Perfect Sampling: A Review and Applications to Signal Processing

Petar M. Djurić, *Senior Member, IEEE*, Yufei Huang, and Tadesse Ghirmai

***Abstract*—In recent years, Markov chain Monte Carlo (MCMC) sampling methods have gained much popularity among researchers in signal processing. The Gibbs and the Metropolis–Hastings algorithms, which are the two most popular MCMC methods, have already been employed in resolving a wide variety of signal processing problems. A drawback of these algorithms is that in general, they cannot guarantee that the samples are drawn exactly from a target distribution. More recently, new Markov chain-based methods have been proposed, and they produce samples that are guaranteed to come from the desired distribution. They are referred to as perfect samplers. In this paper, we review some of them, with the emphasis being given to the algorithm coupling from the past (CFTP). We also provide two signal processing examples where we apply perfect sampling. In the first, we use perfect sampling for restoration of binary images and, in the second, for multiuser detection of CDMA signals.**

***Index Terms*—CFTP, Fill's algorithm, Gibb's coupler, MCMC, perfect (exact) coupling, rejection coupler.**

I. Introduction

In THE last decade of the last century, research in computational statistics has made very significant strides. This is particularly true in the area of Monte Carlo methods, more specifically, in Markov chain Monte Carlo (MCMC) sampling [14]. The newly proposed methods could, in general, tackle problems that not long ago were considered insolvable because of their high complexity. With the increasing availability of powerful computers, these advances have brought great excitement among researchers from various backgrounds. As a result, in a rather short period of time, many books and research papers have been published [12], [14], [33]. Research in signal processing has also played an important role, and a variety of its contributions have enriched the theory and practice of MCMC sampling [2], [13], [34].[1]

In statistical signal processing, key entities are probability distributions. From Bayesian point of view, all the information that can be extracted from data about signal unknowns is contained in the posterior distribution of the unknowns. From a non-Bayesian point of view, the center of interest is usually around the likelihood of the data. A standard task of estimation when point estimates of unknowns are sought is the maximization of the posterior distribution or the likelihood with respect to the unknowns. In Bayesian signal processing, another entity of interest is the mean of the posterior. In many signal processing problems, the maximizations of the posterior distribution and the likelihood function or the evaluation of the mean of the posterior are not trivial at all. When analytical approaches fail, one resorts to numerical techniques. Most of these techniques have the objective to search the maximum of the relevant function by employing iterative techniques. The computation of the mean, on the other hand, requires evaluation of high-dimensional integrals, which has usually been done by invoking standard numerical integration techniques. For many problems, these approaches have provided good results, and they continue to be the method of choice in many applications. However, for highly complex problems, where the dimension of the unknowns is very high, these methods show their limitations.

A completely different paradigm for solving the above problems is used by MCMC sampling methods. The idea there is to generate samples from the posterior distribution or the likelihood function of interest and use them to extract relevant information or compute multidimensional integrals. The reasoning is that all that we can know about the unknowns is summarized by their posterior densities or likelihoods. Thus, if the samples can approximate the posterior densities and the likelihoods very well, we use them for inference and for carrying out the necessary computations. Consequently, a critical step of this approach is the generation of samples from a given distribution function.

The classical theory of random sample generation from simple distributions includes inversion, rejection, squeezing, and transformation methods [32]. The MCMC algorithms have the same objective, and they achieve it by using carefully constructed Markov chains. The basic MCMC method is the Metropolis algorithm [25], which was later generalized to the Metropolis–Hastings method [17]. Much of the recent research of MCMC theory was sparked by the Gibbs sampler, which is a special case of the Metropolis–Hastings method [13]. The samples in MCMC sampling are generated as a sequence of random variables $x^{(t)}$, $t = 0, 1, 2, \ldots$, according to a *transition kernel* $P(x^{(t)}|x^{(t-1)})$ of the chain. The chain is considered time-homogeneous, that is, the transition kernel does not change with time. The chain starts its trajectory at time $t = 0$ in $x^{(0)}$, and in the beginning, its states depend on $x^{(0)}$ or $P(x^{(t)}|x^{(0)}) \neq P(x^{(t)})$. However, provided the chain satisfies some regularity conditions [14], after some time, it forgets the initial condition, and $P(x^{(t)}|x^{(0)})$ converges to its unique stationary distribution. Therefore, with time, when the chain

Manuscript received January 31, 2001; revised October 9, 2001. This work was supported by the National Science Foundation under Award 9903120. The associate editor coordinating the review of this paper and approving it for publication was Prof. Simon J. Godsill.

The authors are with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: djuric@ece.sunysb.edu; yfhuang@ece.sunysb.edu; tadesse@ece.sunysb.edu).

Publisher Item Identifier S 1053-587X(02)00561-5.

[1]An Internet site with many preprints on MCMC sampling can be found at http://www.statslab.cam.ac.uk/~mcmc/.

converges, the samples produced by the chain look increasingly like samples from the stationary distribution. Then, we say that the chain is in *equilibrium* and that its samples are drawn from the stationary distribution. The period from $t = 0$ until convergence is known as *burn-in* period, and the samples from this period are always thrown away.

An important practical problem in applications is the determination of the burn-in period. Typically, the chain is started from an arbitrary state; it is run for some burn-in time, which is believed to be long enough for the chain to have converged, and from then on, the generated samples are assumed as if they are truly samples from the stationary distribution. For assessing convergence, there are various diagnostics [4], [6], [24], but none of them guarantees that the chain has indeed converged and that the generated samples are truly samples from the desired distribution. In fact, since the chains are always run for finite time, the samples are usually only approximate.

This weakness of the MCMC notwithstanding, research in this area proceeded unabated. There was a strong reason for such activity; MCMC methods led to successful resolutions of many highly complex problems. In 1996, a new Markov chain-based algorithm for a generation of random samples was proposed, with a distinct feature that its samples were perfect (exact), i.e., the samples were exactly from the stationary distribution of interest [30]. The algorithm uses a clever scheme for determining the burn-in period, that is, the time at which the Markov chain has converged to its equilibrium. It exploits an important tool in probability known as *coupling*; and therefore, the method is called *coupling from the past* (CFTP). Initially, it was developed for discrete distributions with finite number of states, but later, it was extended to allow for sampling from continuous state spaces [29]. An alternative approach to the CFTP method was proposed in 1998, and it is referred to as Fill's perfect rejection sampling algorithm [9]; as its name suggests, it is based on rejection sampling. In contrast to CFTP, it can be interrupted at any time during the simulation, for example, due to too-long runs, without introducing any bias to the generated samples.

In a brief time, work on further developments of the proposed methods has picked up, and the whole field, now known as *perfect sampling*, became very popular.[2] The purpose of this paper is to review some of the basic theory and to present some of its relevant developments. A goal too is to bring the area of perfect sampling closer to the signal processing community, and to that end, we present two examples of applications of perfect sampling in signal processing. First, we review the CFTP algorithm for discrete state spaces, then briefly describe Fill's interruptible algorithm, and finally explain the CFTP algorithm for continuous state spaces. Several important concepts are also described in detail. They include chain coupling, which plays a crucial role in detecting the moments when the Markov chains have converged, and monotone and anti-monotone chains, which allow for easy implementation of the perfect sampling schemes. One of the signal processing examples is on restoration of binary images by a sandwiched CFTP scheme, and its performance is shown in several scenarios. In the second example, multiuser detection of CDMA signals is carried out, where a perfect sampling scheme, called the Gibbs coupler, is used to draw samples from the posterior of the transmitted symbols.

The paper is organized as follows. In Section II, we describe the basic CFTP algorithm and the main concepts associated with it. In Section III, some extensions to CFTP are provided including the Gibbs coupler, and in Section IV, Fill's interruptible algorithm is presented. Perfect sampling from continuous state spaces is reviewed in Section V. Finally, in Section VI, we illustrate perfect sampling-based methods with examples, and in Section VII, we make a few concluding remarks.

A. Notation and Definitions

Before we proceed, we set some of the notation and recall a few important definitions. The state space on which a Markov chain is defined is denoted by $\mathcal{S}$. If the state space is discrete and finite, $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$. The state of the chain at time t is denoted by $x^{(t)}$. We assume that the Markov chains are irreducible, recurrent, and aperiodic. A Markov chain is irreducible if $P(x^{(t)} = s_i | x^{(0)} = s_j) > 0$ for any states of the chain s_i and s_j. It is recurrent if the expected number of visits of the chain to any state of the state space is infinite. Finally, the chain is aperiodic if for some i, the greatest common divider GCD satisfies

$$\text{GCD}\left\{T > 0 \colon P(x^{(t+T)} = s_i | x^{(t)} = s_i) > 0\right\} = 1.$$

The Markov chains are assumed to have a unique stationary distribution denoted by π.

