and is also the basis for our own work. ARMS combines the ARS and MH approaches [28], [33] in order to obtain a universal sampler that builds a self-tuning proposal (i.e., a proposal automatically constructed and adapted to the target). Unfortunately, the samples generated by ARMS are correlated (unlike ARS, which returns i.i.d. samples). However, due to its robustness and good performance, it has often been preferred to other MCMC techniques within Gibbs and it has been applied in many inference problems (cf. [23], [34], [35]).

[1]Note that Fig. 1 does not take into account the computational cost. If an exact sampler requires a large increase in computation time, then an approximate solution may be preferable. However, an exact or asymptotically exact sampler should always be preferred to an approximate one when the price to be paid is just a moderate increase in the computational cost.

[2]Remarkably, some authors have found that a "bad" choice of the proposal function in the MH step (i.e., a poor approximation of the full conditional) can improve the performance of the Gibbs sampler [2], [24]. This is possibly due to the fact that the acceptance rate in the MH step (lower than 1) induces an "accidental" random scan of the components of the target pdf in the Gibbs sampler, which can improve the performance in some cases. In this work, we only focus on the deterministic scan. However, all the *internal* techniques discussed later can also be applied within a random scan Gibbs scheme.

In this paper, we show that the adaptation mechanism of ARMS can be incomplete: in some cases the proposal does not converge to the target in certain areas or even in the whole domain. This is an important drawback, as the correlation among the samples generated depends on the discrepancy between the proposal and the target pdfs [2], [36]. In order to solve this issue, we present an enhancement of ARMS which ensures that the sequence of proposals converges to the target (as in ARS and unlike in ARMS), while maintaining the computational cost bounded (exactly as in ARS and ARMS) with the addition of a simple control test. This improvement yields a substantial reduction of the correlation (providing asymptotically independent samples) and a lower mean squared error (MSE) in the estimations. We call the novel approach *independent* A^2RMS (IA^2RMS), since the proposal is independent from the current state, and the A^2 emphasizes the fact that we incorporate an additional adaptive control. Furthermore, the new strategy allows us to decouple completely the adaptation mechanism from the proposal construction (unlike ARMS, whose performance depends critically on the proposal building approach), thus allowing us to introduce several examples of simpler construction procedures. Hence, the resulting algorithm is faster and more efficient than standard ARMS and other techniques, as shown in the numerical simulations.[3] Finally, we would like to remark that, although we have concentrated on the use of IA^2RMS as a stand-alone algorithm and within Gibbs sampling, it can also be directly applied within any other MC algorithm that requires sampling from conditional distributions, such as the hit-and-run algorithm [38] or adaptive direction sampling [39].

The paper is structured as follows. First of all, Section II introduces the notation and provides some background on Gibbs sampling, ARS and ARMS. The important structural limitation of ARMS is also discussed at the end of this section. Then, Section III describes the novel algorithm, whereas Section IV provides numerical results, including the use of IA^2RMS as a stand-alone algorithm and within a Gibbs sampler. Section V shows the conclusions and future lines. Finally, several important theoretical results are proved in the Appendices.

II. Problem Statement and Background

A. Gibbs Sampling

Bayesian inference often requires drawing samples from complicated multivariate posterior pdfs, $p(\mathbf{x}|\mathbf{y})$ with $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$. Typical examples in signal processing include model selection, blind equalization and source separation, or spectral analysis [4], [7]. A common approach, when direct sampling from $p(\mathbf{x}|\mathbf{y})$ is unfeasible, is using a Gibbs sampler [2]. At the i-th iteration, a Gibbs sampler obtains the d-th component

[3]A preliminary version of this work has been published in [37]. With respect to that paper, here we discuss the structural limitations of ARMS, we perform a complete theoretical analysis of the IA^2RMS technique, and we provide additional numerical simulations (including comparisons with other sampling algorithms, drawing samples from a heavy-tailed distribution, and three examples of the application of IA^2RMS within a Gibbs sampler).

TABLE I
MAIN NOTATION OF THE WORK

$p_o(x) = \frac{1}{c_p} p(x)$	Target density.
$\widetilde{\pi}_t(x) = \frac{1}{c_{\pi_t}} \pi_t(x)$	Adaptive proposal density.
N	Total number of (independent or correlated) samples generated by the different algorithms.
T	Total number of iterations of the method; for all the techniques in this work we have $T \geq N$.
$\mathcal{S}_t$	Set of support points at the t-th iteration.
m_t	Number of points in $\mathcal{S}_t$, i.e., $m_t = \lvert\mathcal{S}_t\rvert$.

$(d = 1, \ldots, D)$ of $\mathbf{x}$, x_d, drawing from the full conditional of x_d given all the information available [7], [40], i.e.,

$$x_d^{(i)} \sim p\left(x_d | \mathbf{x}_{1:d-1}^{(i)}, \mathbf{x}_{d+1:D}^{(i-1)}\right), \tag{1}$$

with the initial vector drawn from the prior, i.e., $\mathbf{x}^{(0)} \sim q(\mathbf{x})$, and $i = 1, \ldots, N_G$ (where N_G is the total number of Gibbs iterations). However, even sampling from (1) can often be complicated. In these cases, a common approach is using another Monte Carlo technique (e.g., RS or the MH algorithm) within the Gibbs sampler, drawing candidates from a simpler proposal, $\pi(x)$, and accepting or discarding them according to some appropriate rule. The performance of this approach depends largely on the choice of $\pi(x)$. Thus, adaptive strategies, where the proposal is updated using the previously generated samples, are usually preferred.

For the sake of simplicity, let us denote the univariate target pdf (i.e., the full-conditional proposal in (1)) as $p_o(x) = \frac{1}{c_p} p(x)$ for $x \in \mathcal{X} \subseteq \mathbb{R}$, where $p(x)$ is an unnormalized but proper (i.e., integrable) function, and $c_p = \int_{\mathcal{X}} p(x)dx$ is the normalizing constant. Similarly, the sequence of proposals constructed by the adaptive algorithm is denoted as $\widetilde{\pi}_t(x) = \frac{1}{c_{\pi_t}} \pi_t(x)$ for $t = 1, \ldots, T$ (T is the number of iterations of the adaptive algorithm), where $\pi_t(x)$ is again an unnormalized but proper function, and $c_{\pi_t} = \int_{\mathcal{X}} \pi_t(x)dx$ is the normalizing constant. Furthermore, we often refer to the unnormalized functions $p(x)$ and $\pi_t(x)$ (and in general to all unnormalized but proper pdfs) as densities. Table I summarizes the main notation of this work. In the following, we review two popular approaches to build $\pi_t(x)$: ARS and ARMS.

B. Adaptive Rejection Sampling (ARS)

If a direct sampling method is not available to draw from a full-conditional pdf, the best alternative is using the *adaptive rejection sampling* (ARS) technique [14], [15]. Let us consider a set of *support points* at the t-th iteration,

$$\mathcal{S}_t = \{s_1, s_2, \ldots, s_{m_t}\} \subset \mathcal{X},$$

such that $s_1 < \ldots < s_{m_t}$, and let us define $V(x) = \log p(x)$ and $w_i(x)$ as the tangent line to $V(x)$ at s_i for $i = 1, \ldots, m_t$. Then we can build a piecewise linear (PWL) function,

$$W_t(x) = \min\{w_1(x), \ldots, w_{m_t}(x)\}, \quad x \in \mathcal{X}. \tag{2}$$

Hence, the proposal pdf, $\widetilde{\pi}_t(x) \propto \pi_t(x) = \exp(W_t(x))$, is formed by exponential pieces in such a way that $W_t(x) \geq V(x)$ (and thus $\pi_t(x) \geq p(x)$) when $V(x)$ is concave (i.e., $p(x)$ is log-concave).

TABLE II
ADAPTIVE REJECTION SAMPLING (ARS) ALGORITHM

Initialization:
1. Set $t = 0$ and $n = 0$. Choose an initial set $\mathcal{S}_0 = \{s_1, \ldots, s_{m_0}\}$.

Iterations (while $n < N$):
2. Build a proposal, $\pi_t(x)$, given a set of support points $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$, according to Eq. (2).
3. Draw $x' \sim \widetilde{\pi}_t(x) \propto \pi_t(x)$ and $u' \sim \mathcal{U}([0, 1])$.
4. If $u' > \frac{p(x')}{\pi_t(x')}$, then reject x', update $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{x'\}$, $m_{t+1} = m_t + 1$ and set $t = t + 1$. Go back to step 2.
5. Otherwise, if $u' \leq \frac{p(x')}{\pi_t(x')}$, then accept x', setting $x_n = x'$.
6. Set $\mathcal{S}_{t+1} = \mathcal{S}_t$, $m_{t+1} = m_t$, $t = t + 1$, $n = n + 1$ and return to step 2.

Table II summarizes the ARS algorithm. Note that a new sample is added to the support set whenever it is rejected in the RS test. ARS has the important property that the sequence of proposals always converges to the target pdf. Defining the L_1 distance between $\pi_t(x)$ and $p(x)$ as

$$D(\pi_t, p) = \|\pi_t(x) - p(x)\|_1 = \int_{\mathcal{X}} |\pi_t(x) - p(x)| \, dx, \tag{3}$$

ARS ensures that $D(\pi_t, p) \to 0$ when $t \to \infty$. This yields two important consequences:

1) The acceptance rate,
$$\eta_t = \int \frac{p(x)}{\pi_t(x)} \widetilde{\pi}_t(x)dx = \frac{c_p}{c_{\pi_t}}, \tag{4}$$
tends to one as $t \to \infty$. Indeed, typically $\eta_t \to 1$ very quickly and ARS becomes virtually an exact sampler after a few iterations.
2) The computational cost remains bounded, as the probability of adding a new support point, $P_t = 1 - \eta_t = \frac{1}{c_{\pi_t}} D(\pi_t, p)$, tends to zero as $t \to \infty$.

C. Adaptive Rejection Metropolis Sampling (ARMS)

Unfortunately, ARS can only be applied to log-concave target pdfs (i.e., when $V(x) = \log p(x)$ is concave). Although several generalizations of ARS have been proposed (cf. [16]–[18]), they are still only able to handle very specific classes of pdfs. An alternative option is provided by the adaptive rejection Metropolis sampling (ARMS) technique, which combines the ARS method and the Metropolis-Hastings (MH) algorithm [1], [2]. ARMS is summarized in Table III. It performs first an RS test, and the rejected samples are used to improve the proposal pdf, exactly as in ARS. However, unlike ARS, the samples accepted in the RS test go through an MH test. The MH step removes the main limitation of ARS: requiring that $\pi_t(x) \geq p(x) \forall x \in \mathcal{X}$. This allows ARMS to generate samples from a wide variety of target pdfs, becoming virtually a universal sampler from a theoretical point of view.

The choice of the proposal construction approach is critical for the good performance of ARMS [28], as discussed in Section II-D. Consider again the set of support points $\mathcal{S}_t = \{s_1, s_2, \ldots, s_{m_t}\}$, and let us define the intervals $\mathcal{I}_0 = (-\infty, s_1]$, $\mathcal{I}_j = (s_j, s_{j+1}]$ for $j = 1, \ldots, m_t - 1$, and

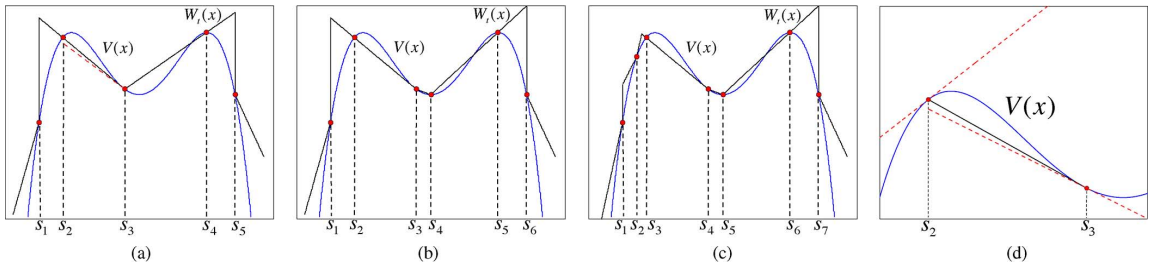

Fig. 2. Example of a critical structural limitation in the adaptive procedure of ARMS. **(a)** Construction of $W_t(x)$ using 5 support points, such that $W_t(x) < V(x)$ inside $\mathcal{I}_2 = (s_2, s_3]$. **(b)-(c)** Adding new support points inside the contiguous intervals the construction of $W_t(x)$ inside $\mathcal{I}_2$ ($\mathcal{I}_3$ in Figure (c)) can never change. **(d)** The secant line $L_{2,3}(x)$ passing through $(s_2, V(s_2))$ and $(s_3, V(s_3))$ (solid black line), and the two tangent lines to $V(x)$ at s_2 and s_3 (dashed red lines), respectively.

TABLE III
ADAPTIVE REJECTION METROPOLIS SAMPLING (ARMS) ALGORITHM

Initialization:

1. Set $n = 0$ (chain's iteration) and $t = 0$ (algorithm's iteration). Choose an initial state x_0 and support set $\mathcal{S}_0 = \{s_1, \ldots, s_{m_0}\}$.

Iterations (while $n < N$):

2. Build a proposal, $\pi_t(x)$, given a set of support points $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$, according to Eq. (5).
3. Draw $x' \sim \widetilde{\pi}_t(x) \propto \pi_t(x)$ and $u' \sim \mathcal{U}([0,1])$.
4. If $u' > \frac{p(x')}{\pi_t(x')}$, then reject x', update $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{x'\}$, $m_{t+1} = m_t + 1$ and set $t = t + 1$. Go back to step 2.
5. Otherwise, draw $u'' \sim \mathcal{U}([0,1])$. If $u'' \leq \alpha$, with
$$\alpha = \min\left[1, \frac{p(x')\min[p(x_n), \pi_t(x_n)]}{p(x_n)\min[p(x'), \pi_t(x')]}\right],$$
then accept x', setting $x_{n+1} = x'$. Otherwise, if $u'' > \alpha$, then reject x', setting $x_{n+1} = x_n$.
6. Set $\mathcal{S}_{t+1} = \mathcal{S}_t$, $m_{t+1} = m_t$, $t = t+1$, $n = n+1$ and return to step 2.

