Our results use algebraic geometry codes based on elliptic curves. In many ways, the elliptic codes are very similar to the Reed–Solomon codes. Intuitively we expect that the decoding problem for elliptic codes is the easiest among all algebraic geometry codes. We leave it as an open problem to prove that both problems are NP-hard for codes based on curves of any fixed genus. We conjecture that the maximum-likelihood decoding is NP-hard even for a family of algebraic geometry codes with a fixed alphabet, and leave it as an open problem.

ACKNOWLEDGMENT

The author would like to thank Daqing Wan and Elizabeth Murray for helpful discussions.

REFERENCES

[1] L. Adleman, "Algorithmic number theory—The complexity contribution," in *Proc. 35th IEEE Symp. Found. Comp. Sci.*, 1994, pp. 88–113.
[2] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *Proc. 33 th ACM Symp. Theory Comput.*, 2001, pp. 601–610.
[3] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 3, pp. 384–386, 1978.
[4] Q. Cheng and D. Wan, "On the list and bounded distance decodibility of the Reed-Solomon codes (extended abstract) (FOCS)," in *Proc. 45th IEEE Symp. Found. Comput. Sci.*, 2004, pp. 335–341.
[5] I. Dumer, D. Micciancio, and M. Sudan, "Hardness of approximating the minimum distance of a linear code," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 22–37, 2003.
[6] S. Goldwasser and J. Kilian, "Primality testing based on elliptic curves," *J. ACM*, vol. 46, no. 4, pp. 450–472, 1999.
[7] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of Reed-Solomon codes is NP-hard," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2249–2256, 2005.
[8] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, 1999.
[9] D. R. H. Brown, "The differences between consecutive primes," *J. London Math. Soc.*, vol. 2, no. 18, pp. 7–13, 1978.
[10] M.-D. Huang and D. Ierardi, "Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve," *J. Symb. Comput.*, vol. 18, pp. 519–539, 1994.
[11] H. W. Lenstra, "Factoring integers with elliptic curves," *Ann. Math.*, vol. 126, pp. 649–673, 1987.
[12] A. Lobstein, "The hardness of solving subset sum with preprocessing," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 943–946, 1990.
[13] R. Pellikaan, B.-Z. Shen, and G. J. M. van Wee, "Which linear codes are algebraic-geometric," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 583–602, 1991.
[14] J. H. Silverman, *The arithmetic of elliptic curves*. New York: Springer-Verlag, 1986.
[15] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1757–1766, 1997.

Computation of Information Rates by Particle Methods

Justin Dauwels, *Member, IEEE*, and
Hans-Andrea Loeliger, *Fellow, IEEE*

***Abstract*—Prior work on the computation of information rates of channels with memory is extended to continuous state spaces by means of sequential Monte-Carlo integration ("particle filtering").**

***Index Terms*—Continuous channels with memory, information rate, particle filtering, sequential Monte-Carlo integration.**

I. INTRODUCTION

We consider the problem of computing the information rate

$$I(X;Y) \triangleq \lim_{n\to\infty} \frac{1}{n} I(X_1,\ldots,X_n;Y_1,\ldots,Y_n) \tag{1}$$

between the input process $X = (X_1, X_2, \ldots)$ and the output process $Y = (Y_1, Y_2, \ldots)$ of a time-invariant discrete-time channel with memory. Let $x_k^n \triangleq (x_k, x_{k+1}, \ldots, x_n)$ and $x^n \triangleq (x_1, x_2, \ldots, x_n)$. We will assume that there is an ergodic stochastic process $S = (S_0, S_1, S_2, \ldots)$ such that

$$p(x^n, y^n, s_0^n) = p(s_0) \prod_{k=1}^{n} p(x_k, y_k, s_k | s_{k-1}) \tag{2}$$

for all $n > 0$ and with $p(x_k, y_k, s_k|s_{k-1})$ not depending on k.