II. CFTP Algorithm

A. Coupling

The CFTP algorithm, which is developed by Propp and Wilson [30], allows for perfect (exact) and independent sampling from a desired distribution. The sampling is implemented by running ergodic Markov chains whose stationary distribution is the desired distribution. The underlying concept of the approach involves running coupled Markov chains that start from all the possible initial states. Once all the chains meet, i.e., *coalesce*, they follow the same path, which implicates that the effect of initial states is worn off. A critical tool of the method is *coupling*, which is an important probability theory concept. We illustrate it with a simple example (a similar example appears in [35]).

Example 1: Consider a queue that can hold only three packets. Assume that at a given time slot, one of the following three possibilities occurs.

1) Only one packet arrives at the queue.
2) One or two packets leave the queue.
3) No packet enters or leaves the queue.

The probabilities for a packet arrival, departure of one packet, and departure of two packets are 0.4, 0.4, and 0.2, respectively.

The state of the queue can be represented by a Markov chain with a state space $\mathcal{S} = \{s_1, s_2, s_3, s_4\} = \{0, 1, 2, 3\}$. The state diagram of the Markov chain is shown in Fig. 1. Clearly, it is straightforward to compute various parameters of the queue such as its blocking probability (the probability that it is in

[2]An excellent source of information on the subject can be found on Internet at http://dimacs.rutgers.edu/~dbwilson/exact.html.

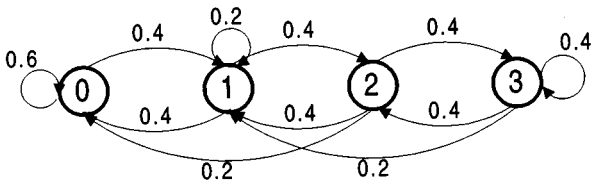

Fig. 1. State diagram of the Markov chain in Example 1.

state 3) or the probability of the queue being in any other state. One can compute the stationary probabilities of the chain π by solving π from $\pi\mathbf{P} = \pi$, where $\mathbf{P}$ is the transition matrix of the chain [8]. Here, however, we pretend that we cannot compute these probabilities using an analytical approach and that instead, we have to resort to a simulation method based on perfect sampling. The idea is to draw a large number of samples from the stationary distribution of the chain and estimate the desired probabilities.

We need to run four coupled Markov chains starting at all the states of $\mathcal{S}$. A very important component of every perfect sampling scheme is its updating function, which must ensure coalescence. The updating function is a *random map* that specifies the next state of the chain as a function of the current state and random numbers. The specification is done according to the transition kernel of the chain. Commonly, for reasons of convenience, the random numbers denoted by R are generated from a uniform distribution on $(0, 1)$. For our example, we define the updating function $\Phi(\cdot, \cdot)$ as (1) shown at the bottom of the page, where $R^{(t+1)}$ is a random number drawn at time $t + 1$. Obviously, the updating function (1) is derived from the Markov chain in Fig. 1 and can be expressed in a different way.

Fig. 2 shows the trajectories followed by all the chains, which started from every state of $\mathcal{S}$. It is important to note that at time $t + 1$, all the chains use the same random number $R^{(t+1)}$ to make the transition from $x^{(t)}$ to $x^{(t+1)}$. Therefore, we say that the chains are *coupled*. As can be seen, the trajectories merge at $t = 4$ and thereafter follow the same path. *Coalescence* occurs because the chains use the same updating rule as shown in (1), and the same random numbers for transitions. In this example, the time progresses from present to the future, and thus, the coupling is called forward coupling.

As indicated earlier, if coalescence occurs, the effect of the initial state is worn off, which seems to entail that the state at which coalescence occurs is a valid sample from the desired distribution. However, this is incorrect, and such scheme would yield biased samples. Bias arises because the time of coalescence does not occur at a fixed time but, rather, at random times [35]. For the example considered above, coalescence never occurs at the state $s_3 = 2$. Thus, for the states of $\mathcal{S}$,

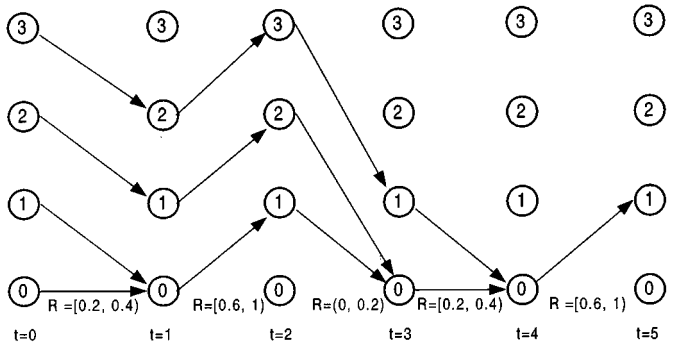

Fig. 2. Coupling of the Markov chain in Example 1.

samples drawn using forward coupling produce a distribution where $P(s_3) = 0$, whereas the correct stationary distribution is $\pi = (14/35,\ 11/35,\ 6/35,\ 4/35)$.

B. Coupling From the Past

The concept of CFTP introduces a simple but important modification to the forward coupling so that the produced samples of the modified scheme are perfect. The basic difference between the two types of coupling is that with CFTP the Markov chains are run from the past ($t < 0$) to the present ($t = 0$), and most importantly, the samples are always drawn at a fixed time, i.e., at $t = 0$, provided coalescence has occurred at $t = 0$ or earlier.

Now, we explain how CFTP is implemented. Consider an ergodic Markov chain with a discrete and finite state space of size N. If N copies of the chain are run, where each copy corresponds to a different initial state, the chains will eventually coalesce and will be stationary by time $t = 0$. In that case, the value of the chains at $t = 0$ is a perfect sample from the stationary distribution. Obviously, it is not feasible to run chains from the infinite past, and instead, one can use the simple and clever scheme known as CFTP. All the N chains are started at $T = -1$ and are checked for coalescence at $t = 0$. If coalescence occurred, the state of the chain at $t = 0$ is accepted as a sample from the desired distribution. Otherwise, the starting time is moved back to $T = -2$, and the chains are evolved and again checked for coalescence at $t = 0$. If coalescence took place by $t = 0$, the state at $t = 0$ is a sample from the target distribution, and if not, the starting time is moved further back to $T = -3$. The whole procedure is repeated, and a sample is drawn if coalescence occurs or the starting time is shifted further back. This process continues until coalescence occurs.

There are two very important points for the implementation of the CFTP scheme.

1) In the attempts to achieve coalescence, for the transitions from $t = k$ to $t = k + 1$, where $k < 0$, one uses

$$x^{(t+1)} = \Phi\left(x^{(t)}, R^{(t+1)}\right) = \begin{cases} 0, & \text{if } R^{(t+1)} \in [0, 0.2) \wedge x^{(t)} \in \{0, 1, 2\} \\ 1, & \text{if } R^{(t+1)} \in [0, 0.2) \wedge x^{(t)} = 3 \\ \max(x^{(t)} - 1, 0), & \text{if } R^{(t+1)} \in [0.2, 0.4) \\ x^{(t)} & \text{if } R^{(t+1)} \in [0.4, 0.6) \wedge x^{(t)} \in \{0, 1\} \\ x^{(t)} - 1 & \text{if } R^{(t+1)} \in [0.4, 0.6) \wedge x^{(t)} \in \{2, 3\} \\ \min(x^{(t)} + 1, 3) & \text{if } R^{(t+1)} \in [0.6, 1] \end{cases} \tag{1}$$

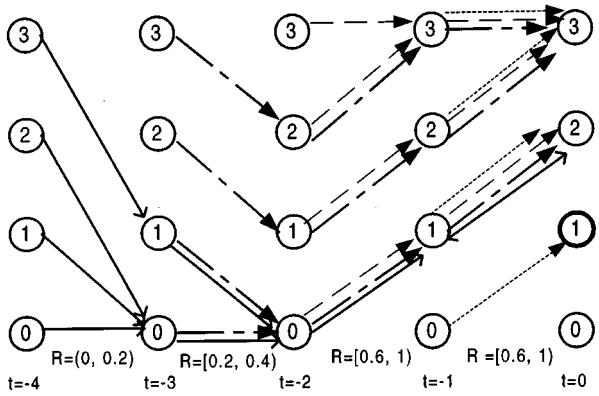

Fig. 3. CFTP of the Markov chain in Example 1. Dotted lines: Trajectories started at $T = -1$; thin broken lines: Trajectories started at $T = -2$; thick broken lines: Trajectories started at $T = -3$; thick solid lines: Trajectories started at $T = -4$.

the same random number. For example, let $R^{(0)}$ be the random number used in the transition from $T = -1$ to $t = 0$ in the first step. In the second step, the starting time is $T = -2$, and there are two transitions. Nevertheless, only one new random number is generated [$R^{(-1)}$], and it is used for the transition from $T = -2$ to $t = -1$. For the transition from $t = -1$ to $t = 0$, the "old" random number $R^{(0)}$ from the first step is applied. Therefore, in the above scheme, every time the starting time is moved back by one time unit, only one new random number is generated [$R^{(T+1)}$], and it is used in the first transition. For the remaining transitions, the old random numbers are reused.

2) The samples are drawn at $t = 0$ only, even if coalescence occurred earlier.