$\mathcal{I}_{m_t} = (s_{m_t}, +\infty)$. Moreover, let us denote as $L_{j,j+1}(x)$ the line passing through $(s_j, V(s_j))$ and $(s_{j+1}, V(s_{j+1}))$ for $j = 1, \ldots, m_t - 1$. Then, a PWL function $W_t(x)$ is constructed in ARMS as

$$W_t(x) = \begin{cases} L_{1,2}(x), & x \in \mathcal{I}_0, \\ \max\{L_{1,2}(x), L_{2,3}(x)\}, & x \in \mathcal{I}_1, \\ \varphi_j(x), & x \in \mathcal{I}_j, \\ \max\{L_{m_t-2,m_t-1}, L_{m_t-1,m_t}(x)\}, & x \in \mathcal{I}_{m_t-1}, \\ L_{m_t-1,m_t}(x), & x \in \mathcal{I}_{m_t}, \end{cases} \quad (5)$$

where

$$\varphi_j(x) = \max\{L_{j,j+1}(x), \min\{L_{j-1,j}(x), L_{j+1,j+2}(x)\}\},$$

and $j = 2, \ldots, m_t - 1$. Hence, the proposal pdf, $\widetilde{\pi}_t(x) \propto \pi_t(x) = \exp(W_t(x))$, is again formed by exponential pieces, as illustrated in Fig. 2.

It is important to remark that the number of linear pieces that form the proposal with this construction is larger than m_t in general, since the proposal can be formed by two segments rather than one in some intervals (e.g., interval $\mathcal{I}_2 = (s_2, s_3]$ in Fig. 2(c)). The computation of intersection points among these two segments is also needed. More sophisticated approaches to build $W_t(x)$ (e.g., using quadratic segments when possible [30]) have been proposed. However, none of them solves the structural problem of ARMS that is briefly described next.

D. Structural Limitations of ARMS

Unlike ARS, the ARMS algorithm cannot guarantee the convergence of the sequence of proposals to the target, i.e., $D(\pi_t, p) \nrightarrow 0$ as $t \to \infty$ in general. In ARMS, the proposal pdf is updated only when a sample x' is rejected by the RS test, something that can only happen when $\pi_t(x') > p(x')$. On the other hand, when a sample is initially accepted by the RS test, as it always happens when $\pi_t(x') \leq p(x')$, the proposal is never updated. Thus, the satisfactory performance of ARMS depends on two issues:

a) $W_{t+1}(x)$ should be constructed in such a way that $W_t(x) \geq V(x)$ (i.e., $\pi_t(x) \geq p(x)$) inside most of the domain of $\mathcal{X}$, so that support points can be added almost everywhere.
b) The addition of a support point inside an interval must entail a change of the proposal pdf inside other neighbouring intervals when building $W_{t+1}(x)$. This allows the proposal to improve inside regions where $\pi_t(x) < p(x)$.

These two conditions lead to unnecessarily complex proposal construction schemes. Furthermore, even if the proposal building approach fulfills these two requirements (as it happens for the procedure proposed in [28] and described by (5)), the convergence of $\pi_t(x)$ to $p(x)$ almost everywhere cannot be guaranteed, due to the fact that support points can never be added inside regions where $\pi_t(x) < p(x)$. Indeed, inside some region $\mathcal{C} \subset \mathcal{X}$, where $\pi_t(x) < p(x)$, we might obtain a sequence of proposals s. t. $\pi_{t+\tau}(x) = \pi_t(x)$ for an arbitrarily large value of τ, or even $\forall \tau \in \mathbb{N}$, i.e., the proposal pdf might never change inside $\mathcal{C} \subset \mathcal{X}$.

This limitation of the ARMS adaptation scheme can be illustrated with a simple graphical example. Consider a multi-modal target density, $p(x) = \exp(V(x))$, with $V(x)$ as shown in Fig. 2(a). Building $W_t(x)$ using 5 support points and the procedure described by (5), we obtain $W_t(x) < V(x) \forall x \in \mathcal{I}_2 = (s_2, s_3]$, as shown in Fig. 2(a). From (5), we note that $W_t(x) = \max\{L_{2,3}(x), \min\{L_{1,2}(x), L_{3,4}(x)\}\}$ inside this interval. From Fig. 2(a), we see that $\min\{L_{1,2}(x), L_{3,4}(x)\} = L_{3,4}(x)$ and $\max\{L_{2,3}(x), L_{3,4}(x)\} = L_{2,3}(x) \forall x \in \mathcal{I}_2 = (s_2, s_3]$. Therefore, $W_t(x) = L_{2,3}(x)$ inside this interval, and this situation can never change when new support points are added. Figs. 2(b) and 2(c) show that we can incorporate new support points, s_4 in Fig. 2(b) and s_2 in Fig. 2(c), arbitrarily close to this interval ($\mathcal{I}_2$ in Fig. 2(a), (b) and $\mathcal{I}_3$ in Fig. 2(c)), without

altering the construction of $W_t(x)$. Indeed, let us consider the limit case where two points are incorporated arbitrarily close to s_2 and s_3. In this extreme situation, the secant lines of the adjacent intervals become tangent lines, as shown in Fig. 2(d), and the minimum between the two tangent lines corresponds to the straight line tangent to s_3, which stays always below the secant line, $L_{2,3}(x)$, passing through $(s_2, V(s_2))$ and $(s_3, V(s_3))$. Hence, since adding support points in adjacent intervals does not improve the construction inside this interval, and support points can never be added directly inside it (as $\pi_t(x) < p(x)$), a very relevant portion of the proposal (containing a mode in this case) can never be updated, meaning that $D(\pi_t, p) \nrightarrow 0$ as $t \to \infty$.

III. Independent Doubly Adaptive Rejection Metropolis Sampling

A. Algorithm Description

Our aim is designing a sequence of self-tuning proposals such that $\pi_t(x) \to p(x)$ when $t \to \infty$ as fast as possible. Namely, we want to obtain an algorithm having a performance as close as possible to the ARS technique (i.e., ensuring that $D(\pi_t, p) \to 0$ as $t \to \infty$ with a bounded computational cost), and the same range of applicability as the ARMS method (i.e., being a universal sampler, able to draw samples virtually from any target pdf). In this section, we describe a simple and extremely efficient strategy which allows us to achieve these two goals. The novel scheme ensures the convergence of the chain to the target distribution and keeps, at the same time, the computational cost bounded. Furthermore, it allows us to completely decouple the adaptation mechanism from the proposal construction, thus allowing us to consider simpler alternatives for the latter, as shown in Section III-C.

The newly proposed algorithm is called *independent doubly adaptive rejection Metropolis sampling* (IA^2RMS), with the A^2 emphasizing that we incorporate an additional adaptive step to improve the proposal pdf w.r.t. ARMS. The IA^2RMS algorithm is summarized in Table IV. Initially, IA^2RMS proceeds like ARMS, drawing a sample from the current proposal (step 3), performing an RS test and incorporating rejected samples to the support set (step 4). Then, initially accepted samples go through an MH step to determine whether they are finally accepted or not (step 5.1), as in ARMS. The key improvement w.r.t. ARMS is the introduction of a new control (step 5.2), which allows us to add samples (in a controlled way) inside regions of the domain where $\pi_t(x) < p(x)$. Therefore, the IA^2RMS algorithm guarantees a complete adaptation of the proposal (i.e., $D(\pi_t, p) \to 0$ as $t \to \infty$) exactly as in ARS (as shown in Appendix B).[4] As a consequence, the correlation among samples is drastically reduced, quickly vanishing to zero, and IA^2RMS becomes an exact sampler after some iterations (like ARS and unlike ARMS), as shown by the numerical results in Section IV.

Finally, let us remark that IA^2RMS requires selecting a single set of parameters: the initial set of support points, $\mathcal{S}_0$. After this choice, the algorithm proceeds automatically without any further intervention required by the user. Regarding the robustness of IA^2RMS w.r.t. $\mathcal{S}_0$, the only requisite is choosing $m_0 \geq 2$ initial support points where the value of the target is different from zero, i.e., $p(s_i) > 0$ for $i = 1, \ldots, m_0$. Furthermore, if the effective support of the target (i.e., the support containing most of its probability mass) is approximately known, then a good initialization consists of selecting the two points delimiting this support and at least another point inside this support. If the user desires to increase the robustness of IA^2RMS, a grid of initial points can be used. This choice speeds up the convergence of the algorithm, but any random selection within the effective support of the target ensures the convergence of IA^2RMS. The robustness of IA^2RMS w.r.t. this initial set can be seen in the Gaussian mixture and heavy tailed distribution examples (see Sections IV-A and IV-B), where a random initial set is selected.

TABLE IV
Independent Doubly Adaptive Rejection Metropolis Sampling (IA^2RMS) Algorithm

Initialization:

1. Set $n = 0$ (chain's iteration) and $t = 0$ (algorithm's iteration). Choose an initial state x_0 and support set $\mathcal{S}_0 = \{s_1, \ldots, s_{m_0}\}$.

Iterations (while $n < N$):

2. Build a proposal, $\pi_t(x)$, given the set $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$, using a convenient procedure (e.g. the ones described in [28], [30] or the simpler ones proposed in Section III-C).
3. Draw $x' \sim \widetilde{\pi}_t(x) \propto \pi_t(x)$ and $u' \sim \mathcal{U}([0,1])$.
4. If $u' > \frac{p(x')}{\pi_t(x')}$, then reject x', update $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{x'\}$, $m_{t+1} = m_t + 1$, set $t = t + 1$, and go back to step 2.
5. Otherwise, if $u' \leq \frac{p(x')}{\pi_t(x')}$, then:
   5.1 Draw $u'' \sim \mathcal{U}([0,1])$. If $u'' \leq \alpha$, with
   $$\alpha = \min\left[1, \frac{p(x')\min[p(x_n), \pi_t(x_n)]}{p(x_n)\min[p(x'), \pi_t(x')]}\right],$$
   then accept x', setting $x_{n+1} = x'$ and $y = x_n$. Otherwise, if $u'' > \alpha$, then reject x', setting $x_{n+1} = x_n$ and $y = x'$.
   5.2 Draw $u''' \sim \mathcal{U}([0,1])$. If
   $$u_2 > \frac{\pi_t(y)}{p(y)},$$
   then set $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{y\}$ and $m_{t+1} = m_t + 1$. Otherwise, set $\mathcal{S}_{t+1} = \mathcal{S}_t$ and $m_{t+1} = m_t$.
   5.3 Update $t = t + 1$, $n = n + 1$ and return to step 2.

B. Convergence of the Chain and Computational Cost

The new control test is performed using an auxiliary variable, y, which is always different from the new state, x_{n+1}. This construction leads to a proposal, $\pi_t(x)$, which is independent of the current state of the chain, x_n. Hence, the convergence of the Markov chain to the target density is ensured by Theorem 2 in [41] (Theorem 8.2.2 in [11]).[5] In Appendix A we show that the conditions required to apply this theorem are fulfilled as long as $\pi_t(x) \to p(x)$ almost everywhere. This crucial issue is proved in Appendix B and implies also that $\widetilde{\pi}_t(x) \to p_o(x)$ almost everywhere as $t \to \infty$, as proved in Appendix C. The convergence of $\pi_t(x)$ to $p(x)$ almost everywhere also implies that probability of adding new support points tends to zero as $t \to \infty$, as proved in Appendix D, thus keeping the computational cost

[4] We remark again that this cannot be guaranteed by the adaptive structure of ARMS, as discussed in Section II-D.

[5] Note that, even though the IA^2RMS algorithm falls inside the broad category of independent adaptive algorithms, its structure is inspired by [28], not by [41]. Indeed, no RS test is performed in [41] and the construction of the proposals is completely different.

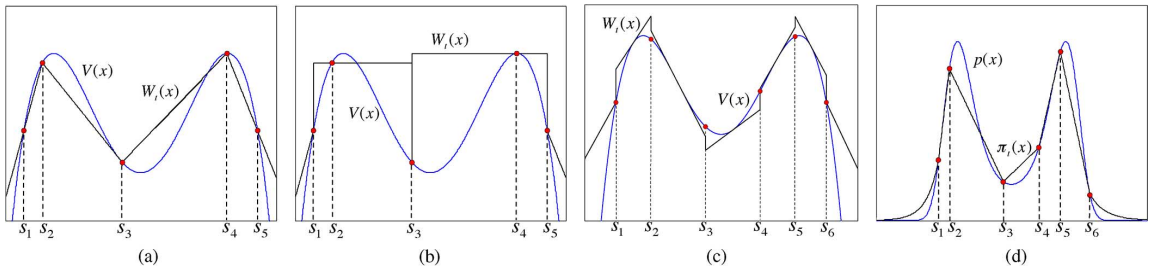

Fig. 3. Examples of proposal constructions using: **(a)** the procedure described by (6) in the log-domain; **(b)** the procedure described by (7) in the log-domain; **(c)** the procedure described by (8) in the log-domain; **(d)** the procedure described by (9) in the pdf's domain.

bounded. Note that there is no contradiction between the results in Appendix A and Appendix D. As more support points are added, $\pi_t(x)$ becomes closer to the target, and this implies a decrease in the probability of adding new support points. However, for $t < \infty$ there is always a non-null (albeit small for large values of t) probability of adding new support points to improve the proposal and make it become closer to the target.