For *finite* input alphabet $\mathcal{X}$ (= range of X_k) and *finite* state space $\mathcal{S}$ (= range of S_k), a practical method for computing the information rate (1) was proposed in [1]–[3]. This method was generalized in [4]–[7] to the computation of upper and lower bounds on the information rate of more general channels. An alternative approach to compute approximations of (1) was presented in [8]. An extension to 2-D channels (using generalized belief propagation [9]) was proposed in [10].

In this correspondence, we extend the methods of [1] and [4] to continuous state spaces $\mathcal{S}$. For the sake of clarity, we will assume that $\mathcal{S}$ is a bounded subset of $\mathbb{R}^\nu$, the ν-dimensional Euclidean space. The input alphabet $\mathcal{X}$ may also be continuous. The key to this extension is the use of sequential Monte-Carlo integration methods ("particle filters") [11], [12].

This correspondence is structured as follows. In Section II, we review the basic idea of [1]. In Section III, we show how particle methods allow to deal with a continuous state space. Two numerical examples are given in Section IV: two channels with phase noise, where the phase noise has memory.

Manuscript received December 16, 2006; revised October 7, 2007. The work of J. Dauwels was supported in part by Swiss National Science Foundation Grant 200021-101955. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, June /July2004.

J. Dauwels was with the Department of Information Technology and Electrical Engineering, ETH Zürich, CH-8092 Zurich, Switzerland. He is now with the RIKEN Brain Science Institute, Saitama 351-0106, Japan (e-mail: justin@dauwels.com).

H.-A. Loeliger is with the Department of Information Technology and Electrical Engineering, ETH Zürich, CH-8092 Zurich, Switzerland (e-mail: loeliger@isi.ee.ethz.ch).

Communicated by V. A. Vaishampayan, Associate Editor At Large.

Digital Object Identifier 10.1109/TIT.2007.911181

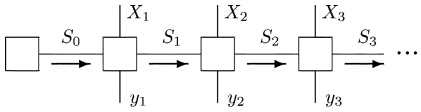

Fig. 1. Computation of $p(y^n)$ by message passing through the factor graph of (2).

II. Review of Basic Method

We briefly review the basic idea of [1] as presented in [7]. We first note that, as a consequence of the Shannon–McMillan–Breiman theorem and the assumptions of stationarity and ergodicity (cf. Section I), the sequence $-\frac{1}{n}\log p(X^n)$ converges with probability 1 to the entropy rate $H(X)$, the sequence $-\frac{1}{n}\log p(Y^n)$ converges with probability 1 to the differential entropy rate $h(Y)$, and the sequence $-\frac{1}{n}\log p(X^n, Y^n)$ converges with probability 1 to $H(X)+h(Y|X)$. From these observations, the quantity $I(X;Y) = h(Y) - h(Y|X)$ can be computed as follows.

1) Sample two "very long" sequences x^n and y^n.
2) Compute $\log p(x^n)\log p(y^n)$, and $\log p(x^n, y^n)$. If $h(Y|X)$ is known analytically, then it suffices to compute $\log p(y^n)$.
3) Conclude with the estimate

$$\hat{I}(X;Y) \stackrel{\triangle}{=} \frac{1}{n}\log p(x^n, y^n) - \frac{1}{n}\log p(x^n) - \frac{1}{n}\log p(y^n) \tag{3}$$

or, if $h(Y|X)$ is known analytically

$$\hat{I}(X;Y) \stackrel{\triangle}{=} -\frac{1}{n}\log p(y^n) - h(Y|X). \tag{4}$$

The computations in Step 2 can be carried out by forward sum-product message passing through the factor graph of (2), as is illustrated in Fig. 1. (See [13] for an introduction to factor graphs.) If the state space $\mathcal{S}$ is finite, this computation is just the forward sum-product recursion of the BCJR algorithm [14].