We applied the CFTP algorithm to our example, and a case of successful drawing of a sample is shown in Fig. 3. As depicted, coalescence did not take place for the starting times $T = -1, -2$ and -3. The trajectories of the chains coalesced when they were started at $T = -4$, and the coalescence occurred at $t = -2$. The trajectory was then completed at $t = 0$, and $x = 2$ was the drawn sample.

In summary, the CFTP algorithm can be described in pseudo code form as follows:

```
CFTP(T):
T ← −1
Repeat
  R^(T+1) ~ U(0, 1)
  S^(T) ← S
  for t = T, T + 1, ..., −1
    S^(t+1) ← Φ(S^(t), R^(t+1))
  T ← T − 1
Until S^(0) is singleton.
```

Next, we provide a heuristic argument why the CFTP algorithm returns a perfect sample in finite time. In the sequel, we also present a formal mathematical proof of it. Let T be a starting time of the CFTP algorithm for which the trajectories coalesce by $t = 0$. In addition, suppose that the Markov chain is run from the infinite past. At time T, the chain is in equilibrium, and its state at T, $x^{(T)}$ is a perfect sample from the stationary distribution. The chain remains in equilibrium beyond T, and the state at $t = 0$, $x^{(0)}$ is also a perfect sample. Note that we do not know the state of the chain at T, and that it does not really matter because from $t = T$ to $t = 0$, the state space of the chain remains a singleton set. Therefore, we can argue that the sample $x^{(0)}$ is a result of infinite time simulation, and consequently, it is a perfect sample from the distribution π. The method can thus be referred to as virtual simulation from time $-\infty$ as it allows us to sample an infinitely long simulation by reconstructing it over a finite time interval [7], [22], [36]. Now, we state the theorem and present its proof [30].

Theorem 1: Let an ergodic Markov chain have a stationary distribution π. Then, a) the CFTP algorithm returns a sample in finite time with probability one, and b) the returned sample is an exact sample from the stationary distribution π.

Proof: The proof basically rests on second Borel–Cantelli lemma [8]. A similar proof has been presented in [5]. Since the Markov chain is irreducible and aperiodic, we can find a finite $L > 0$ such that

$$L = \min\{t\colon P(x^{(t)} = z | x^{(0)} = y) > 0\}, \quad \forall\, y, z \in \mathcal{S}.$$

It then follows that each chain has a positive probability of being in any state at $t = L > 0$. Define the event $C_k = \{$ The N chains coalesce in $(-kL, -(k-1)L)\}$. Thus, $P(C_k) > \epsilon$, where $\epsilon > 0$. Moreover, the C_ks are independent because coalescence in $(-kL, -(k-1)L)$ depends only on the random numbers generated within the interval that are independent from the random numbers generated outside the interval and does not depend on the initial states. Thus, since

$$\sum_{k=1}^{\infty} P(C_k) = \infty$$

we conclude by Borel–Cantelli lemma that P(infinitely many C_k occur) $= 1$. The second part of the theorem is a consequence of the fact that if CFTP finds T to be the starting time from which all the trajectories of the chain coalesce by $t = 0$, the returned value of the chain at $t = 0$ is the same as if the chain was run from $t = -\infty$. Hence, this value is a sample from the target distribution π. ◊

In the CFTP algorithm given above, the values of $T = -1, -2, -3, \ldots$ are successively taken as starting times of the chains. This, however, is not necessary. In fact, any decreasing sequence of T is a valid set of starting times of the algorithm. Propp and Wilson recommended the sequence of starting times given by $T_i = -2^{i-1}$ $(i = 1, 2, 3, \ldots)$, which double at every step [30]. This choice minimizes the worst-case number of required simulation steps and almost minimizes the expected number of steps. The proof of this claim can be found in [30].

C. Monotonicity

The CFTP algorithm as described above is difficult to monitor and is computationally very intensive for problems where the Markov chains have large state spaces. Therefore, its practicality is quite limited. In some important applications, it may be

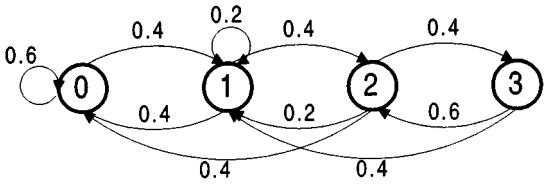

Fig. 4. State diagram of the Markov chain in Example 2.

possible to define an updating function that possesses the property of monotonicity for a partial order imposed on the state space $\mathcal{S}$. As a result, instead of propagating large number of chains, we would work with only two chains. Assume that there is a partial order $x \preceq y$[3] on the state space $\mathcal{S}$ for $x, y \in \mathcal{S}$. Then, a mapping function is called monotone if it satisfies $\Phi(x, R) \preceq \Phi(y, R), \ \forall R$ when $x \preceq y$. Now, denote with $x^{\max}$ and $x^{\min}$ the maximum and minimum elements of the state space of $\mathcal{S}$, respectively, with the partial order $x^{\min} \preceq x \preceq x^{\max}, \ \forall x$. Then, if there exists a monotone updating function $\Phi(\cdot, R)$ for the Markov chain, the use of such a transition rule ensures preservation of the same order in all subsequent paths. Therefore, when applying the CFTP algorithm, it is only necessary to monitor the two chains whose starting states are $x^{\max}$ and $x^{\min}$ since all the other chains are always sandwiched between them.

Suppose that we impose the partial order $s_1 \prec s_2 \prec s_3 \prec s_4$ in Example 1. Then, it can be easily verified that the updating function in (1) is monotone, that is, $\Phi(0, R) \preceq \Phi(1, R) \preceq \Phi(2, R) \preceq \Phi(3, R)$. Thus, when applying the CFTP scheme to Example 1, we need to monitor only the chains that start at $s_1 = 0$ and $s_4 = 3$.

In some situations, where a monotone transition rule cannot be found, it may still be possible to monitor only two paths. This can be achieved by the crossover method [15], [21], which can be applied if the transition rule is antimonotone. A transition rule $\Phi(\cdot, R)$ is referred to as antimonotone if, for a partial order $x \preceq y$, the transition rule satisfies $\Phi(x, R) \succeq \Phi(y, R)$. We illustrate this property with the following example.

Example 2: Consider Example 1 with the modifications presented by Fig. 4.

We may define an antimonotone transition rule, shown in (2) at the bottom of the page. Suppose that on the state space of the Markov chain, we impose a partial order $s_3 \prec s_2 \prec s_1 \prec s_4$. Then, it can easily be verified that the transition rule defined by (2) is antimonotone. Thus, the CFTP algorithm can be applied to this chain by keeping track of the trajectories that start only from states $s_3 = 2$ and $s_4 = 3$. As a final note, CFTP algorithms that exploit monotonicity or antimonotonicity of chains are called sandwiched CFTP algorithms.

[3]A set $\mathcal{S}$ is partially ordered if and only if it admits a transitive ordering relation $x \preceq y$ between some pairs of elements $x, \ y$ so that $x \preceq y, \ y \preceq z$ implies $x \preceq z$.

III. On Some Further Developments of the CFTP Algorithm

Here, we address three interesting extensions of the CFTP algorithm. The first is on a modification of the CFTP that allows for running the Markov chain forward in time [37]. The second is the Gibbs coupler method, which represents an implementation of CFTP on binary state spaces and combines the CFTP scheme with Gibbs sampling. Finally, the third extension is the method known as dominated CFTP, which provides perfect samples of some point processes.

A. Read-Once CFTP

The original CFTP requires that one keep track of seeds of the random number generator. This may entail either frequent regeneration of the random samples $R^{(t)}$ (which sets additional time requirements to the system that implements the CFTP) or storing the random samples $R^{(t)}$ (which requires additional memory for storage). It turns out that perfect samples can be obtained by coupling from the past without reusing the "old" random numbers, that is, by running a read-once stream of random numbers [37].

Before we describe the implementation of the read-once CFTP method, recall the following. First, if we apply forward coupling in the usual way as shown in Section II-B, at some point, coalescence occurs. Second, as already explained, we cannot assume that the state where coalescence takes place is a perfect sample (recall that bias arises in forward coupling). Third, once we know that a sample is perfect, the following samples of the chain are also perfect but not independent. For the identification of the independent samples that follow a perfect sample on the trajectory of the Markov chain, this procedure runs another Markov chain, which is independent from the first chain.