The coding and implementation complexity of IA^{2}RMS is virtually identical to ARMS, since all the quantities involved in the ratio of step 5.2 have been previously calculated in steps 4 and 5.1. Thus, no additional evaluation of the proposal and target pdfs is required. Given a specific construction procedure for $\pi_t(x)$, the total number of support points increases w.r.t. ARMS, but it always remains within the same order of magnitude, as shown in the simulations. Indeed, it is important to emphasize that the number of support points does not diverge: it remains bounded thanks to the two control tests, exactly as in ARS and ARMS, since $\frac{p(x)}{\pi_t(x)} \to 1$ almost everywhere as $t \to \infty$, as shown in the Appendix.

Finally, let us remark that, although most of the results provided in the Appendix are asymptotic, Theorem 2 in [41] also provides us with solid theoretical guarantees for a finite number of iterations of the algorithm. Indeed, it ensures that IA^{2}RMS is drawing samples from the target distribution within a finite number of iterations with a probability arbitrarily close to 1. See Appendix A for further details.

C. Alternative Proposal Constructions

Since IA^{2}RMS improves the adaptive structure of ARMS, simpler procedures can be used to build the function $W_t(x)$, reducing the overall computational cost and the coding effort. A first possibility is defining $W_t(x)$ inside the i-th interval simply as the straight line $L_{i,i+1}(x)$ going through $(s_i, V(s_i))$ and $(s_{i+1}, V(s_{i+1}))$ for $1 \leq i \leq m_t - 1$, and extending the straight lines corresponding to $\mathcal{I}_1$ and $\mathcal{I}_{m_t-1}$ towards $\pm\infty$ for the first and last intervals. Mathematically,

$$W_t(x) = L_{i,i+1}(x), \quad x \in \mathcal{I}_i = (s_i, s_{i+1}], \tag{6}$$

for $1 \leq i \leq m_t - 1$, $W_t(x) = L_{1,2}(x)$ in $\mathcal{I}_0 = (-\infty, s_1]$ and $W_t(x) = L_{m_t-1,m_t}(x)$ in $\mathcal{I}_{m_t} = (s_{m_t}, \infty)$. This is illustrated in Fig. 3(a). Note that, although this procedure looks similar to the one used in ARMS, as described by (5), it is actually much simpler, since it does not require the calculation of intersection points. Furthermore, an even simpler procedure to construct $W_t(x)$ can be devised from (6): using a piecewise constant approximation with two straight lines inside the first and last intervals. Mathematically,

$$W_t(x) = \max\{V(s_i), V(s_{i+1})\}, \quad x \in \mathcal{I}_i = (s_i, s_{i+1}], \tag{7}$$

for $1 \leq i \leq m_t - 1$, $W_t(x) = L_{1,2}(x)$ in $\mathcal{I}_0 = (-\infty, s_1]$ and $W_t(x) = L_{m_t-1,m_t}(x)$ in $\mathcal{I}_{m_t} = (s_{m_t}, \infty)$. This construction leads to the simplest proposal possible: a collection of uniform pdfs with two exponential tails. Fig. 3(b) shows an example of the construction of the proposal using this approach. A more sophisticated approach can be devised if the first derivative of $V(x)$ is available. Denoting the straight line tangent to $V(x)$ at $x^* \in \frac{s_i+s_{i+1}}{2}$ as $w_{i,i+1}(x)$, we can construct

$$W_t(x) = w_{i,i+1}(x), \quad x \in \mathcal{I}_i = (s_i, s_{i+1}], \tag{8}$$

for $1 \leq i \leq m_t - 1$, and two exponential tails for $\mathcal{I}_0$ and $\mathcal{I}_{m_t}$, built also using the first derivative of $V(x)$ (see Fig. 3(d)).

Alternatively, we could build the proposal $\pi_t(x)$ directly, instead of constructing $W_t(x)$ and setting $\pi_t(x) = \exp(W_t(x))$. Following this approach, we could apply the procedure described in [32] for adaptive trapezoidal Metropolis sampling (ATRAMS), even though the structure of this algorithm is completely different to ours. In this case, the proposal is constructed using straight lines $\widetilde{L}_{i,i+1}(x)$ passing through $(s_i, p(s_i))$ and $(s_{i+1}, p(s_{i+1}))$, i.e., *directly* in the domain of the target pdf, $p(x)$. Mathematically,

$$\pi_t(x) = \widetilde{L}_{i,i+1}(x), \quad x \in \mathcal{I}_i = (s_i, s_{i+1}], \tag{9}$$

for $1 \leq i \leq m_t - 1$, and the tails are two exponential pieces. Fig. 3(c) shows an example of a proposal using this approach. Finally, note that (7) would be identical in the pdf's domain, since $\exp(\max\{V(s_i), V(s_{i+1})\}) = \max\{p(s_i), p(s_{i+1})\}$. Furthermore, applying (9) directly in the pdf's domain could yield invalid proposals with $\pi_t(x) < 0$ inside some regions. Indeed, although many other alternatives can be considered to build the proposal, they have to satisfy the following basic properties:

1) Valid proposals are always obtained, i.e., $\pi_t(x) = \exp(W_t(x)) > 0 \forall x \in \mathcal{X} \subseteq \mathbb{R}$ and $t \in \mathbb{N}^+$.
2) The sequence of proposals $\{\pi_t\}_{t=0}^{\infty}$ tends to $p(x)$ (i.e., $D(\pi_t, p) \to 0$) when new support points are added.
3) Samples from $\widetilde{\pi}_t(x) \propto \pi_t(x)$ can be efficiently drawn.

The first condition is easily fulfilled in the log-domain, but restricts the use of some constructions in the pdf's domain. The second condition is the key for the good performance of the algorithm. The last condition is essential for practical purposes to

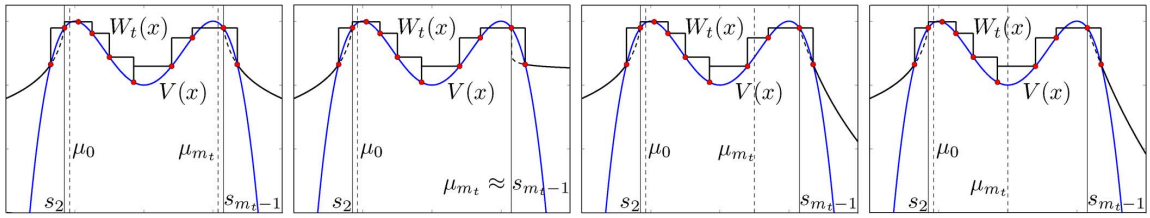

Fig. 4. Examples of proposal constructions in the log-domain, using the procedure described by (7) and the Pareto tails in (10), for different values of $\mu_{m_t} < s_{m_t-1}$. The tails always pass through the points $(s_1, V(s_1))$ and $(s_2, V(s_2))$ (left tail), and through $(s_{m_t-1}, V(s_{m_t-1}))$ and $(s_{m_t}, V(s_{m_t}))$ (right tail). **(a)-(b)** When $\mu_{m_t} \approx s_{m_t-1}$, the right tail becomes flatter. **(c)-(d)** When $\mu_{m_t} \ll s_{m_t-1}$, the right tail becomes closer to a vertical line.

obtain an efficient algorithm. When the proposal is a piecewise function, as in the four constructions proposed in this section, it implies being able to compute the area below each piece and drawing samples efficiently from each piece.

D. Alternative Construction for the Tails

Note that the construction of the tails can be modified in order to adapt the proposal to a specific class of targets (e.g., heavy-tailed distributions). For instance, an alternative to the exponential tails is using Pareto pieces:

$$\pi_t(x) = e^{\rho_j} \frac{1}{|x - \mu_j|^{\gamma_j}}, \quad \forall x \in \mathcal{I}_j, \tag{10}$$

for $j \in \{0, m_t\}$ and $\gamma_j > 1$. Fixing the parameter μ_0, ρ_0 and γ_0 are obtained solving the following system of equations:

$$V(s_1) = \rho_0 + \gamma_0 \log\left(\frac{1}{|s_1 - \mu_0|}\right),$$
$$V(s_2) = \rho_0 + \gamma_0 \log\left(\frac{1}{|s_2 - \mu_0|}\right),$$

which ensure that the left tail passes through the points $(s_1, V(s_1))$ and $(s_2, V(s_2))$. Similarly, fixing μ_{m_t}, we obtain an equivalent system of equations by forcing the right tail to pass through $(s_{m_t-1}, V(s_{m_t-1}))$ and $(s_{m_t}, V(s_{m_t}))$. The solution of these two systems of equations is straightforward, and the parameters μ_j can be arbitrarily chosen by the user, as long as $\mu_0 > s_2$, $\mu_{m_t} < s_{m_t-1}$ and they lead to $\gamma_0, \gamma_{m_t} > 1$.[6] Selecting $\mu_0 \approx s_2$ and $\mu_{m_t} \approx s_{m_t-1}$ yields smaller values of γ_0 and γ_{m_t} (i.e., closer to 1), and thus fatter tails. Larger values (i.e., $\mu_0 \gg s_2$ and $\mu_{m_t} \ll s_{m_t-1}$) yield larger values of γ_0 and γ_{m_t}, and thus lighter tails. Finally, note that the integral of $\pi_t(x)$ in $\mathcal{I}_0$ and $\mathcal{I}_{m_t}$ can be computed analytically, and that we can easily draw from each Pareto tail using the inversion method [2].

IV. Numerical Results

In this section, we compare the performance of IA^{2}RMS with other methods developed in the literature by drawing samples from different target distributions. First of all, we consider two one-dimensional examples, where we generate samples directly from two univariate distributions: a Gaussian mixture and a heavy-tailed distribution. Then, we also consider three examples of the performance of IA^{2}RMS within Gibbs sampling: two bi-dimensional toy examples and a real-world target localization application in a wireless sensor network.[7]

A. Multimodal Target: Mixture of Gaussians

1) Comparison With Standard ARMS: As a first toy example, we compare the performance of IA^{2}RMS and ARMS on a multimodal one-dimensional target pdf, $p_o(x)$, generated as a mixture of 3 Gaussian densities,

$$p_o(x) = 0.3\mathcal{N}(x; -5, 1) + 0.3\mathcal{N}(x; 1, 1) + 0.4\mathcal{N}(x; 7, 1), \tag{11}$$

where $\mathcal{N}(x; \mu, \sigma^2)$ indicates that the random variable X has a Gaussian pdf with mean μ and variance σ^2. We test the four alternative procedures previously described to build the proposal, both for ARMS and IA^{2}RMS. In all cases, we consider $N = 5000$ iterations of the Markov chain (without removing any samples to account for the "burn-in" period), 2000 runs of the algorithm to average the results, and an initial support set $\mathcal{S}_0 = \{s_1 = -10, s_2 = a, s_3 = b, s_4 = 10\}$ with $m_0 = 4$ support points, where $a, b \sim \mathcal{U}([-10, 10])$ with $a < b$.

Table V compares the results obtained for ARMS and IA^{2}RMS. The standard ARMS algorithm, as proposed in [28], corresponds to the first row on the left-hand side of Table V, i.e., ARMS adaptive structure with the construction in (5) for the proposal. The first column in both cases shows the estimated mean of $p_o(x)$ (true value, $\mu = \mathrm{E}\{X\} = 1.6$) averaged over 2000 runs, altogether with the estimator's standard deviation. Note that IA^{2}RMS always outperforms ARMS, regardless of the proposal construction scheme. Indeed, the three approaches of (7)–(9) provide estimates of the mean which are very close to the true value with a large decrease in the standard deviation w.r.t. ARMS. This can also be appreciated in the mean squared error (MSE) value, $\mathrm{MSE} = (\hat{\mu} - \mu)^2 + \hat{\sigma}^2$, shown in the second columns: the worst-case result for IA^{2}RMS is almost as good as the best-case result for ARMS.

Additionally, we also provide an estimation of the linear correlation among consecutive samples at lag 1, $\hat{R}_{xx}[1]$, and an estimate of the final L_1 distance between the unnormalized proposal and the target pdf, $\hat{D}(\pi_T, p)$.[8] These results confirm the better performance of IA^{2}RMS w.r.t. ARMS. On the one hand, the proposal always converges to the target for IA^{2}RMS, as evidenced by the low values of $\hat{D}(\pi_T, p)$ when compared with the corresponding values for ARMS. On the other hand, this causes

[6] If $\gamma_0 \leq 1$ (resp. $\gamma_{m_t} \leq 1$) is obtained, then a larger value of μ_0 (resp. a smaller value of μ_{m_t}) must be selected and the system of equations solved again until $\gamma_0 > 1$ (resp. $\gamma_{m_t} > 1$) is obtained. Fig. 4 depicts some examples of this construction in the log-domain for different values of μ_{m_t}.

[7] Matlab code for IA^{2}RMS using the procedures in (6)–(9) to build the proposal is available at http://a2rms.sourceforge.net/.

[8] $D(\pi_t, p)$ is related to the probability of accepting the proposed samples [42], and thus also to the number of support points added (see Appendix D).