Consider, for example, the computation of

$$p(y^n) = \int_{x^n}\int_{s_0^n} p\,(x^n, y^n, s_0^n)\,dx^n\,ds_0^n. \tag{5}$$

(In [1] and [7], the integral (5) is actually a finite sum.) Define the state metric $\mu_k(s_k) \stackrel{\triangle}{=} p(s_k, y^k)$. By straightforward application of the sum-product algorithm [13] we recursively compute the messages (state metrics)

$$\begin{aligned}\mu_k(s_k) &= \int_{x_k}\int_{s_{k-1}} \mu_{k-1}(s_{k-1}) \\ &\quad\cdot p(x_k, y_k, s_k|s_{k-1})dx_k ds_{k-1} &(6)\\ &= \int_{x^k}\int_{s_0^{k-1}} p(x^k, y^k, s_0^k)dx^k ds_0^{k-1} &(7)\end{aligned}$$

for $k = 1, 2, 3, \ldots$ with $\mu_0(s_0) \stackrel{\triangle}{=} p(s_0)$. The desired quantity (5) is then obtained as

$$p(y^n) = \int_{s_n} \mu_n(s_n),\ ds_n \tag{8}$$

the sum of (or the integral over) all final state metrics.

For large k, the state metrics μ_k computed according to (6) quickly tend to zero. In practice, the recursion (6) is therefore changed to

$$\mu_k(s_k) = \lambda_k \int_{x_k}\int_{s_{k-1}} \mu_{k-1}(s_{k-1})p(x_k, y_k, s_k|s_{k-1})dx_k ds_{k-1} \tag{9}$$

where $\lambda_1, \lambda_2, \ldots$ are positive scale factors. We will choose these factors such that

$$\int_{s_k} \mu_k(s_k)\,ds_k = 1 \tag{10}$$

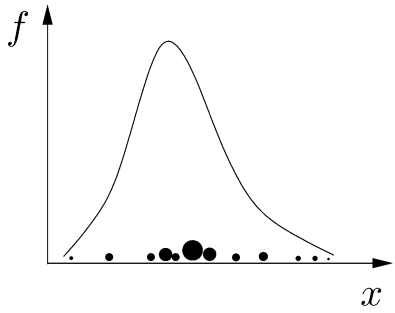

Fig. 2. A probability density function $f : \mathbb{R} \to \mathbb{R}^+$ and its representation as a list of particles.

holds for all k, i.e.

$$\begin{aligned}\lambda_k^{-1} &= \int_{x_k}\int_{s_{k-1}}\int_{s_k} \mu_{k-1}(s_{k-1}) \\ &\quad\cdot p(x_k, y_k, s_k|s_{k-1})dx_k ds_{k-1} ds_k &(11)\\ &= \int_{x_k}\int_{s_{k-1}}\int_{s_k} \mu_{k-1}(s_{k-1})p(x_k, s_k|s_{k-1}) \\ &\quad\cdot p(y_k|x_k, s_k, s_{k-1})dx_k ds_{k-1} ds_k. &(12)\end{aligned}$$

It follows that

$$\frac{1}{n}\sum_{k=1}^{n}\log\lambda_k = -\frac{1}{n}\log p(y^n). \tag{13}$$

The quantity $-\frac{1}{n}\log p(y^n)$ thus appears as the average of the logarithms of the scale factors, which converges (almost surely) to $h(Y)$.

If necessary, the quantities $\log p(x^n)$ and $\log p(x^n, y^n)$ can be computed by the same method, see [7].

For use in Section III, we note that λ_k^{-1} (12) may be written as an expectation; due to the normalization (10), the state metric $\mu_k(s_k)$ now equals $p(s_k|y^k)$, and therefore

$$\begin{aligned}\lambda_k^{-1} &= \int_{x_k}\int_{s_{k-1}}\int_{s_k} p(s_{k-1}|y^{k-1})p(x_k, s_k|s_{k-1}) \\ &\quad\cdot p(y_k|x_k, s_k, s_{k-1})dx_k ds_{k-1} ds_k &(14)\\ &= \int_{x_k}\int_{s_{k-1}}\int_{s_k} p(s_{k-1}, s_k, x_k|y^{k-1}) \\ &\quad\cdot p(y_k|x_k, s_k, s_{k-1})dx_k\,ds_{k-1} ds_k &(15)\\ &= \mathrm{E}[p(y_k|X_k, S_k, S_{k-1})|Y^{k-1}] &(16)\end{aligned}$$