With these remarks, we return to the read-once CFTP procedure and note that it is very important that it is initialized properly. The initialization is carried out by running two independent chains using forward coupling. When coalescence of the second chain occurs, say, at time t_1, the coalescence of the first chain is checked. If the first chain has already coalesced by t_1, the value of the chain at t_1, for example s_i, is stored. This value is a candidate for a perfect and independent sample. If the first chain has not coalesced by time t_1, the procedure is repeated until the first chain has coalesced before the second chain. When this occurs, the initialization is completed.

$$x^{(t+1)} = \Phi(x^{(t)}, R^{(t+1)}) = \begin{cases} 0, & \text{if } R^{(t+1)} \in [0, 0.4) \wedge x^{(t)} \in \{0, 1, 2\} \\ 1, & \text{if } R^{(t+1)} \in [0, 0.4) \wedge x^{(t)} = 3 \\ x^{(t)}, & \text{if } R^{(t+1)} \in [0.4, 0.6) \wedge x^{(t)} \in \{0, 1\} \\ 2, & \text{if } R^{(t+1)} \in [0.4.1] \wedge x^{(t)} = 3 \\ 3, & \text{if } R^{(t+1)} \in [0.4, 0.8) \wedge x^{(t)} = 2 \\ x^{(t)} + 1, & \text{if } R^{(t+1)} \in [0.6, 1] \wedge x^{(t)} \in \{0, 1\} \\ 1, & \text{if } R^{(t+1)} \in (0.8, 1) \wedge x^{(t)} = 2 \end{cases} \tag{2}$$

After initialization, the forward coupling procedure of the two chains is again invoked and run independently in parallel. In addition, the trajectory of the first chain started out at $x^{(t_1)} = s_i$ is tracked. When the second chain coalesces, for instance at t_2, coalescence of the first chain is also checked. If at t_2 the first chain has coalesced, say, at s_k, the candidate s_i is a perfect sample. Note that s_i but *not* s_k is the perfect sample. However, s_k is a candidate for a perfect sample in the next run. It will be a perfect sample (with probability 1/2) if the first chain coalesces before the second chain after they are both restarted at t_2 and run independently. If the first chain does not coalesce before t_3, at which the second chain coalesced, s_k is not accepted as a perfect sample. Instead, it is thrown away, and the two chains are restarted at t_3. The sample $x^{(t_3)} = s_l$ of the trajectory of the first chain that went through $x^{(t_2)} = s_k$ is the new candidate for a perfect sample.

In terms of memory and time requirements, the read-once CFTP in general is comparable with the standard CFTP. In some applications, however, the read-once CFTP is faster than the standard CFTP. Another important advantage of the read-once CFTP is that it requires only one sequence of pseudorandom numbers, whereas the standard CFTP needs many such sequences. These sequences have to be independent, which is a requirement that is difficult to achieve. In fact, the random sequences in practice may be quite correlated, and that affects the quality of the "perfect" samples.

B. Gibbs Coupler

Recently, a new coupling algorithm has been proposed [16], [18], [19], which we refer to as the Gibbs coupler. An advantage of the Gibbs coupler over the standard CFTP method is that it can be applied to large state spaces without the need for monotone or antimonotone properties of the updating function. In [16] and [19], specific algorithms are proposed for Markov random fields such as k-colorings, the hard-core model, and the Potts model. In the following, we describe a general algorithm for high-dimensional binary spaces. The coupling method of the Gibbs coupler, as opposed to the CFTP method, is component based. Suppose that the state of the chain $x^{(t)}$ has binary representation given by $x_1^{(t)}, x_2^{(t)}, \ldots, x_M^{(t)}$, and let $\mathcal{S}^{(t)} = \{\mathcal{S}_1^{(t)}, \mathcal{S}_2^{(t)}, \ldots, \mathcal{S}_M^{(t)}\} \in \{-1, 1\}^M$ denote the support of $x^{(t)}$, where $\mathcal{S}_i^{(t)}$, $i = 1, 2, \ldots, M$ represents the support of the component $x_i^{(t)}$. An important ingredient of the method is the concept of sandwich distributions. They are defined for every t and $i = 1, 2, \ldots, M$ by

$$L_i^{(t)}(x_i = 1) = \min_{x_{-i}^{(t)} \in \mathcal{S}_{-i}^{(t)}} \left\{ P\left(x_i = 1 \,\middle|\, x_{-i}^{(t)}\right) \right\} \tag{3}$$

and

$$U_i^{(t)}(x_i = 1) = \max_{x_{-i}^{(t)} \in \mathcal{S}_{-i}^{(t)}} \left\{ P\left(x_i = 1 \,\middle|\, x_{-i}^{(t)}\right) \right\} \tag{4}$$

where $\mathcal{S}_{-i}^{(t)} = \{\mathcal{S}_1^{(t)}, \ldots, \mathcal{S}_{i-1}^{(t)}, \mathcal{S}_{i+1}^{(t-1)}, \ldots, \mathcal{S}_M^{(t-1)}\}$ is the collection of supports of $x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \ldots, x_M^{(t)}$. The update of $\mathcal{S}_i^{(t)}$ is then carried out according to

$$\begin{aligned} \mathcal{S}_i^{(t)} &= \Phi\left(\mathcal{S}_{-i}^{(t)}, R_i^{(t)}\right) \\ &= \begin{cases} \{1\}, & \text{if } R_i^{(t)} \le L_i^{(t)}(x_i = 1) \\ \{-1\}, & \text{if } R_i^{(t)} \ge U_i^{(t)}(x_i = 1) \\ \{-1, 1\}, & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

where, as before, $R_i^{(t)}$ is a random number drawn from the uniform density on $(0, 1)$.

Coalescence occurs if the supports of all the $\mathcal{S}_i$s are singletons. As in the CFTP method, if coalescence occurs at $t = 0$, the singleton supports $\mathcal{S}^{(0)} = \{\mathcal{S}_1^{(0)}, \ldots, \mathcal{S}_M^{(t)}\}$ define the generated sample. It can be proved that the sandwich distributions in (3) and (4) achieve the largest probability of coalescense, and hence, their use leads to the fastest coalescense. It can also be shown that for problems where monotonicity exists, the Gibbs coupler has the same rate of coalescense as CFTP and that it is equivalent to the sandwiched CFTP algorithm.

In summary, the basic Gibbs coupler is given by the following pseudo-code:

```
Gibbs coupler(T):
  t ← -T
  while t < 0
    t ← t + 1
    i ← 0
    while i <= M
      update S_i^(t) using p(x_i | x_1^(t), ..., x_{i-1}^(t), x_{i+1}^(t), ..., x_N^(t))
      for all x_j ∈ S_j^(t) with j = 1,2,...N and j ≠ i
  if size of all S_i^(0) for i = 1,2,...M is equal to 1
then
    return(S^(0))
  else
    Gibbs coupler(2T).
```

C. Dominated CFTP

Dominated coupling from the past (DCFTP) is another method that extends the use of CFTP [20], [21]. The CFTP method as defined is only applicable to uniformly ergodic Markov chains. However, if the chains are geometrically ergodic, which is true for Markov chains that converge to point process distributions, the CFTP may not produce samples at all. To explain the method, we modify Example 1. A similar example is also given in [35].

Example 3: Consider a queue where packets arrive at a rate $\lambda(N(t)+1)/(N(t)+2)$ and leave at a rate $N(t)$, where $N(t) \in \mathbb{N}_0$, and where the queue may be of infinite size.

The objective is to sample $N(t)$, which is the number of packets in the queue. Note that the process $N(t)$ is not time-homogeneous, and thus, CFTP cannot be applied directly. DCFTP overcomes the problem of time inhomogeneity by using a process $D(t)$ that is time homogeneous and dominates $N(t)$, i.e., $D(t) \ge N(t)$. For the above example, $D(t)$ could be a process with a packet arrival rate λ and a departure rate $D(t)$. The process $D(t)$, therefore, can be straightforwardly sampled. To use $D(t)$ for sampling $N(t)$, the two processes must be coupled since we must have $D(t) \ge N(t)$. The coupling is achieved as follows: Every arrival of a packet in $D(t)$ is also an arrival in $N(t)$ with probability $(N(t-)+1)/(N(t-)+2)$, and every departure in $D(t)$ is a departure in $N(t)$ with probability $N(t-)/D(t-)$ (note that the notation $t-$ means $t - \epsilon$, where ϵ is an arbitrarily small positive number). There are no arrivals or departures in $N(t)$ if there are no associated arrivals or departures in $D(t)$. Since $N(t)$ is bounded by $D(t)$ and 0, once

the value of $D(T)$, $T < 0$ is known, only two chains of $N(t)$ are started: one at $N(T) = 0$ and the other at $N(T) = D(T)$. If the two chains coalesce at $t = 0$, the obtained value comes from the distribution of $N(t)$. If they do not coalesce, two new chains are started at $D(2T)$ and 0, and so on. For more details on DCFTP, see [20], [21], and [35].

IV. Fill's Interruptible Algorithm

In the CFTP algorithm, the running time T and the returned sample $x^{(0)}$ are dependent random variables. It is very important that the algorithm is run until it returns a sample. If, however, the algorithm is interrupted and restarted whenever T is long, the generated samples will be biased in favor of those samples that take less time to generate.

Fill developed an algorithm for generation of perfect samples that can be interrupted at any time, and yet, the samples that are generated are not biased [9]. This is because in Fill's algorithm, the running time of the algorithm and the returned values are independent. Fill's algorithm is similar to CFTP in that it uses Markov chains to produce perfect samples, but it is based on the idea of rejection sampling instead of ion the concept of coupling.