TABLE V
COMPARISON OF IA^{2}RMS VS. ARMS ADAPTIVE STRUCTURES FOR THE GAUSSIAN MIXTURE EXAMPLE: STANDARD ARMS IS OBTAINED USING THE PROPOSAL IN (5). $\hat{\mu} \pm \hat{\sigma}$: ESTIMATED MEAN $\pm$ STANDARD DEVIATION; MSE: MEAN SQUARED ERROR; $\hat{R}_{xx}[1]$: CORRELATION AT LAG 1; $\hat{D}(\pi_T, p)$: ESTIMATE OF $D(\pi_t, p)$ at $t = T$; $|\mathcal{P}_T|$: AVERAGE NUMBER OF PIECES IN $\pi_t(x)$ at $t = T$; TIME: NORMALIZED TIME W.R.T. STANDARD ARMS

Proposal	**ARMS Adaptive Structure**						**IA^{2}RMS Adaptive Structure**					
$\pi_t(x)$	$\hat{\mu} \pm \hat{\sigma}$	MSE	$\hat{R}_{xx}[1]$	$\hat{D}(\pi_T, p)$	$\lvert\mathcal{P}_T\rvert$	Time	$\hat{\mu} \pm \hat{\sigma}$	MSE	$\hat{R}_{xx}[1]$	$\hat{D}(\pi_T, p)$	$\lvert\mathcal{P}_T\rvert$	Time
Eq. (5)	1.648 ± 0.730	0.396	0.535	3.002	91.272	1.000	1.623 ± 0.124	0.016	0.004	0.061	123.844	1.034
Eq. (6)	1.754 ± 1.091	1.214	0.772	8.052	**12.049**	**0.137**	1.724 ± 0.219	0.063	0.020	0.253	85.644	0.208
Eq. (7)	1.594 ± 0.230	0.053	0.613	6.152	164.188	0.406	1.601 ± 0.095	0.009	**0.002**	0.201	317.536	0.612
Eq. (8)	1.569 ± 0.386	0.149	0.747	7.034	66.960	0.294	$\mathbf{1.599 \pm 0.084}$	**0.007**	0.007	0.115	102.149	0.357
Eq. (9)	1.567 ± 0.496	0.355	0.708	7.134	37.823	0.212	1.601 ± 0.131	0.017	0.005	**0.058**	92.133	0.281

most of the proposed samples to be accepted, thus leading to a greatly reduced correlation among consecutive samples (more than two orders of magnitude in some cases).

Table V also provides the average number of linear pieces in the final proposal, $|\mathcal{P}_T|$. For the proposal construction schemes of (6)–(9), $|\mathcal{P}_T|$ corresponds directly to the average final number of intervals. However, using the procedure proposed in [28] in general we have more pieces than intervals, as noted before. In any case, regardless of the proposal building scheme, the number of intervals in IA^{2}RMS always increases w.r.t. ARMS. However, this increase is always moderate, controlled (see Appendix D for a bound on the number of support points), and should not be considered a disadvantage of IA^{2}RMS. Indeed, it is an evidence of the structural limitation of ARMS (see Section II-D), and allows IA^{2}RMS to obtain much better results than ARMS (in terms of estimated mean, standard deviation, MSE, autocorrelation, and distance between the proposal and the target) with a moderate increase in storage and computational cost.

Finally, the last columns in Table V show the normalized time spent by the different techniques, considering 1.0 to be the simulation time required by the standard ARMS method of [28]. On the one hand, the simulation time for IA^{2}RMS always increases w.r.t. ARMS when using the same proposal construction scheme in both cases. On the other hand, the simulation time is greatly reduced by using some of the simpler proposal construction procedures considered in the paper. Indeed, IA^{2}RMS with any of the three simple procedures proposed requires less time than the standard ARMS. This is due to the fact that these methods reject less candidate points in the RS test than the standard ARMS. For instance, using the procedure of (9) only 62.5% of the added support points have been rejected in the RS test (where the chain does not move forward), whereas the remaining points are incorporated in the second control test (where the chain is not stopped). In ARMS, since the second control test does not exist, support points can only be added after a rejection in the RS test, which implies that the chain always has to be stopped in order to improve the proposal. Table VI shows the average number of support points added in the control tests for ARMS and IA^{2}RMS using all the proposed constructions. Note that, with the exception of the construction in (5) (first row), the sum of the number of points incorporated in both tests plus $m_0 = 4$ (the number of initial support points) provides the final number of pieces in the proposal in Table V, $|\mathcal{P}_T|$.

2) Comparison With Other Techniques: In order to show the good performance of IA^{2}RMS, we compare it to three

TABLE VI
NUMBER OF SUPPORT POINTS ADDED IN THE CONTROL TESTS

Proposal	**ARMS**		**IA^{2}RMS**	
	RS test	**Second test**	**RS test**	**Second test**
Eq. (5)	73.87	0	85.16	9.68
Eq. (6)	8.049	0	10.01	71.62
Eq. (7)	160.19	0	306.25	7.28
Eq. (8)	62.96	0	89.43	8.71
Eq. (9)	33.82	0	55.01	33.11

widely used sampling techniques: slice sampling [1, Chapter 6], the standard MH algorithm [2], and Multiple Try Metropolis (MTM) [1, Chapter 5]. For the MH and MTM schemes, we use Gaussian proposal pdfs with different means ($\mu \in \{0, 1.6\}$) and variances ($\sigma^2 \in \{1, 4, 25.84\}$). We also test independent and random walk proposals, i.e., $\widetilde{\pi}(x_n) \propto \exp((x_n - \mu)^2/(2\sigma^2))$ and $\widetilde{\pi}(x_n|x_{n-1}) \propto \exp((x_n - x_{n-1})^2/(2\sigma^2))$ respectively.[9] Regarding the MTM schemes, we experiment with $M \in \{10^2, 10^3\}$ tries, and importance weights designed to choose the best candidate in each step [43].[10]

Table VII shows the results obtained, for the target pdf in (11), using similar settings as before: 2000 runs, $\widetilde{N} = 5400$ iterations of the Markov chain,[11] and performing the estimation with all the samples generated. The results of standard ARMS (i.e., ARMS with the proposal construction procedure of (5)) and IA^{2}RMS using the procedure of (7) to build the proposal, are recalled in the first two rows. Note that the only technique that outperforms IA^{2}RMS is MTM with an independent proposal (MTM-ind), $M = 10^3$ tries, $\mu = 1.6$ and $\sigma^2 = 25.84$. However, this approach is highly tailored to the particular target considered in this case and its computational cost is much higher than that of IA^{2}RMS. Furthermore, we notice that the choice of the parameters has a great influence on the results for the MH and MTM approaches, even though the target distribution is simply a mixture of 3 Gaussians. Indeed, Table VII shows that a poor performance can be easily obtained when an unsuitable choice is made. For this reason, it is preferable to use an adaptive black-box technique like IA^{2}RMS.

[9]Note that the values $\mu = 1.6$ and $\sigma^2 = 25.84$ are exactly the mean and the variance of the target distribution (i.e., in this case the proposal shares the first and second order moments with the target).

[10]In general, MTM schemes need to draw M samples per iteration plus $M - 1$ auxiliary points [1]. On the one hand, MTM degenerates into a standard MH when $M = 1$. On the other hand, its computational cost per iteration of the Markov chain increases substantially as M is increased.

[11]We use $\widetilde{N} = 5400 > N = 5000$ iterations of the Markov chain to ensure a fair comparison with IA^{2}RMS, since the Markov chain is stopped whenever a sample is rejected in the RS test, both for ARMS and IA^{2}RMS.

TABLE VII
RESULTS WITH DIFFERENT TECHNIQUES USING $N = 5400$ FOR THE GAUSSIAN MIXTURE EXAMPLE

Technique	**Estimated mean** $\hat{\mu} \pm \hat{\sigma}$	**Mean Squared Error** MSE	**Correlation** $\hat{R}_{xx}[1]$	**Parameters and further informations**
ARMS	1.648 ± 0.730	0.396	0.535	standard ARMS as proposed in [28]
IA^{2}RMS	**1.601 ± 0.095**	0.009	0.002	with proposal pdf in Eq. (7)
Slice Sampler	1.593 ± 0.141	0.020	0.577	standard implementation in MATLAB
MH-ind	0.222 ± 0.980	2.859	0.874	$\mu = 0$, $\sigma^2 = 1$
	1.215 ± 1.719	3.103	0.948	$\mu = 0$, $\sigma^2 = 4$
	1.594 ± 0.153	0.023	0.621	$\mu = 1.6$, $\sigma^2 = 25.84$
MH-rw	1.578± 0.656	0.431	0.973	$\sigma^2 = 4$
MTM-ind	-0.150± 0.151	3.085	0.704	$M = 100$, $\mu = 0$, $\sigma^2 = 1$,
	-0.765 ± 1.247	7.148	0.756	$M = 1000$, $\mu = 0$, $\sigma^2 = 1$
	1.598± 0.069	**0.005**	**0.0002**	$M = 1000$, $\mu = 1.6$, $\sigma^2 = 25.84$
MTM-rw	1.590 ± 0.525	0.276	0.954	$M = 1000$, $\sigma^2 = 1$
	1.597 ± 0.175	0.031	0.733	$M = 1000$, $\sigma^2 = 4$

TABLE VIII
ESTIMATION OF THE CONSTANT $\frac{1}{c_p} = \frac{1}{\sqrt{\pi}} = 0.5642$ FOR THE LÉVY DISTRIBUTION

Technique	**Estimation of** $\frac{1}{c_p}$	**Mean Squared Error MSE**	**Iterations of the chain** (N)	**Further informations**
IA^{2}RMS	**0.5652 ± 0.0014**	**0.0014**	5000	with proposal pdf in Eq. (9)
MTM-ind	0.6056 ± 0.0012	0.0027	5400	$M = 1000$, $\mu = 10$, $\sigma^2 = 2500$
	0.5994 ± 0.0010	0.0022	5400	$M = 1000$, $\mu = 100$, $\sigma^2 = 2500$
MTM-rw	0.5819 ± 0.0050	0.0053	5400	$M = 1000$, $\sigma^2 = 2500$

B. Heavy-Tailed Distribution

In this section, we show that IA^{2}RMS can be applied to draw samples from heavy-tailed distributions, even by using a proposal pdf with exponential (i.e., "light") tails. As an example, we consider the *Lévy distribution*, which is a special case of the inverse-gamma distribution, and has a pdf

$$p_o(x) \propto p(x) = \frac{1}{(x-\lambda)^{\frac{3}{2}}} \exp\left(-\frac{\nu}{2(x-\lambda)}\right), \qquad (12)$$

$\forall x \geq \lambda$. The normalizing constant, $\frac{1}{c_p}$, such that $p_o(x) = \frac{1}{c_p} p(x)$ integrates to one, can be determined analytically, and is given by $\frac{1}{c_p} = \sqrt{\frac{\nu}{2\pi}}$. However, given a random variable $X \sim p_o(x)$, all the moments $E[X^k]$ with $k \geq 1$ do not exist, due to the heavy-tailed feature of the Lévy distribution.

Our goal is estimating the normalizing constant $\frac{1}{c_p}$ via Monte Carlo simulation, when $\lambda = 0$ and $\nu = 2$. We use IA^{2}RMS with the construction procedure in (9), which provides the best trade-off between performance and computational cost. We start with only $m_0 = 3$ support points, $\mathcal{S}_0 = \{s_1 = 0, s_2, s_3\}$, where $s_2, s_3 \sim \mathcal{U}([1, 10])$ with $s_2 < s_3$. We also apply three different MTM techniques using $M = 1000$ tries and importance weights designed again to choose the best candidate in each step [43]. In the first two schemes (MTM-ind), we use an independent proposal $\widetilde{\pi}(x_n) \propto \exp((x_n - \mu)^2/(2\sigma^2))$ with $\mu \in \{10, 100\}$ and $\sigma^2 = 2500$. In the last one (MTM-rw), we use a random walk proposal $\widetilde{\pi}(x_n|x_{n-1}) \propto \exp((x_n - x_{n-1})^2/(2\sigma^2))$ with $\sigma^2 = 2500$. Note that we need to choose huge values of σ^2 due to the heavy-tailed feature of the target. As before, we use $N = 5000$ iterations for IA^{2}RMS and $\widetilde{N} = 5400$ for the MTM schemes in order to ensure a fair comparison.[12]

[12]Note that this choice penalizes IA^{2}RMS with the construction in (9) in the comparison with the other methods, since the number of rejections in the RS test is only ≈ 55 in this case (see Table VI).

The results, averaged over 2000 runs, are summarized in Table VIII. Note that the real value of $\frac{1}{c_p}$ when $\nu = 2$ is $\frac{1}{\sqrt{\pi}} = 0.5642$. The IA^{2}RMS algorithm provides better results than all of the MTM approaches tested with only a fraction of their computational cost. Furthermore, IA^{2}RMS avoids the critical issue of parameter selection (selecting a small value of σ^2 in this case can easily lead to very poor performance).

C. Application Within Gibbs

1) Toy Example 1: Consider two Gaussian full-conditional densities,

$$p(x_1|x_2) \propto \exp\left(-\frac{(x_1 - 0.5x_2)^2}{2\sigma_1^2}\right), \qquad (13)$$

$$p(x_2|x_1) \propto \exp\left(-\frac{(x_2 - 0.5x_1)^2}{2\sigma_2^2}\right), \qquad (14)$$

with $\sigma_1 = 1$ and $\sigma_2 = 0.2$. The joint pdf is a bivariate Gaussian pdf with mean vector $\boldsymbol{\mu} = [0, 0]^{\mathsf{T}}$ and covariance matrix $\boldsymbol{\Sigma} = [1.08\ 0.54;\ 0.54\ 0.31]$. We apply a Gibbs sampler with N_G iterations to estimate both the mean and the covariance of the joint pdf. Then, we calculate the average MSE in the estimation of all the elements in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, averaged over 2000 independent runs. We use this simple case, where we can draw directly from the full-conditionals, to check the performance of MH and IA^{2}RMS within Gibbs as a function of N and N_G. For the MH scheme we use a Gaussian random walk proposal, $\pi(x_{d,n}^{(i)}) \propto \exp(-(x_{d,n}^{(i)} - 0.5x_{d,n-1}^{(i)})^2/(2\sigma_p^2))$ for $d \in \{1, 2\}$, $1 \leq n \leq N$ and $1 \leq i \leq N_G$. For IA^{2}RMS we use the construction of (9) with $\mathcal{S}_0 = \{-2, 0, 2\}$.