where the expectation is with respect to the probability density

$$\begin{aligned}p(s_{k-1}, s_k, x_k|y^{k-1}) &= p(s_{k-1}|y^{k-1})p(x_k, s_k|s_{k-1}) &(17)\\ &= \mu_{k-1}(s_{k-1})p(x_k, s_k|s_{k-1}). &(18)\end{aligned}$$

III. A Particle Method

If both the input alphabet $\mathcal{X}$ and the state space $\mathcal{S}$ are finite sets (and the alphabet of $\mathcal{X}$ and $\mathcal{S}$ is not too large), then the method of the previous section is a practical algorithm. However, we are now interested in the case where $\mathcal{S}$ (and perhaps also $\mathcal{X}$) is continuous, as stated in the introduction. In this case, the computation of (9) and (16) is a problem.

This problem can be addressed by Monte-Carlo methods known as sequential Monte-Carlo integration ("particle filtering") [11], [12]. Such algorithms may be viewed as message passing algorithms where the messages (which represent probability distributions) are represented by a list of samples ("particles") [15]–[19] (see Fig. 2); a list $\mathcal{L}_f$ of N particles representing the probability density $f(x)$ with $x \in \mathcal{X}$ is formally defined as a list of pairs

$$\mathcal{L}_f \stackrel{\triangle}{=} \left\{\left(\hat{x}^{(1)}, w^{(1)}\right), \left(\hat{x}^{(2)}, w^{(2)}\right), \ldots, \left(\hat{x}^{(N)}, w^{(N)}\right)\right\} \tag{19}$$

$$\triangleq \left\{ \left(\hat{x}^{(\ell)}, w^{(\ell)} \right) \right\}_{\ell=1}^{N} \tag{20}$$

with $\hat{x}^{(\ell)} \in \mathcal{X}$ and where the weights $w^{(\ell)}$ are positive real numbers such that $\sum_{\ell=1}^{N} w^{(\ell)} = 1$.

In particular, we will represent the message μ_k by a list of N particles $\{\hat{s}_k^{(\ell)}, w_k^{(\ell)}\}_{\ell=1}^{N}$, and we will represent the distribution $p(s_{k-1}, s_k, x_k | y^{k-1})$ (18) by a list of N (weighted) three-tuples $\{(\hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)}), w_{k-1}^{(\ell)}\}_{\ell=1}^{N}$. The expectation (16) is then approximately computed as an average over those N (weighted) three-tuples

$$\lambda_k^{-1} \approx \sum_{\ell=1}^{N} w_{k-1}^{(\ell)} p(y_k | \hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)}). \tag{21}$$

The recursive computation of (9) is accomplished as follows.

1) Begin with a particle list $\{\hat{s}_{k-1}^{(\ell)}, w_{k-1}^{(\ell)}\}_{\ell=1}^{N}$ that represents μ_{k-1}.
2) Extend each particle $\hat{s}_{k-1}^{(\ell)}$ to a three-tuple $(\hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)})$ by sampling from $p(x_k, s_k | s_{k-1})$, resulting in the particle list $\{(\hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)}), w_{k-1}^{(\ell)}\}_{\ell=1}^{N}$.
3) Compute an estimate of λ_k using (21).
4) Compute the weights w_k:

$$w_k^{(\ell)} = \lambda_k w_{k-1}^{(\ell)} p(y_k | \hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)}). \tag{22}$$

(Note that those weights sum to one.)