The rejection sampler first takes a proposal x from a distribution $f(\cdot)$, and then accepts x as a sample from a target distribution π with probability $\pi(x)/Lf(x)$, where

$$L \geq \sup_x \frac{\pi(x)}{f(x)}.$$

To illustrate the application of rejection sampling in Fill's algorithm, consider an ergodic Markov chain with a stationary distribution π. Assume that for the Markov chain x, we can define a monotone updating function $\Phi(x^{(t)}, R^{(t+1)})$, and for the time-reversed Markov chain, a corresponding updating function $\tilde{\Phi}(x^{(t)}, \tilde{R}^{(t-1)})$, where $x^{(t-1)} = \tilde{\Phi}(x^{(t)}, \tilde{R}^{(t-1)})$. Then, the partially ordered state-space $\mathcal{S}$ possesses a unique minimum state $x^{\min}$ and a unique maximum state $x^{\max}$. Furthermore, let $P(\cdot\,|\,\cdot)$ and $\tilde{P}(\cdot\,|\,\cdot)$ denote the transition kernels of the two chains, respectively. Recall that time reversibility is defined by $P(y|x)\pi(x) = \tilde{P}(x|y)\pi(y)$, or more generally, $P^{(t)}(y|x)\pi(x) = \tilde{P}^{(t)}(x|y)\pi(y)$, where $P^{(t)}(y|x)$ denotes the probability of a transition from x to y in t steps. This definition in conjunction with the monotonic property implies

$$\frac{\pi(x)}{P^{(t)}(x|x^{\min})} = \frac{\pi(x^{\min})}{\tilde{P}^{(t)}(x^{\min}|x)} \leq \frac{\pi(x^{\min})}{\tilde{P}^{(t)}(x^{\min}|x^{\max})}. \quad (6)$$

Fill's algorithm applies rejection sampling by choosing $P^{(t)}(\cdot\,|x^{\min})$ as a proposal density and sets

$$L = \frac{\pi(x^{\min})}{\tilde{P}^{(t)}(x^{\min}|\,x^{\max})}$$

according to (6). Thus, a proposal x from $P^{(t)}(\cdot\,|\,x^{\min})$ is accepted as a sample from π with probability

$$\frac{\pi(x)}{L \times P^{(t)}(x|x^{\min})} = \frac{\tilde{P}^{(t)}(x^{\min}|x^{\max})}{\tilde{P}^{(t)}(x^{\min}|x)}. \quad (7)$$

In realizing the above rejection sampler, Fill's algorithm starts from the minimal state of the Markov chain and makes T transitions, where T is a prespecified value. The value of the chain after T transitions is a candidate for a perfect sample. This sample is accepted if a second chain started from the maximal state and coupled with the time-reversed chain arrives at the minimal state. Otherwise, the sample is rejected, and a new run with doubled T is restarted. More specifically, we can write the pseudo code of the algorithm as in the following.

```
Fill(T):
  t ← 0
  x^(0) ← x^min
  y^(T) ← x^max
  while t < T
    t ← t + 1
    x^(t) ← Φ(x^(t−1), R^(t))
    draw R̃^(T−t) ~ P(R̃ | Φ(x^(t−1), R^(t)) = x^(t))
    y^(T−t) ← Φ̃(y^(T−t+1), R̃^(T−t))
  If y^(0) = x^min then
    return x^(T) as a perfect sample from π
  else
    Fill(2T).
```

From the above pseudo code, it is easy to see that $x^{(T)}$ is generated as a proposal from $P^{(T)}(\cdot|x^{\min})$. Therefore, the returned sample $x^{(T)}$ will be a perfect sample from π as long as $P(y^{(0)} = x^{\min}|y^{(T)} = x^{\max}, x^{(0)} = x^{\min}, x^{(T)}) = \tilde{P}^{(T)}(x^{\min}|x^{\max})/\tilde{P}^{(T)}(x^{\min}|x^{(T)})$, which is demonstrated by the following lemma, which we state without proof.

Lemma: If $y^{(t)}$ is a realization of the reversed monotonic Markov chain constructed as in Fill's algorithm, then

$$P(y^{(0)} = x^{\min}|y^{(T)} = x^{\max}, x^{(0)} = x^{\min}, x^{(T)}) = \frac{\tilde{P}^{(T)}(x^{\min}|x^{\max})}{\tilde{P}^{(T)}(x^{\min}|x^{(T)})}.$$

Fill's original algorithm is designed on finite state spaces and requires the monotonicity on the transition kernel to be applicable. Later, it was extended to infinite state spaces [35] as well as to antimonotonic systems with infinite state spaces [28]. The algorithm was also generalized to apply to nonmonotonic and continuous systems [11].

V. Perfect Sampling From Continuous State Spaces

The protocol of CFTP may also be used to construct perfect sampling algorithms for continuous state spaces. However, direct use of CFTP is prohibitive or impossible because of the uncountable infinite number of states in the state space. It can be impractical to implement the procedure, and/or the chains will not coalesce in finite time. To circumvent the problem, a logical approach is to attempt a design that is based on discretization of the continuous space. In this section, we restrict our attention to algorithms for bounded continuous spaces. The extension to unbounded spaces is done by dominating techniques that are discussed in Section III-C.

In [29], Murdoch and Green have developed several perfect sampling algorithms for continuous state spaces. They include the *multigamma coupler*, the *rejection coupler*, and the *Metropolis–Hastings coupler*. A common feature of these methods is a transition scheme that updates a subset of the support into a single state. When the union of the subsets equals

the whole support, discretization of the continuous space is accomplished.

Here, we explain the rejection coupler and thereby demonstrate the basic idea. Suppose that $p(x|y)$ represents the transition kernel of a Markov chain, where $x, y \in \mathcal{S}$. In addition, suppose also that $p(x|y) = cg(x|y)$, where c is a normalizing constant and where instead of $p(x|y)$, $g(x|y)$ is known. We also assume that we can determine an upper bounding function $h(x)$ on $g(x|y)$ such that $g(x|y) \leq h(x), \ \forall y \in \mathcal{S}$ and that we can sample from the density $h(x)/v$, where $v = \int h(x)\,dx$. Therefore, given the state y at time t, a new state at time $t+1$ is drawn from $h(x)/v$ as a proposal density, and the generated sample x is accepted if

$$R < \frac{g(x|y)}{h(x)} \tag{8}$$

where $R \sim U(0, 1)$. We further notice that for the drawn x and R, any state y satisfying (8) would accept the same state x. Therefore, by solving (8) w.r.t. y when x and R are given, we can identify a subset of $\mathcal{S}$ that shrinks to a single state x. This is the basic idea of the discretization process, which in general represents coupling of infinite number of states.

The rejection coupler operates as follows. It starts at $T < 0$ with support $\mathcal{S}^{(T)} = \mathcal{S}$. During the update of $\mathcal{S}^{(T)}$, first, a sample and a random number are generated from $h(\cdot)/v$ and $U(0, 1)$, respectively, and second, given the generated sample and the random number, a subset of $\mathcal{S}^{(T)}$ is determined from (8). The procedure is repeated until the union of all the produced subsets equals $\mathcal{S}^{(T)}$. All the samples x that were drawn from $h(\cdot)/v$ represent the support $\mathcal{S}^{(T+1)}$. This support is discrete and is recorded together with the used random numbers. If, at time 0, the number of states in $\mathcal{S}^{(0)}$ is 1, this single state is returned as a perfect sample, and the algorithm terminates. Otherwise, the algorithm is restarted at time $-2T$, and it reuses the recorded random numbers for the updates from $t = T$ to $t = 0$.

For problems with large dimensions, Murdoch and Green proposed componentwise perfect sampling algorithms [29]. The main idea of componentwise updating is to reduce the complexity of sampling from high-dimensional spaces by applying a sequence of lower dimensional samplings. These methods inherit the framework of the Gibbs sampler while using the perfect sampling algorithms discussed above as building blocks. Another componentwise method for sampling from continuous distributions was proposed in [27].