In the first experiment, we set $N_G = 10^3$ and $x_{d,0}^{(i)} = 1$ (both for MH and IA^{2}RMS), and increase the value of N. The results can be seen in Fig. 5(a), (b): IA^{2}RMS attains almost the same performance as the ideal case (sampling directly from the full

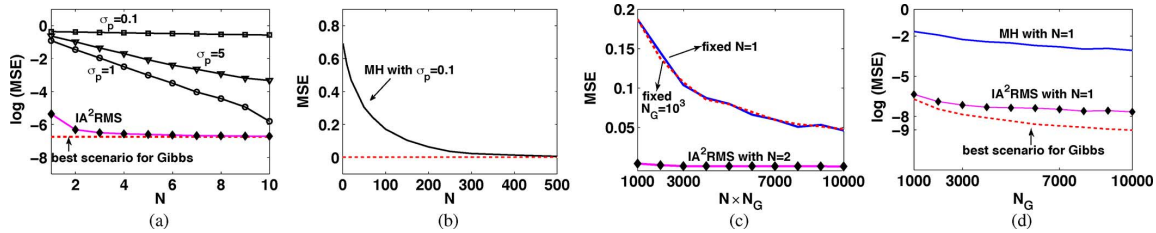

Fig. 5. **(a)-(b)** MSE as function of the number of iterations in the Markov chain (N) for $N_G = 1000$. The constant dashed line is the MSE (≈ 0.0012) obtained drawing directly from the full-conditionals. **(c)** MSE as a function of the product $N \times N_G$ for the MH-within-Gibbs (both fixing $N = 1$ and letting $N_G \in [10^3, 10^4]$, and fixing $N_G = 10^3$ and letting $N \in [1, 10]$) and IA^{2}RMS within Gibbs (fixing $N = 2$ and letting $N_G \in [500, 5000]$). **(d)** MSE as function of the number of iterations in the Gibbs sampler (N_G) for $N = 1$. The dashed line is again the MSE drawing from the full-conditionals.

conditionals) for small values of N, whereas the MH scheme needs a substantially larger value of N (up to $N = 500$ for $\sigma_p = 0.1$) to attain a similar performance. In a second experiment, we check the behaviour of both approaches as N_G is increased. Now, we set $x_{d,0}^{(0)} = 1$ and $x_{d,0}^{(i)} = x_{d,N}^{(i-1)}$ for $i \geq 1$ (both for MH and IA^{2}RMS), and compare the performance of MH with $\sigma_p = 0.1$ and IA^{2}RMS.[13] Fig. 5(c) compares the performance of MH-within-Gibbs (both fixing $N = 1$ and letting $N_G \in [10^3, 10^4]$, and fixing $N_G = 10^3$ and letting $N \in [1, 10]$) and IA^{2}RMS within Gibbs (fixing $N = 2$ and letting $N_G \in [500, 5000]$). First of all, we notice that N and N_G have the same effect for MH-within-Gibbs: as long as $N \times N_G$ is constant we obtain the same results by increasing any of them. Then, we notice that IA^{2}RMS (with an MSE $= 0.0029$ for $N \times N_G = 10^3$ and MSE $= 0.0003$ for $N \times N_G = 10^4$) clearly outperforms the MH scheme. Finally, Fig. 5(d) shows the effect of increasing N_G for a fixed value of N. Once more, the MSE of IA^{2}RMS is well below the MSE attained by MH.

2) Toy Example 2: Let us consider the target density

$$p_o(x_1, x_2) \propto \exp\left(-\frac{\left(x_1^2 - A + Bx_2\right)^2}{4} - \frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2}\right),$$

with $A = 16$, $B = 10^{-2}$, and $\sigma_1^2 = \sigma_2^2 = \frac{10^4}{2}$. Densities of this analytic form are frequently used in the statistical literature (cf. [43], [44]) to compare the performance of different Monte Carlo algorithms. We apply a Gibbs sampler to draw from $p_o(x_1, x_2)$, using the standard ARMS method [28] (i.e., with the construction in (5)) and the IA^{2}RMS technique (with the proposal construction in (9)) within the Gibbs sampler to generate samples from the full-conditionals, starting always with the initial support set $\mathcal{S}_0 = \{-10, -6, -4.3, -0.01, 3.2, 3.8, 4.3, 7, 10\}$. For each full-conditional pdf, we draw N samples and take the last one as the output from the Gibbs sampler. We also apply a standard MH algorithm with a random walk proposal $\widetilde{\pi}(x_{d,n}^{(i)}|x_{d,n-1}^{(i)}) \propto \exp(-(x_{d,n}^{(i)} - x_{d,n-1}^{(i)})^2/(2\sigma_p^2))$ for $d \in \{1, 2\}$, $\sigma_p \in \{1, 2, 10\}$, $1 \leq n \leq N$ and $1 \leq i \leq N_G$. Furthermore, we test an *ad-hoc MH* scheme (i.e., specifically designed for this target pdf): for the full-conditionals w.r.t. x_1, we consider an independent proposal pdf $\widetilde{\pi}(x_{1,n}^{(i)}) = 0.5\mathcal{N}(x_{1,n}^{(i)}; -4, \sigma_{p,1}^2) + 0.5\mathcal{N}(x_{1,n}^{(i)}; 4, \sigma_{p,1}^2)$, with $\sigma_{p,1} \in \{1, 2, 3\}$, whereas for the full-conditionals w.r.t. x_2, we consider again the random walk proposal $\widetilde{\pi}(x_{2,n}^{(i)}|x_{2,n-1}^{(i)}) \propto \exp(-(x_{2,n}^{(i)} - x_{2,n-1}^{(i)})^2/(2\sigma_{p,2}^2))$ with $\sigma_{p,2} = 10$.

We consider two initializations for all the methods-within-Gibbs: **(In1)** $x_{d,0}^{(i)} = 1$; **(In2)** $x_{d,0}^{(0)} = 1$ and $x_{d,0}^{(i)} = x_{d,N}^{(i-1)}$ for $i = 1, \ldots, N_G$. We run N_G iterations of the Gibbs sampler, using all the samples to estimate four statistics that involve the first four moments of the target: mean, variance, skewness and kurtosis. Fig. 6(a) illustrates the target $p_o(\mathbf{x})$, and Fig. 6(b), (c) show the *mean absolute error* (MAE) as a function of N for different techniques averaged over 1000 runs. Table IX provides the numerical results (i.e., the average MAE for each of the four statistics estimated), and the time required by the Gibbs sampler (normalized by considering 1.0 to be the time required by ARMS with $N = 50$). We also provide a measure of efficiency, defined as the inverse of the averaged MAE over the required time and normalized w.r.t. the maximum value (IA^{2}RMS with In2, $N = 3$ and $N_G = 2000$).

First of all, we notice that IA^{2}RMS outperforms ARMS for all values of N, showing that the IA^{2}RMS adaptive structure speeds up the convergence of the Markov chain. For instance, IA^{2}RMS with only $N = 3$ provides better results than ARMS with $N = 50$, saving 95% of the computation time. Regarding the use of the MH algorithm within Gibbs, the results depend largely on the choice of the variance of the proposal, σ_p^2, and the initialization, showing the need for adaptive MCMC strategies. Indeed, for an inadequate scale parameter (e.g., $\sigma_p = 1$ or $\sigma_p = 2$), even IA^{2}RMS with only $N = 3$ provides better results than MH with $N = 100$ or $N = 1000$. On the other hand, when a good σ_p (i.e., $\sigma_p = 10$) and initialization (i.e., In2) are selected, MH with $N = 100$ provides virtually the same performance (and with the same computational cost) as IA^{2}RMS with $N = 50$, showing that nothing is lost by using IA^{2}RMS and there is much to gain in terms of robustness w.r.t. parameter selection. Finally, Table IX shows also the importance of increasing N in this case: $N = 50$ and $N_G = 2000$ provides better results (and with a lower computational cost) than $N = 1$ and $N_G = 30000$. In any case, for a fixed value of $N \times N_G$, IA^{2}RMS always provides the best averaged MAE for a given N with just a slight increase in the computation time.

3) Target Localization in a Wireless Sensor Network: In this section, we consider the problem of positioning a target in a two-dimensional space using range and angle measurements. This is a problem that appears frequently in localization applications using sensor networks [18], [45], [46]. More formally, we

[13]Note that we have chosen a "bad" value of σ_p for Fig. 5(c), (d) to illustrate the danger of a bad parameterization in the MH-within-Gibbs approach. However, even with a "good" value of σ_p, the performance of the MH scheme is much worse than that of IA^{2}RMS, as shown by the following numerical example.

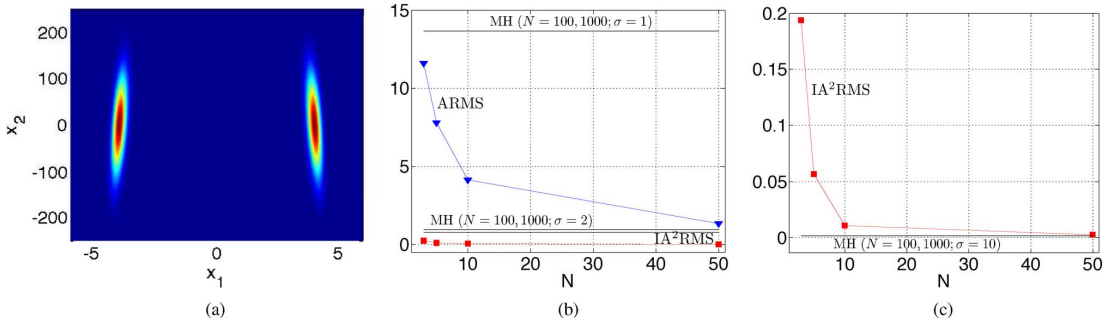

Fig. 6. **(a)** The target pdf, $p_o(x_1, x_2)$, used in Section IV-C-2. **(b)** MAE in estimation of the kurtosis (first component) as a function of N for ARMS (triangles) and IA^{2}RMS (squares), and the MAEs for the MH algorithm with $N \in \{100, 1000\}$ and $\sigma \in \{1, 2\}$ (constant lines). (**c**) MAE in estimation of the kurtosis (first component) as a function of N for IA^{2}RMS (squares), and the MH algorithm with $N \in \{100, 1000\}$ and $\sigma = 10$ (constant line).

TABLE IX
MEAN ABSOLUTE ERROR (MAE) IN THE ESTIMATION OF FOUR STATISTICS (FIRST COMPONENT) AND NORMALIZED TIME AND EFFICIENCY FOR THE EXAMPLE IN Section IV-C-2. ALL THE TECHNIQUES ARE USED WITHIN A GIBBS SAMPLER

Technique	N	N_G	Init.	MAE				Avg. MAE	Time	Efficiency $\frac{(\text{Avg. MAE})^{-1}}{\text{Time}}$
				Mean	Variance	Skewness	Kurtosis			
IA^{2}RMS	3	2000	In1	0.834	0.688	0.435	0.193	0.537	0.047	0.115
	5			0.453	0.186	0.233	0.056	0.232	0.086	0.146
	10			0.186	0.043	0.095	0.010	0.083	0.154	0.228
	50			**0.077**	**0.036**	**0.039**	**0.002**	**0.038**	0.555	0.138
ARMS	3	2000	In1	3.408	11.580	3.384	11.572	7.486	0.077	0.005
	5			3.151	9.839	2.650	7.079	5.679	0.116	0.004
	10			2.798	7.665	2.024	4.124	4.152	0.223	0.003
	50			1.918	3.407	1.134	1.292	1.937	1.000	0.001
MH ($\sigma_p = 1$)	100	2000	In1	3.509	12.308	3.671	13.666	8.288	0.540	$6 \cdot 10^{-4}$
MH ($\sigma_p = 2$)				1.756	3.077	0.978	0.963	1.693	0.540	0.003
MH ($\sigma_p = 10$)				**0.075**	**0.037**	**0.036**	**0.002**	**0.038**	0.540	0.142
MH ($\sigma_p = 1$)	1000	2000	In1	3.508	12.302	3.665	13.624	8.274	3.229	10^{-4}
MH ($\sigma_p = 2$)				1.601	2.560	0.874	0.769	1.451	3.229	$6 \cdot 10^{-4}$
MH ($\sigma_p = 10$)				**0.074**	**0.036**	**0.036**	**0.002**	**0.037**	3.229	0.024
MH ($\sigma_p = 10$)	1	2000	In1	0.697	11.598	0.883	3.622	4.200	**0.033**	0.021
		6000		0.587	10.329	0.721	3.317	3.738	0.089	0.009
		10000		0.493	9.881	0.611	2.905	3.472	0.162	0.005
	3	2000		0.352	6.510	0.290	0.927	2.019	0.042	0.034
	5			0.205	3.916	0.134	0.460	1.178	0.053	0.047
	10			0.085	1.411	0.043	0.160	0.424	0.081	0.085
	50			**0.077**	0.053	**0.037**	0.007	0.043	0.290	0.234
IA^{2}RMS	3	2000	In2	0.118	0.063	0.061	0.005	0.062	0.047	**1**
	5			0.091	0.046	0.047	**0.002**	0.046	0.086	**0.736**
	10			**0.077**	**0.035**	**0.038**	**0.002**	**0.038**	0.154	0.498
MH ($\sigma_p = 10$)	3	2000	In2	0.331	0.246	0.167	0.045	0.197	0.042	0.352
	5			0.247	0.172	0.123	0.024	0.141	0.053	**0.861**
	10			0.184	0.101	0.092	0.013	0.097	0.081	0.370
	1	10000		0.178	0.126	0.091	0.012	0.102	0.162	0.176
		20000		0.151	0.112	0.090	0.008	0.090	0.331	0.098
		30000		0.138	0.063	0.068	0.007	0.069	0.502	0.084
Ad-hoc MH-1	5	2000	In2	0.100	0.091	0.050	0.004	0.061	0.054	**0.884**
	10			**0.076**	0.073	**0.038**	**0.002**	0.047	0.082	**0.756**
Ad-hoc MH-2	5	2000	In2	0.136	0.098	0.068	0.007	0.077	0.054	**0.700**
	10			0.109	0.075	0.054	0.004	0.061	0.082	0.582
Ad-hoc MH-3	5	2000	In2	0.149	0.110	0.074	0.009	0.085	0.054	0.634
	10			0.120	0.079	0.060	0.005	0.066	0.082	0.538

consider a random vector $\mathbf{X} = [X_1, X_2]^\mathsf{T}$ denoting the target's position in $\mathbb{R}^2$. The range measurements are obtained from 6 sensors located at $\mathbf{h}_1 = [-5, 1]^\mathsf{T}$, $\mathbf{h}_2 = [-2, 6]^\mathsf{T}$, $\mathbf{h}_3 = [0, 0]^\mathsf{T}$, $\mathbf{h}_4 = [5, -6]^\mathsf{T}$, $\mathbf{h}_5 = [6, 4]^\mathsf{T}$ and $\mathbf{h}_6 = [-4, -4]^\mathsf{T}$. The observation equations are given by

$$R_j = -20 \log \left(\frac{\|\mathbf{X} - \mathbf{h}_j\|}{0.3} \right) + \Omega_j, \quad j = 1, \ldots, 6, \qquad (15)$$

where Ω_j are i.i.d. Gaussian random variables, $\Omega_j \sim \mathcal{N}(\omega_j; 0, 50)$. Moreover, we consider 4 additional sensors that measure angle variations,

$$Z_j = 20 \arctan \left(\frac{X_2 - \psi_{j,1}}{X_1 - \psi_{j,2}} \right) + \Theta_j, \quad j = 1, \ldots, 4, \qquad (16)$$

with $(\psi_{1,1}, \psi_{1,2}) = (7, 7)$, $(\psi_{2,1}, \psi_{2,2}) = (7, -7)$, $(\psi_{3,1}, \psi_{3,2}) = (-7, 7)$, and $(\psi_{4,1}, \psi_{4,2}) = (-7, -7)$. Θ_j are

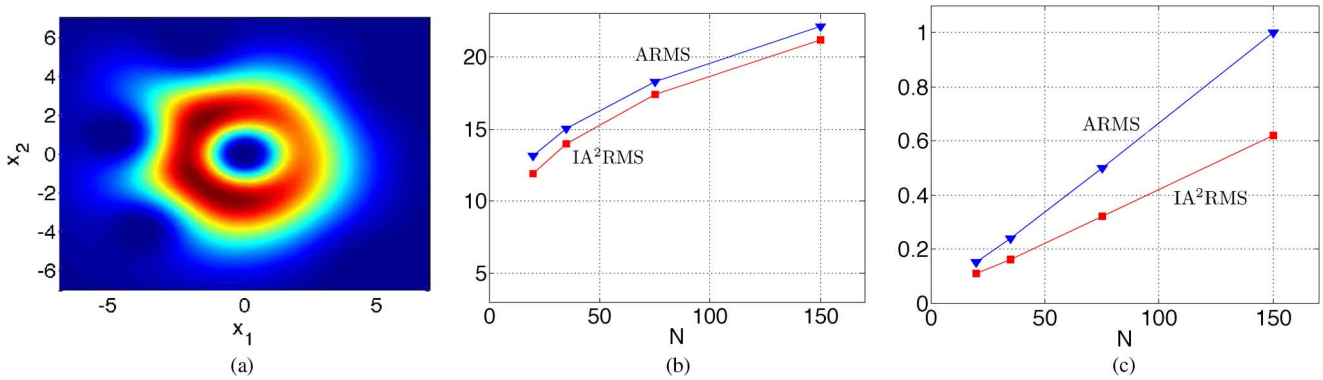

Fig. 7. ($\mathbf{a}$) The target pdf $p_o(\mathbf{x})$ for the localization example in Section IV-C-3. ($\mathbf{b}$) Average number of linear pieces in the proposal, $\mathcal{P}_T$, as function of N for ARMS (triangles) and IA^{2}RMS (squares). ($\mathbf{b}$) Simulation time (normalized w.r.t. the time required by ARMS with $N = 150$) as function of N of ARMS (triangles) and IA^{2}RMS (squares).

TABLE X
NUMERICAL RESULTS FOR THE LOCALIZATION EXAMPLE. N: NUMBER OF ITERATIONS FOR THE INTERNAL MCMC; MAE: MEAN ABSOLUTE ERROR IN THE ESTIMATION OF E$\{X_1\}$; $|\mathcal{P}_T|$: AVERAGE NUMBER OF PIECES IN THE FINAL PROPOSAL; TIME: NORMALIZED SIMULATION TIME

Technique	N	**MAE**	$\lvert\mathcal{P}_T\rvert$	**Time**	**Technique**	N	**MAE**	$\lvert\mathcal{P}_T\rvert$	**Time**
ARMS	20	0.0039	13.17	0.152	IA^{2}RMS	20	**0.0030**	**11.88**	**0.110**
ARMS	150	0.0031	22.14	1.000	IA^{2}RMS	150	**0.0030**	21.19	0.620

also Gaussian random variables: $\Theta_j \sim \mathcal{N}(\theta_j; 0, 5/3)$ for $j = 1, 2, 3$, and $\Theta_4 \sim \mathcal{N}(\theta_4; 0, 5)$.

Let us assume that the measurement vector for the range is $\mathbf{r} = [26, 26.5, 25, 28, 28, 25.3]^{\mathsf{T}}$, whereas the angular measurement vector is $\mathbf{z} = [5\pi, -5\pi, -4.5\pi, 5\pi]^{\mathsf{T}}$, and the complete observation vector is $\mathbf{y} = [\mathbf{r}^{\mathsf{T}}, \mathbf{z}^{\mathsf{T}}]^{\mathsf{T}}$. In order to perform Bayesian inference, we consider a non-informative prior over $\mathbf{X}$ (i.e., an improper uniform density on $\mathbb{R}^2$), and study the posterior pdf, $p_o(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$. Our goal is performing a Monte Carlo computation of the expected value of $\mathbf{X}$ given $\mathbf{Y}$, $E\{\mathbf{X}|\mathbf{Y}\}$, which is the minimum MSE (MMSE) estimate of the target's position.

In order to draw samples from $p_o(\mathbf{x})$ we perform $N_G = 2000$ iterations of a Gibbs sampler, using both the standard ARMS method of [28] (with the construction in (5)) and the IA^{2}RMS technique (with the construction in (9)) as internal MCMC schemes. In both cases, we start with an initial support set $S_0 = \{-6, 4, 6\}$ and consider $N \in \{20, 150\}$ iterations of the chain. Fig. 7(a) depicts the target pdf, $p_o(\mathbf{x})$, whereas Fig. 7(b) and (c) illustrate the average number of pieces of the final proposal ($|\mathcal{P}_T|$) and the simulation time as a function of N respectively. Numerical results are provided in Table X. The mean absolute error (MAE) in the estimation of the expected value of the first component, E$\{X_1\}$, averaged over 2000 runs is displayed in the first column. The second column shows the average number of pieces of the final proposal, $|\mathcal{P}_T|$. The last column provides the average simulation time, normalized w.r.t. the time required by standard ARMS with $N = 150$. Once more, the results show that IA^{2}RMS outperforms ARMS in terms of estimation accuracy, as evidenced by the lower value of MAE. Furthermore, in this case IA^{2}RMS turns out to require a lower number of linear pieces for the proposal and thus also a reduced simulation time w.r.t. ARMS.

V. CONCLUSIONS

In this work, we have introduced a new adaptive Monte Carlo technique (IA^{2}RMS) that solves an important structural limitation of the popular ARMS algorithm, which is widely used within Gibbs sampling. Unlike ARMS, IA^{2}RMS builds a sequence of self-tuned proposals that always converges to the target distribution, while keeping the computational cost bounded. As a consequence, the convergence of the chain is speeded up w.r.t. ARMS and the correlation vanishes quickly to zero. Furthermore, IA^{2}RMS effectively decouples the adaptation mechanism from the proposal building scheme, thus allowing us to reduce the complexity in the construction of the sequence of proposals. Thus, we have also proposed four simpler procedures to build the proposal densities. Indeed, IA^{2}RMS can be applied, both as a stand-alone algorithm or within any approach that requires sampling from conditional distributions (e.g., the Gibbs sampler, the hit-and-run algorithm or adaptive directiom sampling), virtually to any target distribution. Numerical results show that IA^{2}RMS outperforms ARMS, as well as other Monte Carlo approaches (Metropolis-Hastings, Slice Sampling and Multiple Try schemes) in terms of estimation accuracy, correlation and speed. Unlike other adaptive MCMC algorithms, IA^{2}RMS performs a complete adaptation of the proposal, which always converges to the target and thus guarantees the ergodicity of the chain (as proved in the theoretical study performed). A promising future line is extending the IA^{2}RMS scheme to draw directly from multivariate distributions. Regarding this issue, we note that the adaptive structure of IA^{2}RMS is valid regarless of the dimension of the target pdf, but the challenge is finding efficient procedures to build the proposal pdfs in high-dimensional spaces. In this sense, the constructions proposed in [47]–[50] for other types of methods could be used.

APPENDIX

Ergodicity of the IA^{2}RMS *Chain*

The new control test in IA^{2}RMS is performed using an auxiliary variable, y, which is always different from the new state, x_{n+1}. This approach leads to a proposal, $\pi_t(x)$, which is independent of the current state of the chain, x_n. Therefore, the convergence of the Markov chain to a bounded target density is ensured by Theorem 2 in [41] (Theorem 8.2.2 in [11]), since IA^{2}RMS satisfies the *strong Doeblin condition* if the target is bounded. The strong Doeblin condition is satisfied if, given a proposal pdf $\widetilde{\pi}_t(x)$, there exists some $a_t \in (0, 1]$, such that

$$\frac{1}{a_t}\widetilde{\pi}_t(x) \geq p_o(x) \quad \forall x \in \mathcal{X}. \tag{17}$$

In our case, since $\widetilde{\pi}_t(x) = (1/c_{\pi_t})\pi_t(x)$ and $p(x) = (1/c_p)p_o(x)$, (17) can be rewritten as

$$a_t \leq \frac{c_p}{c_{\pi_t}} \frac{\pi_t(x)}{p(x)} \quad \forall x \in \mathcal{X}. \tag{18}$$

Hence, as $\min\{1, x\} \leq x$, we could simply set

$$a_t = \min\left\{1, \frac{c_p}{c_{\pi_t}} \min_{x\in\mathcal{X}}\left\{\frac{\pi_t(x)}{p(x)}\right\}\right\}.$$

in order to fulfill (18). Furthermore, by using an appropriate construction for the tails (exponential tails for Gaussian or sub-Gaussian pdfs and Pareto pieces with the proper value of γ for heavy-tailed distributions) we can guarantee that $\pi_t(x) > p(x)$ in the tails. Thus, we can finally take

$$a_t = \min\left\{1, \frac{c_p}{c_{\pi_t}} \min_{x\in\mathcal{I}}\left\{\frac{\pi_t(x)}{p(x)}\right\}\right\}, \tag{19}$$

where $\mathcal{I} = \cup_{i=1}^{m_t}\mathcal{I}_i = \cup_{i=1}^{m_t}(s_i, s_{i+1}]$. For any of the constructions in Section III-C, the a_t in (19) satisfy all the conditions required: $a_t > 0$, $a_t \leq 1$ and $a_t \to 1$ as $t \to \infty$, since $\pi_t(x) \to p(x)$ as $t \to \infty$ (as proved in Appendix B) and thus also $c_{\pi_t} = \int_{\mathcal{X}} \pi_t(x)dx \to \int_{\mathcal{X}} p(x)dx$ as $t \to \infty$. Therefore, we can ensure that

$$\lim_{T\to\infty} \prod_{t=1}^{T}(1 - a_t) \to 0,$$

all the conditions for Theorem 2 in [41] are fulfilled, and geometric convergence is obtained for the Markov chain. Indeed, this theorem also states that the algorithm samples from the target distribution within a finite number of samples with a probability arbitrarily close to 1 if $a_t > a > 0$ *infinitely many often*. In our case, considering $a = a_0$, we can ensure that $a_t > a = a_0 > 0$ infinitely many often. This is simply due to the fact that the distance between $\pi_t(x)$ and $p(x)$ tends to decrease as t increases (see Appendix B), and thus $a_t > a_{t-1}$ in general. Consequently, we can ensure that the IA^{2}RMS algorithm samples from the target distribution within a finite number of samples with a probability arbitrarily close to 1.