VI. Applications in Signal Processing

A. Restoration of Binary Images

Consider the restoration of a degraded $N \times N$ binary image, whose pixels $\mathbf{x}$ are either black ($+1$) or white (-1). Suppose that the original image is contaminated by Bernoulli noise. More specifically, the Bernoulli noise reverses independently and with probability p the value of each pixel to its complement. The likelihood of observing $\mathbf{y}$ when $\mathbf{x}$ is the true image can then be expressed as

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \prod_{i=1}^{N\times N} p^{I(x_i \neq y_i)}(1-p)^{I(x_i = y_i)} \\ &= p^{(N\times N)}\{(1-p)/p\}^{\sum_i^{N\times N} I(x_i = y_i)} \end{aligned} \tag{9}$$

where $I(\cdot)$ is an indicator function. Furthermore, since the images in reality are often composed of relatively smooth pieces, a smooth prior is chosen as follows:

$$p(\mathbf{x}) \propto \exp\left(\beta \sum_{\langle i, j\rangle} x_i x_j\right) \tag{10}$$

where $\langle i, j\rangle$ indicates all the neighboring pixel pairs in the image that are adjacent in either the vertical or the horizontal directions, and $\beta > 0$. The coefficient β determines the degree of the smoothness in that the larger the value of β, the smoother the prior. In fact, the prior in (10) is also a simplified Ising model, which is a useful Markov random field (MRF) model. Now, the posterior distribution is readily expressed as

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \\ &\propto \exp\left(\beta \sum_{\langle i, j\rangle} x_i x_j + \tfrac{1}{2}\log((1-p)/p) \sum_{i}^{N\times N} x_i y_i\right). \end{aligned} \tag{11}$$

To obtain an estimate of the true image, various Bayesian estimators can be used including the maximum *a posteriori* (MAP) and the marginal posterior mode (MPM) estimators [38]. These estimators involve evaluation of the posterior distribution, which for large N is a computationally prohibitive task. Alternatively, efficient perfect-sampling algorithms like the sandwiched CFTP can be applied to simulate the posterior distribution. Recall that the sandwiched CFTP requires a monotonic Markov chain. For this problem, the monotonic chain can be realized by the Gibbs sampler. The Gibbs sampler takes samples from the full conditional distributions which can be obtained from (11) as

$$\begin{aligned} &p(x_u = 1|\mathbf{x}_{-u}, \mathbf{y}) \\ &\quad = \left\{1+\exp\left(-2\beta \sum_{\langle i, u\rangle} x_i + \tfrac{1}{2}\log((1-p)/p) y_u\right)\right\}^{-1} \end{aligned} \tag{12}$$

and $p(x_u = -1|\mathbf{x}_{-u}, \mathbf{y}) = 1 - p(x_u = 1|\mathbf{x}_{-u}, \mathbf{y})$, where $u = 1, 2, \ldots, N \times N$, and $\mathbf{x}_{-u}$ denotes a vector of all the components in $\mathbf{x}$, except x_u. Accordingly, the updating function of the Gibbs sampler is

$$\begin{aligned} x_u^{(t+1)} &= \phi(\mathbf{x}_{-u}^{(t)}, R_u^{(t)}) \\ &= \begin{cases} 1, & \text{if } R_u^{(t)} < p(x_u = 1|\mathbf{x}_{-u}^{(t)}, \mathbf{y}) \\ -1, & \text{otherwise} \end{cases} \end{aligned} \tag{13}$$

where $R_u^{(t)}$ is a sample from $U(0, 1)$. Now, we impose partial order $\succeq$ on the $N \times N$ binary state space such that $\mathbf{x} \succeq \tilde{\mathbf{x}}$ if $x_i = 1$ whenever $\tilde{x}_i = 1, \ \forall i$. We can see that the distribution (12) is an increasing function of $\mathbf{x}_{-u}$ on the partial order. This is equivalent to saying that $\phi(\mathbf{x}_{-u}, R) \succeq \phi(\tilde{\mathbf{x}}_{-u}, R)$, and the Markov chain so constructed is monotonic. Consequently, two extreme states on the state space can easily be determined; one is $\mathbf{x}^{\max}$ when all the pixels are black, and the other is $\mathbf{x}^{\min}$

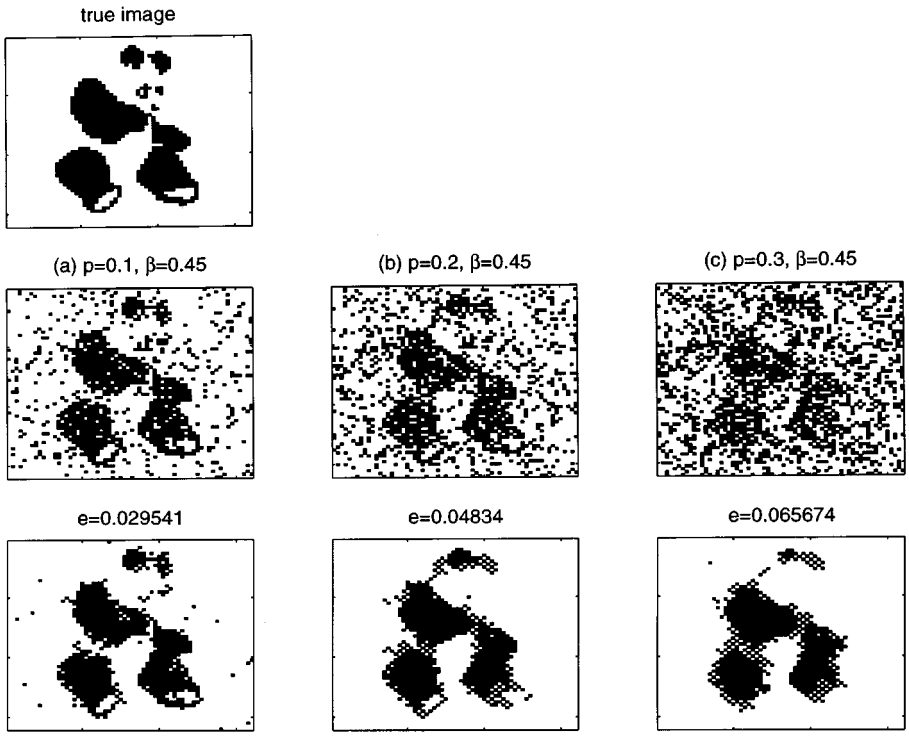

Fig. 5. Image in the top row is a 64 × 64 binary image of a panda. The three images in the second row show the degraded image by Bernoulli noises with different value of p. The MPM restorations of the degraded images are plotted in the third row. Misclassification rates e are also provided for each restored image.

when they are all white. Now, we are able to implement the sandwiched CFTP by tracing two chains started from the two extreme states and checking for coalescence at $t = 0$.

We show the performance of the sandwiched CFTP method on a 64 × 64 binary image of a panda. Different values were chosen for p in the three experiments. However, in all the cases, $\beta = 0.45$, and 1000 samples from the posterior distribution were collected. To recover the true image, MPM estimates were computed by using the obtained samples. In Fig. 5, the true image, degraded images, and the corresponding restored images are displayed. The misclassification rates in each experiment are also provided. The rates are defined as the ratio of incorrectly restored pixels and the total number of pixels.

B. Multiuser Detection

Optimum multiuser detection is the optimum solution to recovering data transmitted by many users in a code-division multiple-access (CDMA) system [36]. In this subsection, we present a perfect sampling approach to an approximate solution of optimum multiuser detection.

A K-user synchronous CDMA white Gaussian channel can be modeled as [36]

$$y(t) = \sum_{k=1}^{K} A_k b_k s_k(t) + n(t), \qquad t \in [0, T] \tag{14}$$

where

$y(t)$	received signal;
$s_k(t)$	antipodal signature waveform of the kth user;
A_k	amplitude of the kth user's signal;
$b_k \in \{-1, 1\}$	bit transmitted by the kth user;
$n(t)$	additive white Gaussian noise with zero mean and variance σ^2;
T	symbol duration.

We assume here that all the parameters except the b_ks are known, and our objective is to estimate $\mathbf{b}^{\mathrm{T}} = [b_1 b_2 \cdots b_K]$. From a Bayesian perspective, the optimum estimate of $\mathbf{b}$ is the set that maximizes the posterior distribution. The solution leads to the maximum *a posteriori* (MAP) detector. Now, if noninformative prior is chosen for $\mathbf{b}$, the posterior distribution of $\mathbf{b}$ is

$$\begin{aligned} &p(\mathbf{b}|y(t), t \in [0, T]) \\ &\quad \propto \exp\left(-\frac{1}{2\sigma^2}\int_0^T \left[y(t) - \sum_{k=1}^{K} A_k b_k s_k(t)\right]^2 dt\right) \\ &\quad \propto \exp\left(\frac{1}{2\sigma^2}\left(2\sum_{k=1}^{K} A_k y_k b_k - \sum_{k=1}^{K}\sum_{l=1}^{K} A_k A_l \rho_{kl} b_k b_l\right)\right) \end{aligned} \tag{15}$$

where $\rho_{kl} = \int_0^T s_k(t) s_l(t)\, dt$ represents the crosscorrelation between the kth and the lth signature waveform, and $y_k = \int_0^T s_k(t) y(t)\, dt$ is the kth matched filter output. Here, we show how an efficient perfect sampling algorithm can be applied to calculate the output of the MAP detector.

First of all, due to the multidimensionality of the variable space, the Gibbs sampler is particularly preferable for constructing the Markov chain. The full conditional distributions

required by the Gibbs sampler are readily derived from the posterior distribution (15) as

$$p(b_i = 1 | \mathbf{b}_{-i}, y(t), t \in [0, T]) = \left[1+\exp\left(\frac{2}{\sigma^2}\left(-A_i y_i + \sum_{k=1, k\neq i}^{K} A_i A_k \rho_{ik} b_k\right)\right)\right]^{-1} \quad (16)$$

for $i = 1, 2, \ldots K$. Now, if we let $\beta_{ik} = A_i A_k \rho_{ik}$, we would notice that when $\beta_{ik} < 0 \ \forall i$ and k, the distribution (16) is equivalent to (12). Thus, using analogous reasoning as in Section VI-A, we can show that in this problem, there is a monotonic Markov chain only when $\beta_{ik} < 0 \ \forall i, k$. Since the amplitudes A_i of the signals are assumed positive, the condition of $\beta_{ik} < 0$ also implies negative crosscorrelation, or $\rho_{ik} < 0$. Therefore, an efficient sandwiched CFTP can only be applied if the crosscorrelations in the CDMA system are only negative. For problems with arbitrary crosscorrelations, an efficient perfect sampling solution is provided by the Gibbs coupler [18].