Convergence of the Proposal to the Target

Here we show that $D(\pi_t, p) \to 0$ as $t \to 0$, implying that $\pi_t(x) \to 0$ as $t \to \infty$ almost everywhere. Let us consider a set of support points, $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$, with $s_1 < \ldots < s_{m_t}$, at time step t, and a continuous and bounded target $p(x)$ with bounded derivatives. Hence, by using any of the procedures described in Section III-C, the corresponding proposal function, $\pi_t(x)$, is also a bounded function. Moreover,

$$D(\pi_t, p) = \int_{\mathcal{X}} |\pi_t(x) - p(x)|\, dx \leq c_{\pi_t} + c_p < \infty,$$

since $c_{\pi_t} = \int_{\mathcal{X}} \pi_t(x)dx < \infty$ and $c_p = \int_{\mathcal{X}} p(x)dx < \infty$. Let us consider now the finite interval $\mathcal{I} = [s_1,\ s_{m_t}]$. The interpolation methods proposed in Section III-C to build $\pi_t(x)$ can be seen as an ℓ-th order Taylor approximation (e.g., $\ell = 0$ for the procedure in (7) and $\ell = 1$ for the one in (9)) inside each interval. The discrepancy between $\pi_t(x)$ and $p(x)$ over $\mathcal{I}$ is then given by

$$\begin{aligned}\int_{\mathcal{I}} |\pi_t(x) - p(x)|\, dx &= \sum_{i=1}^{m_t-1} \int_{\mathcal{I}_i} |\pi_t(x) - p(x)|\, dx \\ &= \sum_{i=1}^{m_t-1} \int_{\mathcal{I}_i} \left|r_\ell^{(i)}(x)\right| dx, \end{aligned} \tag{20}$$

where $r_\ell^{(i)}(x)$ is the remainder associated to the ℓ-th order (with $\ell \in \{0, 1\}$ in our case) polynomial approximation of $p(x)$ inside the interval $\mathcal{I}_i$, as given by Taylor's theorem. The Lagrange form of this remainder is

$$r_\ell^{(i)}(x) = \frac{(x - s_i)^{\ell+1}}{(\ell+1)!} \left.\frac{d^{\ell+1}p(x)}{dx^{\ell+1}}\right|_{x=\xi},$$

for some $\xi \in [s_i, x]$. Moreover, since $x \in \mathcal{I}_i = [s_i, s_{i+1}]$, it is straightforward to show that

$$\left|r_\ell^{(i)}(x)\right| \leq \frac{(s_{i+1} - s_i)^{\ell+1}}{(\ell+1)!} C_\ell^{(i)}, \tag{21}$$

where $C_\ell^{(i)} = \max_{x\in\mathcal{I}_i} |p^{(\ell+1)}(x)|$, and $p^{(\ell+1)}(x)$ denotes the $(\ell+1)$-th derivative of $p(x)$, i.e., $p^{(\ell+1)}(x) = d^{\ell+1}p(x)/dx^{\ell+1}$. Hence, replacing (21) in (20), we obtain

$$\sum_{i=1}^{m_t-1} \int_{\mathcal{I}_i} \left|r_\ell^{(i)}(x)\right| dx \leq \sum_{i=1}^{m_t-1} \frac{(s_{i+1} - s_i)^{\ell+1}}{(\ell+1)!} C_\ell^{(i)}. \tag{22}$$

Now, let us assume that a new point, $s' \in \mathcal{I}_k = [s_k, s_{k+1}]$ for $1 \leq k \leq m_t - 1$, is added at the next iteration. The construction of the proposal density changes only inside the interval $\mathcal{I}_k$, which can be split into $\mathcal{I}^{(1)} = [s_k, s']$ and $\mathcal{I}^{(2)} = [s', s_{k+1}]$. Then, $\max_{x\in\mathcal{I}^{(j)}} |p^{(\ell+1)}(x)| \leq \max_{x\in\mathcal{I}_k} |p^{(\ell+1)}(x)|$ with $j \in \{1, 2\}$, and $(s' - s_k)^{\ell+1} + (s_{k+1} - s')^{\ell+1} \leq (s_{k+1} - s_k)^{\ell+1}$, for any $\ell \geq 0$, since $A^{\ell+1} + B^{\ell+1} \leq (A + B)^{\ell+1}$ for any $A, B > 0$ thanks to Newton's binomial theorem.[14] Hence, the bound in (22) can never increase when a new support point is incorporated, and indeed tends to decrease (unless $\ell = 0$ and the derivative of $p(x)$ is constant everywhere), thus implying that

$$\lim_{t\to+\infty} \sum_{i=1}^{m_t-1} \int_{\mathcal{I}_i} \left|r_\ell^{(i)}(x)\right| dx = 0,$$

[14]Indeed, for $\ell \geq 1$ we have a strict inequality: $(s' - s_k)^{\ell+1} + (s_{k+1} - s')^{\ell+1} < (s_{k+1} - s_k)^{\ell+1}$.

since support points become arbitrarily close as $t \to \infty$ (i.e., $s_{i+1} - s_i \to 0$), implying that the bound on the right hand side of (22) tends to zero as $t \to \infty$. Hence, we can guarantee that $\int_{\mathcal{I}} |\pi_t(x) - p(x)|dx \to 0$ for $t \to +\infty$. Note that a monotonic decrease of the distance between $\pi_t(x)$ and $p(x)$ inside $\mathcal{I}$ cannot be guaranteed, since adding a new support point might occasionally lead to an increase in the discrepancy. However, we can guarantee that the upper bound on this distance decreases monotonically, thus ensuring that $\pi_t(x) \to p(x)$ as $t \to \infty$, i.e., adding support points will eventually take us arbitrarily close to $p(x)$.

Finally, regarding the tails, note that the distance between π_t and p remains bounded even for heavy tailed distributions. Furthermore, the interval $\mathcal{I}$ will become larger as $t \to +\infty$, since there is always a non-null probability of adding new support points inside the tails. Therefore, the probability mass associated to the tails decreases as $t \to \infty$. Hence, even though the distance between the target and the proposal may again increase occasionally due the introduction of a new support point in the tails, we can guarantee that such a distance goes to zero as t goes to infinity. Overall, this guarantees that $\pi_t(x)$ converges to $p(x)$ almost everywhere.

Discrepancy Between Normalized Proposals and Target

In this section, we prove that $D(\pi_t, p) \to 0$ as $t \to \infty$ implies $D(\widetilde{\pi}_t, p_o) \to 0$ as $t \to \infty$. This discrepancy between the normalized proposal and target densities is given by

$$D(\widetilde{\pi}_t, p_o) = \int_{\mathcal{X}} |\widetilde{\pi}_t(x) - p_o(x)|\, dx = \int_{\mathcal{X}} \left| \frac{1}{c_{\pi_t}} \pi_t(x) - \frac{1}{c_p} p(x) \right| dx. \quad (23)$$

Let us focus now on the absolute value inside the integral:

$$d(x) = \left| \left(\frac{1}{c_{\pi_t}} \pi_t(x) - \frac{1}{c_p} \pi_t(x) \right) + \left(\frac{1}{c_p} \pi_t(x) - \frac{1}{c_p} p(x) \right) \right| \leq \left| \frac{1}{c_{\pi_t}} - \frac{1}{c_p} \right| \pi_t(x) + \frac{1}{c_p} |\pi_t(x) - p(x)|, \quad (24)$$

where we have used the *triangle inequality*.

Inserting (24) into (23), we obtain

$$D(\widetilde{\pi}_t, p_o) \leq \left| \frac{1}{c_{\pi_t}} - \frac{1}{c_p} \right| \int_{\mathcal{X}} \pi_t(x) dx + \frac{1}{c_p} \int_{\mathcal{X}} |\pi_t(x) - p(x)|\, dx.$$

Finally, noting that the first integral is equal to c_{π_t}, whereas the second one is simply $D(\pi_t, p)$, we have

$$D(\widetilde{\pi}_t, p_o) \leq \frac{1}{c_p} \left[|c_p - c_{\pi_t}| + D(\pi_t, p) \right]. \quad (25)$$

Hence, since $D(\pi_t, p) \to 0$ and $c_{\pi_t} \to c_p$ as $t \to \infty$, we can guarantee that $D(\widetilde{\pi}_t, p_o) \to 0$ as $t \to \infty$, implying that $\widetilde{\pi}_t(x) \to p_o(x)$ almost everywhere.

Probability of Adding a New Support Point

The probability P_t of adding a support point at the t-th iteration is $P_t = P_{a1} + P_{a2}$, where P_{a1} and P_{a2} are the probabilities of adding points in the first and second tests, respectively. Let us also define $\mathcal{X}^+ = \{x \in \mathcal{X} : \pi_t(x) \geq p(x)\}$ and $\mathcal{X}^- = \{x \in \mathcal{X} : \pi_t(x) < p(x)\}$, such that $\mathcal{X}^+ \cup \mathcal{X}^- = \mathcal{X}$ with $\mathcal{X}^+ \cap \mathcal{X}^- = \emptyset$.

1) First Test: The probability of adding a support point in the RS test is

$$P_{a1} = \int_{\mathcal{X}^+} \left(1 - \frac{p(x)}{\pi_t(x)} \right) \widetilde{\pi}_t(x) dx = \frac{1}{c_{\pi_t}} \int_{\mathcal{X}^+} (\pi_t(x) - p(x))\, dx = \frac{1}{c_{\pi_t}} D(\pi_t, p|\mathcal{X}^+), \quad (26)$$

where $D(\pi_t, p|\mathcal{X}^+)$ denotes the L_1 distance between the unnormalized proposal, $\pi_t(x)$, and the unnormalized target, $p(x)$, with $x \in \mathcal{X}^+$.

2) Second Test: The probability of adding a new point in the second test is given by the following triple integral,

$$P_{a2} = (1 - P_{a1}) \int_{\mathcal{X}} \int_{\mathcal{X}} \int_{\mathcal{Y}} \mathrm{P}_2\{m_{t+1} = m_t + 1|y\} h(y|x, x_n) g(x_n) q_2(x) dy dx_n dx, \quad (27)$$

where we have

- $\mathrm{P}_2\{m_{t+1} = m_t + 1|y\} = \max[0, 1 - (\pi_t(y)/p(y))]$ is the probability of adding y in $\mathcal{S}_t$.
- $h(y|x, x_n) = \alpha(x, x_n)\delta(y - x_n) + [1 - \alpha(x, x_n)]\delta(y - x)$, where $\alpha(x, x_n)$ is given in Table IV.
- $g(x_n)$ is the pdf of the chain at the n-th iteration. For the sake of simplicity, let us assume that $g(x_n) = p_o(x_n) = (1/c_p)p(x_n)$, i.e., $g(x_n)$ has converged to the chain's invariant pdf, $p_o(x)$.[15]
- Since x has been already accepted in an RS test, then

$$q_2(x) = \left(\frac{1}{1 - P_{a1}} \right) \widetilde{\pi}_t(x) \min \left[1, \frac{p(x)}{\pi_t(x)} \right], = \frac{1}{c_{\pi_t}(1 - P_{a1})} \min \left[\pi_t(x), p(x) \right],$$

where $1 - P_{a1} = (1/c_{\pi_t}) \int_{\mathcal{X}} \min[p(x), \pi_t(x)] dx$.

Inserting all the previous terms in (27), P_{a2} becomes

$$\begin{aligned} P_{a2} = {} & \frac{1}{c_{\pi_t}} \int_{\mathcal{X}} \int_{\mathcal{X}} \min \left[p(x), \pi_t(x) \right] \max \left[0, 1 - \frac{\pi_t(x_n)}{p(x_n)} \right] \alpha p_o(x_n) dx_n dx \\ & + \frac{1}{c_{\pi_t}} \int_{\mathcal{X}} \int_{\mathcal{X}} \min \left[p(x), \pi_t(x) \right] \max \left[0, 1 - \frac{\pi_t(x)}{p(x)} \right] (1 - \alpha) p_o(x_n) dx_n dx \\ = {} & P_{a2}^{(1)}(x \in \mathcal{X}^+, x_n \in \mathcal{X}^+) + P_{a2}^{(2)}(x \in \mathcal{X}^-, x_n \in \mathcal{X}^+) \\ & + P_{a2}^{(3)}(x \in \mathcal{X}^-, x_n \in \mathcal{X}^+) + P_{a2}^{(4)}(x \in \mathcal{X}^-, x_n \in \mathcal{X}^-), \end{aligned}$$

where we can clearly distinguish 4 different cases. Table XI shows the 4 terms involved in the calculation of P_{a2}.

[15]Note that this assumption is only fulfilled with probability one asymptotically. However, since IA^{2}RMS fulfills Theorem 2 in [41], we can ensure that $g(x_n) = p_o(x_n)$ with a probability arbitrarily close to 1 within a finite number of samples.

TABLE XI
PROBABILITY $P_{a2} = \sum_{i=1}^{4} P_{a2}^{(i)}$ - SUMMARY

	$x \in \mathcal{X}^+$, $x_n \in \mathcal{X}^+$	$x \in \mathcal{X}^+$, $x_n \in \mathcal{X}^-$	$x \in \mathcal{X}^-$, $x_n \in \mathcal{X}^+$	$x \in \mathcal{X}^-$, $x_n \in \mathcal{X}^-$
$\alpha(x, x_n)$	1	$\frac{\pi_t(x_n)}{p(x_n)}$	1	$\min\left[1, \frac{p(x)\pi_t(x_n)}{p(x_n)\pi_t(x))}\right]$
$\min\left[p(x), \pi_t(x)\right]$	$p(x)$	$p(x)$	$\pi_t(x)$	$\pi_t(x)$
$\max\left[0, 1 - \frac{\pi_t(x)}{p(x)}\right]$	0	0	$1 - \frac{\pi_t(x)}{p(x)}$	$1 - \frac{\pi_t(x)}{p(x)}$
$\max\left[0, 1 - \frac{\pi_t(x_n)}{p(x_n)}\right]$	0	$1 - \frac{\pi_t(x_n)}{p(x_n)}$	0	$1 - \frac{\pi_t(x_n)}{p(x_n)}$
$P_{a2}^{(i)}$	$P_{a2}^{(1)} = 0$	$P_{a2}^{(2)} < \frac{c_p^+}{c_{\pi_t} c_p} D(\pi_t, p\|\mathcal{X}^-)$	$P_{a2}^{(3)} = 0$	$P_{a2}^{(4)} < \left(\frac{c_{\pi_t}^- + c_p^-}{c_{\pi_t} c_p}\right) D(\pi_t, p\|\mathcal{X}^-)$
$P_{a2} = \sum_{i=1}^{4} P_{a2}^{(i)} < \kappa_t D(\pi_t, p\|\mathcal{X}^-)$ with $\kappa_t = \frac{c_p + c_{\pi_t}^-}{c_{\pi_t} c_p}$.				

First of all, it is easy to see that $P_{a2}^{(1)} = P_{a2}^{(3)} = 0$. Moreover, for the probability $P_{a2}^{(2)}$ only the first integral is non-null. Hence,

$$\begin{aligned} P_{a2}^{(2)} &= \frac{1}{c_{\pi_t}} \int_{\mathcal{X}^+} \int_{\mathcal{X}^-} p(x) \left(1 - \frac{\pi_t(x_n)}{p(x_n)}\right) \frac{\pi_t(x_n)}{p(x_n)} p_o(x_n) dx_n dx \\ &= \frac{c_p^+}{c_{\pi_t} c_p} \int_{\mathcal{X}^-} \left(1 - \frac{\pi_t(x_n)}{p(x_n)}\right) \pi_t(x_n) dx_n \\ &< \frac{c_p^+}{c_{\pi_t} c_p} \int_{\mathcal{X}^-} \left(1 - \frac{\pi_t(x_n)}{p(x_n)}\right) p(x_n) dx_n \\ &= \frac{c_p^+}{c_{\pi_t} c_p} D(\pi_t, p|\mathcal{X}^-), \end{aligned}$$

where $c_p^+ = \int_{\mathcal{X}^+} p(x)dx$, and the inequality comes from the fact that $\pi_t(x_n) < p(x_n)$ for $x_n \in \mathcal{X}^-$. Similarly, replacing the corresponding terms shown in Table XI in (27), and taking into account that $0 \leq \alpha(x, x_n) \leq 1$,

$$\begin{aligned} P_{a2}^{(4)} &\leq \frac{1}{c_{\pi_t}} \int_{\mathcal{X}^-} \int_{\mathcal{X}^-} \pi_t(x) \left(1 - \frac{\pi_t(x_n)}{p(x_n)}\right) p_o(x_n) dx_n dx \\ &\quad + \frac{1}{c_{\pi_t}} \int_{\mathcal{X}^-} \int_{\mathcal{X}^-} \pi_t(x) \left(1 - \frac{\pi_t(x)}{p(x)}\right) p_o(x_n) dx_n dx \\ &< \frac{c_{\pi_t}^-}{c_{\pi_t} c_p} \int_{\mathcal{X}^-} \left(1 - \frac{\pi_t(x_n)}{p(x_n)}\right) p(x_n) dx_n \\ &\quad + \frac{c_p^+}{c_{\pi_t} c_p} \int_{\mathcal{X}^-} \left(1 - \frac{\pi_t(x)}{p(x)}\right) p(x) dx \\ &= \left(\frac{c_{\pi_t}^- + c_p^-}{c_{\pi_t} c_p}\right) D(\pi_t, p|\mathcal{X}^-), \end{aligned}$$

where $c_p^- = \int_{\mathcal{X}^-} p(x)dx$. Finally,

$$P_{a2} = P_{a2}^{(2)} + P_{a2}^{(4)} < \frac{c_p + c_{\pi_t}^-}{c_{\pi_t} c_p} D(\pi_t, p|\mathcal{X}^-).$$

3) Total Probability:

$$\begin{aligned} P_t &= P_{a1} + P_{a2} \\ &< \frac{1}{c_{\pi_t}} D(\pi_t, p|\mathcal{X}^+) + \frac{c_p + c_{\pi_t}^-}{c_{\pi_t} c_p} D(\pi_t, p|\mathcal{X}^-) \\ &< \left(\frac{2c_p + c_{\pi_t}^-}{c_{\pi_t} c_p}\right) D(\pi_t, p|\mathcal{X}) = \left(\frac{2c_p + c_{\pi_t}^-}{c_{\pi_t} c_p}\right) D(\pi_t, p). \end{aligned}$$

Therefore, since $\pi_t(x) \to p(x)$ almost everywhere as $t \to \infty$, then $D(\pi_t, p) \to 0$, $c_{\pi_t} \to c_p$ and $c_{\pi_t}^- \to 0$, implyting that the probability of adding support points vanishes to zero.

ACKNOWLEDGMENT

The authors would like to thank F. Leisen (University of Kent), R. Casarin (Universitá Ca' Foscari di Venezia), and J. Míguez (Universidad Carlos III de Madrid) for many useful comments and discussions on the behavior of the ARMS technique and its main drawbacks. They would also like to thank the Reviewers for their many helpful comments and suggestions.

REFERENCES

[1] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Berlin, Germany: Springer-Verlag, 2004.
[2] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Berlin, Germany: Springer-Verlag, 2004.
[3] J. K. O. Ruanaidh and W. J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*. Berlin, Germany: Springer-Verlag, 1996.
[4] W. J. Fitzgerald, "Markov chain Monte Carlo methods with applications to signal processing," *Signal Process.*, vol. 81, no. 1, pp. 3–18, January 2001.
[5] P. M. Djurić and S. J. Godsill, Eds., "Perfect sampling: A review and applications to signal processing," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 345–356, Feb. 2002.
[6] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, Sep. 2003.
[7] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 152–170, Nov. 2005.
[8] J. R. Larocque and P. Reilly, "Reversible jump MCMC for joint detection and estimation of sources in colored noise," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 231–240, Feb. 1998.
[9] J. Míguez, T. Ghirmai, M. F. Bugallo, and P. M. Djurić, "A sequential Monte Carlo technique for blind synchronization and detection in frequency-flat Rayleigh fading wireless channels," *Signal Process.*, vol. 84, no. 11, pp. 2081–2096, Nov. 2004.
[10] M. Hong, M. F. Bugallo, and P. M. Djurić, "Joint model selection and parameter estimation by population Monte Carlo simulation," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 3, pp. 526–539, Jun. 2010.
[11] F. Liang, C. Liu, and R. Caroll, *Advanced Markov Chain Monte Carlo Methods: Learning From Past Samples*. London, U.K.: Wiley Series in Comput. Statist., 2010.
[12] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, no. 1-2, pp. 5–43, January 2003.
[13] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[14] W. R. Gilks, "Derivative-free adaptive rejection sampling for Gibbs sampling," *Bayesian Statist.*, vol. 4, pp. 641–649, 1992.
[15] W. R. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *Appl. Statist.*, vol. 41, no. 2, pp. 337–348, 1992.
[16] W. Hörmann, "A rejection technique for sampling from T-concave distributions," *ACM Trans. Math. Softw.*, vol. 21, no. 2, pp. 182–193, 1995.

[17] D. Görür and Y. W. Teh, "Concave convex adaptive rejection sampling," *J. Comput. Graphical Statist.*, vol. 20, no. 3, 2011.
[18] L. Martino and J. Míguez, "Generalized rejection sampling schemes and applications in signal processing," *Signal Process.*, vol. 90, no. 11, pp. 2981–2995, November 2010.
[19] L. Tierney, "Markov chains for exploring posterior distributions," *Ann. Statist.*, vol. 22, no. 4, pp. 1701–1728, 1994.
[20] M. Davy, C. Doncarli, and J.-Y. Tourneret, "Classification of chirp signals using hierarchical Bayesian learning and MCMC methods," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 377–388, Feb. 2002.
[21] N. Dobigeon, J.-Y. Tourneret, and C.-I. Chang, "Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2684–2695, Jul. 2008.
[22] T. Elguebaly and N. Bouguila, "Bayesian learning of finite generalized Gaussian mixture models on images," *Signal Process.*, vol. 91, no. 4, pp. 801–820, 2011.
[23] M. Brewer and C. Aitken, "Discussion on the meeting on the Gibbs sampler and other Markov Chain Monte Carlo methods," *J. Roy. Statist. Soc. Series B*, vol. 55, no. 1, pp. 69–70, 1993.
[24] M. H. Chen and B. Schmeiser, "Toward black-box sampling: A random-direction interior-point markov chain approach," *J. Comput. Graphic. Statist.*, vol. 7, no. 1, pp. 1–22, 1998.
[25] P. Müller, "A generic approach to posterior integration and Gibbs sampling," Dept. Statist., Purdue Univ., West Lafayette, IN, USA, Tech. Rep. 91-09, 1991.
[26] A. E. Gelfand and T. M. Lee, "Discussion on the meeting on the Gibbs sampler and other Markov Chain Monte Carlo methods," *J. Roy. Statist. Soc. Series B*, vol. 55, no. 1, pp. 72–73, 1993.
[27] C. Fox, "A Gibbs sampler for conductivity imaging and other inverse problems," *Proc. SPIE, Image Reconstruction from Incomplete Data VII*, vol. 8500, pp. 1–6, 2012.
[28] W. R. Gilks, N. G. Best, and K. K. C. Tan, "Adaptive rejection metropolis sampling within Gibbs sampling," *Appl. Statist.*, vol. 44, no. 4, pp. 455–472, 1995.
[29] K. R. Koch, "Gibbs sampler by sampling-importance-resampling," *J. Geodesy*, vol. 81, no. 9, pp. 581–591, 2007.
[30] R. Meyer, B. Cai, and F. Perron, "Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2," *Comput. Statist. Data Anal.*, vol. 52, no. 7, pp. 3408–3423, March 2008.
[31] W. Shao, G. Guo, F. Meng, and S. Jia, "An efficient proposal distribution for Metropolis-Hastings using a b-splines technique," *Comput. Statist. Data Anal.*, vol. 53, pp. 465–478, 2013.
[32] B. Cai, R. Meyer, and F. Perron, "Metropolis-Hastings algorithms with adaptive proposals," *Statist. Comput.*, vol. 18, pp. 421–433, 2008.
[33] W. R. Gilks, R. M. Neal, N. G. Best, and K. K. C. Tan, "Corrigendum: Adaptive rejection metropolis sampling within Gibbs sampling," *Appl. Statist.*, vol. 46, no. 4, pp. 541–542, 1997.
[34] B. Cai and R. Meyer, "Bayesian semiparametric modeling of survival data based on mixtures of b-spline distributions," *Comput. Statist. Data Anal.*, vol. 55, no. 3, pp. 1260–1272, 2011.
[35] H. Putter, S. Heisterkamp, J. Lange, and F. Wolf, "A Bayesian approach to parameter estimation in HIV dynamical models," *Statist. Med.*, vol. 21, no. 15, pp. 2199–2214, 2002.
[36] L. Holden, "Adaptive chains," Norwegian Computing Center, Tech. Rep., 1998.
[37] L. Martino, J. Read, and D. luengo, "Independent doubly adaptive rejection Metropolis sampling," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2014.
[38] C. J. P. Bélisle, H. E. Romeijn, and R. L. Smith, "Hit-and-run algorithms for generating multivariate distributions," *Math. Oper. Res.*, vol. 18, no. 2, pp. 255–266, 1993.
[39] W. R. Gilks, N. G. O. Robert, and E. I. George, "Adaptive direction sampling," *The Statistician*, vol. 43, no. 1, pp. 179–189, 1994.
[40] J. Kotecha and P. M. Djurić, "Gibbs sampling approach for generation of truncated multivariate Gaussian random variables," presented at the Acoust., Speech, Signal Process. (ICASSP), 1999.
[41] L. Holden, R. Hauge, and M. Holden, "Adaptive independent Metropolis-Hastings," *Ann. Appl. Probabil.*, vol. 19, no. 1, pp. 395–413, 2009.
[42] Y. Fan, S. Brooks, and A. Gelman, "Output assessment for Monte Carlo simulations via the score statistic," *J. Comput. Graphic. Statist.*, vol. 15, pp. 178–206, 2006.
[43] L. Martino and J. Read, "On the flexibility of the design of multiple try metropolis schemes," *Comput. Statist.*, vol. 28, no. 6, pp. 2797–2823, 2013.
[44] H. Haario, E. Saksman, and J. Tamminen, "Adaptive proposal distribution for random walk Metropolis algorithm," *Comput. Statist.*, vol. 14, pp. 375–395, 1999.
[45] A. M. Ali, K. Yao, T. C. Collier, E. Taylor, D. Blumstein, and L. Girod, "An empirical study of collaborative acoustic source localization," in *Proc. Inf. Process. Sensor Netw. (IPSN07)*, Boston, Apr. 2007.
[46] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Statist. Assoc.*, vol. 446, pp. 590–599, 1999.
[47] W. Hörmann, J. Leydold, and G. Derflinger, *Automatic Nonuniform Random Variate Generation*. Berlin, Germany: Springer-Verlag, 2003.
[48] J. Leydold, "A rejection technique for sampling from log-concave multivariate distributions," *ACM Trans. Model. Comput. Simulat.*, vol. 8, no. 3, pp. 254–280, Jul. 1998.
[49] J. Leydold and W. Hörmann, "A sweep plane algorithm for generating random tuples in a simple polytopes," *Math. Comput.*, vol. 67, no. 224, pp. 1617–1635, October 1998.
[50] R. Karawatzki, "The multivariate Ahrens sampling method," Dept. Statist. Math., Tech. Rep. 30, 2006.

Luca Martino was born in Palermo (Italy) in 1980. He received the Electronic Eng. degree (M.Sc.) in 2006 from Politecnico di Milano (Italy) and the Ph.D. degree in statistical signal processing in 2011 from Universidad Carlos III de Madrid (Spain), where he was assistant professor until July 2013.

Since then he has worked as a postdoc researcher at the University of Helsinki (Aug. 2013–Jan. 2015) and University of Sao Paulo (currently). His research interests include computational statistics, Monte Carlo methods and Bayesian inference.

Jesse Read obtained his PhD in Computer Science from the University of Waikato, New Zealand, in 2010.

He then worked as a postdoc researcher in the Department of Signal Theory and Communications at the Carlos III University of Madrid until 2014. He is now in the Department of Information and Computer Science at Aalto University in Finland. His interests include machine learning (particularly in multi-label and data stream contexts), Monte Carlo methods, and neural networks.

David Luengo was born in Santander (Spain) in 1974. He received the B.Sc., M.Sc. and Ph.D. degrees in Electrical Eng. from Universidad de Cantabria (Spain) in 1995, 1998 and 2006, respectively.

From 2003 to 2011 he was assistant professor at Universidad Carlos III de Madrid (Spain). Since 2011 he is associate professor at Universidad Politecnica of Madrid (Spain). He has been a visiting scholar at the University of Illinois (Aug.–Dec. 2002) and the University of Manchester (Aug.–Nov. 2007 and Oct. 2008–July 2009). His research interests include statistical signal processing, Monte Carlo methods, multi-task learning, electrocardiographic signal processing and sparse inference.