An important issue of the Gibbs coupler, as pointed out in Section III, is the specification of sandwich distributions on the full conditional distributions. They are obtained by maximizing and minimizing (16) on the support of $\mathbf{b}_{-i}$. Here, the maximum and the minimum of (16) with respect to $\mathbf{b}_{-i}$ can easily be determined by only checking the sign of β_{ik}. It thus follows that at time t, the sandwich distributions are

$$L_i^{(t)}(b_i = 1) = \left[1+\exp\left(\frac{2}{\sigma^2}\left(-A_i y_i + \sum_{k\in\mathbf{I}_{i1}^{(t)}} |\beta_{ik}| + \sum_{k\in\mathbf{I}_{i2}^{(t)}} \beta_{ik} b_k^{(t)}\right)\right)\right]^{-1}$$

and

$$U_i^{(t)}(b_i = 1) = \left[1+\exp\left(\frac{2}{\sigma^2}\left(-A_i y_i - \sum_{k\in\mathbf{I}_{i1}^{(t)}} |\beta_{ik}| - \sum_{k\in\mathbf{I}_{i2}^{(t)}} \beta_{ik} b_k^{(t)}\right)\right)\right]^{-1}$$

where $\mathbf{I}_{i1}^{(t)} \subset \{1, 2, \ldots, i-1, i+1, \ldots, K\}$ contains the indices of the elements of $\{b_k^{(t)}\}_{k=1, k\neq i}^{K}$ that have not coalesced at t, and $\mathbf{I}_{i2}^{(t)} \subset \{1, 2, \ldots, i-1, i+1, \ldots, K\}$ are the indices of the remaining elements of the set $\{b_k^{(t)}\}_{k=1, k\neq i}^{K}$ that have coalesced at t. According to the algorithm then, at any t, the support of the ith component is updated according to (5), and the coalesced state at $t = 0$ is recorded as a perfect sample from the posterior (15).

Suppose that the desired number of samples is N. Once N samples are acquired, the MAP estimate is computed. There are several approaches to find the MAP estimate. For instance, the posterior probability of each perfect sample is computed, and the MAP estimate is the one that has the largest posterior probability. As an alternative, one can consider the samples of each b_i, for $i = 1, 2, \ldots, N$ independently. In this case, the MAP estimate of b_i is set to be the sample that appears most frequently.

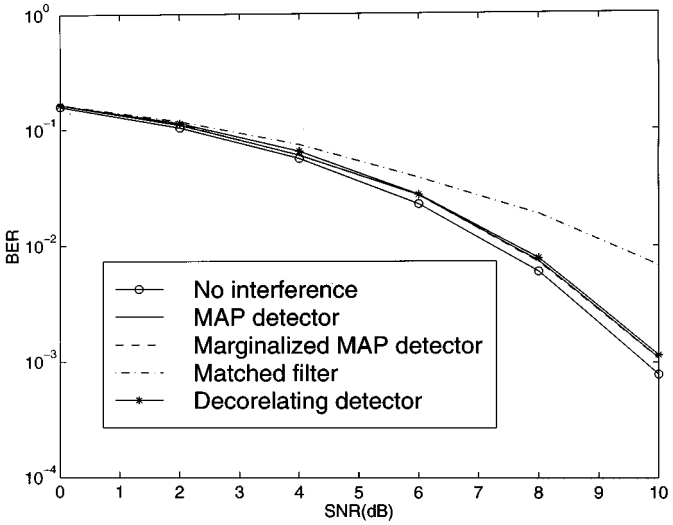

Fig. 6. BER of the different methods on a seven-user-equal-power system as a function of SNR.

This procedure yields the marginalized MAP estimate of the transmitted bits.

We present the results of several experiments that demonstrate the performance of the MAP detector. In the first two experiments, we chose a seven-user system whose common crosscorrelation of the spreading code was 0.1. Apparently, there is no monotonic Markov chain for this system. To compute a MAP estimate, 50 samples were collected. In the first experiment, bit-error rates (BERs) of both the MAP and the marginalized MAP detectors of the first user were examined under different SNRs. The results are illustrated in Fig. 6. The results of the conventional matched filter and the decorrelating detector [36] on the same user are also presented. The theoretical performance in the absence of multiuser interference is plotted as a lower bound. In order to compute the BER at a specific SNR for any detector, Monte Carlo trials were performed. The number of Monte Carlo trials was precomputed by assuring that with the BER of the lower bound at this SNR, there would be at least 300 errors among the trials. We see that the curves corresponding to the MAP and the marginalized MAP detectors are closely intertwined, which indicates similar performance of these detectors. However, the BERs of the single-user matched filter and the decorrelating detector are larger than those of the two MAP detectors, which is more explicitly shown at higher SNRs.

In the second experiment, we introduced a near–far effect into the system where different energy levels were assigned to the seven users. Specifically, the three largest energy levels were 5 dB higher, and the three middle energy levels were 3 dB higher than the smallest energy level, respectively. BERs of the MAP and the marginalized MAP detectors of the user with the smallest energy are plotted versus SNR in Fig. 7. Again, we notice that the two MAP detectors have very similar performance. For comparison, the results of the single-user matched filter and the decorrelating detector are also presented. As expected, the performance of the matched filter degrades dramatically, whereas, due to resistance to near-far effect [36], the performance of the decorrelating detector remains the same as that in the first experiment. However, at every tested SNR, the two

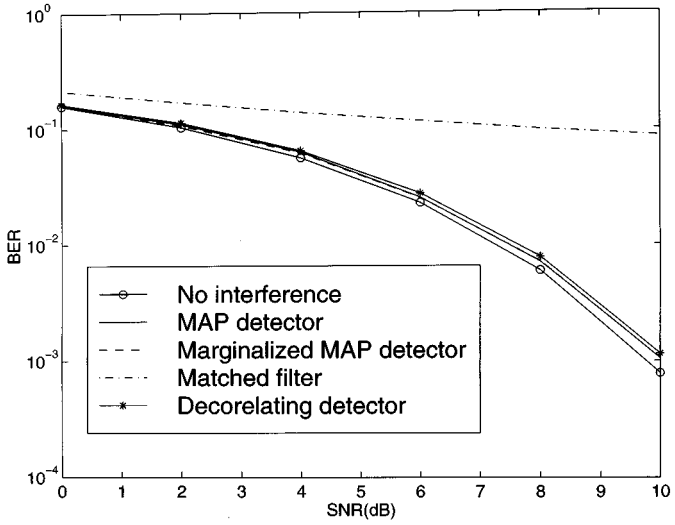

Fig. 7. BER of the user with smallest amplitude level as a function of SNR. There were seven users with different amplitude levels.

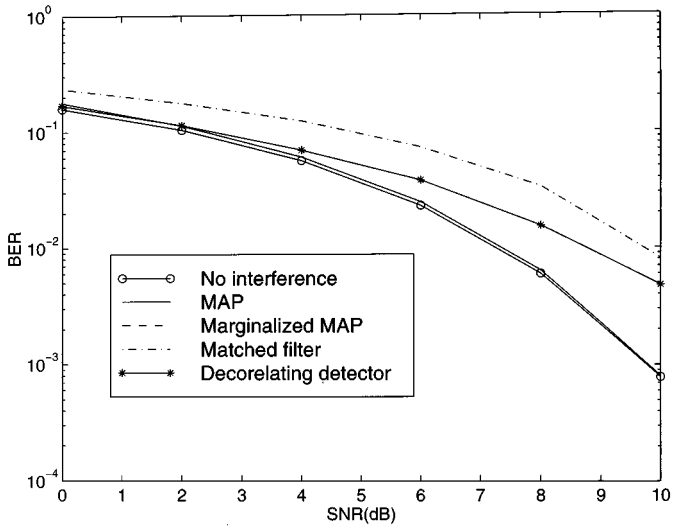

Fig. 8. BER of the different methods on a 15-user-equal-power system as a function of SNR.

MAP detectors outperform both the decorrelating detector and the matched filter. Furthermore, the BERs of the two MAP detectors in this experiment are also smaller than those in the first experiment. This is a notable feature of the optimum detectors, and it is due to the increased energy level of the other users. Note that the matched filter and the decorrelating detector do not have this feature. This observation indicates that the two MAP detectors by the Gibbs coupler correctly represent the optimum detectors.

Finally, in the third experiment, a more realistic scenario was simulated, in which a 31-bit Gold sequence was used as the spreading code for a 15-user system. We assumed that all users were with equal power. To compute a MAP estimate for this experiment, 200 samples were used. The BERs of the MAP detectors, the matched filter, and the decorrelating detector were calculated on the system, and the results are depicted in Fig. 8. We can see that the MAP detectors have a similar performance, which is very close to the lower bound. On the other hand, for higher SNRs, the single-user matched filter and the decorrelating detector performed rather poorly.

VII. Conclusions

In this paper, we reviewed algorithms for perfect sampling and showed examples of its use in signal processing. The emphasis in the review was given to CFTP and some of its extensions as well as to Fill's interruptible algorithm. It should be noted that there are also other Markov chain based approaches to perfect sampling [1], [3], [23], [26], [31]. A new interruptible perfect sampler not based on Markov chains and called the randomness recycler is presented in [10]. Many of the perfect sampling methods, however, are not very practical, and therefore, their impact on many applied fields has been limited. This is particularly true when sampling has to be performed on continuous state spaces. In our examples, for instance, some of the variables, like the Bernoulli probability p and the Gaussian noise variance σ^2, which are both continuous variables, were assumed known. In practice, this is rarely the case, but here, we had to assume them as known because it would have been very difficult to obtain perfect samples jointly with the available algorithms. Another important weakness of the current perfect algorithms is that they cannot meet deadlines. This implies that in theory, they cannot be used for sequential signal processing. In the near future, this may change, and with advances in the theory and practice in the field, perfect sampling algorithms may become a standard tool for researchers in many disciplines. One such discipline will certainly be statistical signal processing.

Acknowledgment

The authors thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] D. Aldous, "On simulating a Markov chain stationary distribution when transition probabilities are unknown," in *Discrete Probability and Algorithms*, D. Aldous, P. Diaconis, J. Spencer, and J. M. Steele, Eds. New York: Springer-Verlag, 1995, vol. 72, pp. 1–9.

[2] C. Andrieu, P. M. Djurić, and A. Doucet, "Model selection by Markov chain Monte Carlo computations," *Signal Process.*, pp. 19–37, 2001.

[3] S. Asmussen, P. W. Glynn, and H. Thorisson, "Stationary detection in the initial transient problem," *ACM Trans. Modeling Comput. Simul.*, vol. 2, no. 2, pp. 130–157, 1992.

[4] S. Brooks and G. Roberts, "Assessing convergence of Markov chain Monte Carlo algorithms," *Statist. Comput.*, pp. 319–335, 1999.

[5] G. Casella, M. Lavine, and C. Robert, "Explaining the perfect sampler,", 2000, to be published.

[6] M. Cowles and B. Carlin, "Markov chain Monte Carlo convergence diagnostics: A comparative study," *J. Amer. Statist. Assoc.*, vol. 91, pp. 883–904, 1996.

[7] X. K. Dimakos, "A guide to exact simulation," *Int. Statist. Rev.*, vol. 69, no. 1, pp. 27–48, Apr. 2001.

[8] W. Feller, *An Introduction to Probability Theory and Its Applications*. New York: Wiley, 1968.

[9] J. A. Fill, "An interruptible algorithm for perfect sampling via Markov chains," *Ann. Appl. Probab.*, vol. 8, no. 1, pp. 131–162, 1998.

[10] J. A. Fill and M. Huber, "The randomness recycler approach to perfect sampling," in *Proc. 53rd Session Int. Statist. Inst.*, 2001.

[11] J. A. Fill, M. Machida, D. J. Murdoch, and J. S. Rosenthal, "Extension of Fill's perfect rejection sampling algorithm to general chains (extended abstract)," in *Monte Carlo Methods*, N. Madras, Ed. New York: Amer. Math. Soc., 2000, vol. 26, pp. 37–52.

[12] D. Gamerman, *Markov Chain Monte Carlo*. New York: Chapman & Hall, 1997.

[13] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, 1984.
[14] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. New York: Chapman & Hall, 1996.
[15] O. Häggström and K. Nelander, "Exact sampling from anti-monotone systems," *Statistica Neerlandica*, vol. 52, pp. 360–380, 1998.
[16] ——, "On exact simulation of Markov random fields using coupling from the past," *Scand. J. Statist.*, vol. 26, pp. 395–411, 1999.
[17] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their application," *Biometrika*, vol. 57, pp. 97–109, 1970.
[18] Y. Huang and P. M. Djurić, "Multiuser detection of synchronous CDMA signals by the Gibbs coupler," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, 2001.
[19] M. Huber, "Exact sampling and approximate counting techniques," in *Proc. 30th ACM Symp. Theory Comput.*, 1998, pp. 31–40.
[20] W. S. Kendall, "Perfect simulation for spatial point processes," in *Bulletin of the International Statistical Institute 51st Session*, vol. 3, Istanbul, August 1997, pp. 163–166.
[21] ——, "Perfect simulation for the area-interaction point process," in *Probability Toward 2000*. New York: Springer, 1998, pp. 218–234.
[22] W. S. Kendall and E. Thönnes, "Perfect simulation in stochastic geometry," *Pattern Recogn.*, vol. 32, no. 9, pp. 1569–1586, 1999.
[23] L. Lovász and P. Winkler, "Exact mixing in an unknown Markov chain," *Electron. J. Combinator.*, 1995.
[24] K. Mengersen, C. Robert, and C. Guihenneuc-Jouyaux, "MCMC convergence diagnostics: A review," in *Bayesian Statistics 6*, J. Bernardo, J. Berger, A. Dawid, and A. Smith, Eds. Oxford, U.K.: Oxford Univ. Press, 1999.
[25] N. Metropolis, A. W. Rosenblath, M. N. Rosenblath, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1091, 1954.
[26] A. Mira, J. Moller, and G. O. Roberts, "Perfect slice samplers," *J. R. Statist. Soc.*, ser. B, vol. 63, no. 3, pp. 593–606, 2001.
[27] J. Møller, "Perfect simulation of conditionally specified models," *J. R.Statist. Soc.*, ser. B, vol. 61, no. 1, pp. 251–264, 1999.
[28] J. Møller and K. Schladitz, "Extensions of Fill's algorithm for perfect simulation," *J. R. Statist. Soc.*, ser. B, vol. 61, no. 4, pp. 955–969, 1999.
[29] D. J. Murdoch and P. J. Green, "Exact sampling from a continuous state space," *Scand. J. Statist.*, vol. 25, no. 3, pp. 483–502, 1998.
[30] J. G. Propp and D. B. Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics," *Random Struct. Alg.*, vol. 9, no. 1 & 2, pp. 223–252, 1996.
[31] ——, "How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph," *J. Algorithms*, vol. 27, pp. 170–217, 1998.
[32] B. D. Ripley, *Stochastic Simulation*. New York: Wiley, 1987.
[33] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer, 1999.
[34] J. J. K. Ó Ruanaidh, *Numerical Bayesian Methods Applied to Signal Processing*. New York: Springer-Verlag, 1996.
[35] E. Thönnes, "A primer on perfect simulation,", 1999, to be published.
[36] S. Verdu, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[37] D. B. Wilson, "How to couple from the past using a read-once source of randomness," *Random Struct. Alg.*, vol. 16, no. 1, pp. 85–113, 2000.
[38] G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. New York: Springer-Verlag, 1995.

Petar M. Djurić (SM'99) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1981 and 1986, respectively, and the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, in 1990.

He is currently a Professor with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook. He works in the area of statistical signal processing, and his primary interests are in the theory of modeling, detection, estimation, and time-series analysis and its applications to a wide variety of disciplines, including telecommunications, biomedicine, and power engineering.

Prof. Djurić has served on numerous technical committees for the IEEE and SPIE and has been invited to lecture at universities in the United States and overseas. He has been Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and currently is the Vice Chair of the IEEE Signal Processing Society Committee on Signal Processing—Theory and Methods. He is Treasurer of the IEEE Signal Processing Conference Board and a member of both the American Statistical Association and the International Society for Bayesian Analysis.

Yufei Huang was born in Shanghai, China, in 1973. He received the B.S. degree in applied electronics from Northwestern Polytechnic University, Xi'an, China, in 1995 and the M.S. and Ph.D. degrees in electrical engineering from the State University of New York (SUNY) at Stony Brook, in 1997 and 2001, respectively.

He is now working as a post-doctoral researcher with the Department of Electrical and Computer Engineering, SUNY at Stony Brook. His current research interests are in the theory of Monte Carlo methods and their applications to array processing and multiuser communications.

Tadesse Ghirmai was born in 1966 in Asmara, Eritrea. He received the B.Sc. degree in electrical engineering from Addis Ababa University, Addis Ababa, Ethiopia, in 1988 and the M.S.E.E. degree from University of South Florida, Tampa, in 1995. He is currently pursuing the Ph.D. degree at the State University of New York at Stony Brook.

He worked at the University of Asmara from 1988 to 1993 and from 1995 to 1999 as an Assistant Lecturer and as a Lecturer, respectively. His research interests are in the areas of statistical signal processing and communications.