5) Drop $\hat{s}_{k-1}^{(\ell)}$ and $\hat{x}_k^{(\ell)}$ of each three-tuple $(\hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)})$; the resulting particle list $\{\hat{s}_k^{(\ell)}, w_k^{(\ell)}\}_{\ell=1}^{N}$ represents μ_k.
6) If the number of "effective" particles $N_{k,\text{eff}}$ in the list $\{\hat{s}_k^{(\ell)}, w_k^{(\ell)}\}_{\ell=1}^{N}$ is "small", i.e., if

$$N_{k,\text{eff}} \triangleq \frac{1}{\sum_{\ell=1}^{N} \left(w_k^{(\ell)} \right)^2} < \varepsilon N \tag{23}$$

where ε is a positive number (e.g., $\varepsilon = 0.3$), "resample" the list $\{\hat{s}_k^{(\ell)}, w_k^{(\ell)}\}_{\ell=1}^{N}$:

a) Draw N samples from the list $\{\hat{s}_k^{(\ell)}\}_{\ell=1}^{N}$ with probability proportional to $w_k^{(\ell)}$. (If $w_k^{(\ell)}$ is large, the sample $s_k^{(\ell)}$ may be drawn several times, otherwise, it may not be drawn at all.)
b) Associate the (uniform) weight $\frac{1}{N}$ to each obtained sample $s_k^{(\ell)}$, resulting in the new list $\{\hat{s}_k^{(\ell)}, \frac{1}{N}\}_{\ell=1}^{N}$, which represents μ_k.

Some remarks follow.

- In Step 2 of the above algorithm, one needs to draw samples from $p(x_k, s_k | s_{k-1})$. A closed-form expression for $p(x_k, s_k | s_{k-1})$ is not required for that purpose. The state transitions may for example be described by a stochastic difference equation. The observation model $p(y_k | x_k, s_k, s_{k-1})$, however, has to be available in closed-form (cf. Step 3 and 4).
- Without resampling (Step 6), all but one particle will have negligible weight after a few iterations ("degeneracy"); the resampling step reduces this effect (Step 6) [11], [12].
- It is well known that particle-based estimates of logarithmic Lyapunov exponents (or "log partition functions", cf. (13) and (21)) are unbiased [20], [21]. The mean square error of those estimates is upper bounded by an expression that is inversely proportional to the number of particles N (for $n > \sqrt{N}$) [22, Theorem 2, Corollary 2]; those two properties carry over to the particle-based estimate $\hat{I}(X,Y)$ (3), since the latter is a linear combination of particle-based estimates of logarithmic Lyapunov exponents.

IV. A Numerical Example

We consider the channel

$$Y_k = X_k e^{j\Theta_k} + N_k, \tag{24}$$

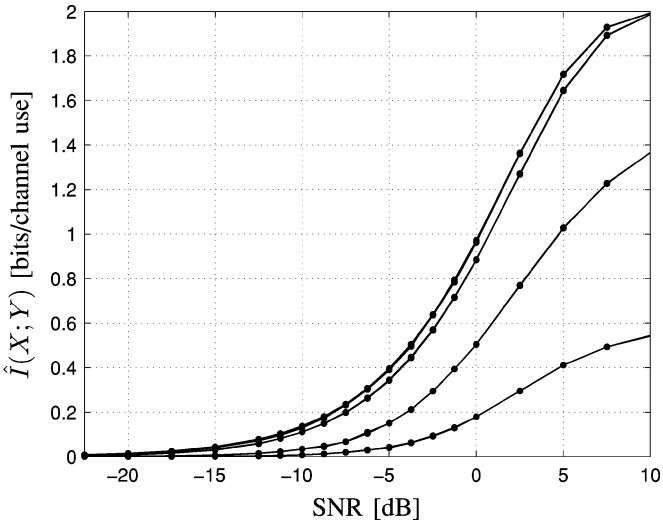

Fig. 3. Information rates for the random-walk phase noise channel (25). From top to bottom: $\sigma_W = 0$ and $\sigma_W = 0.01$ (on top of each other), $\sigma_W = 0.1$, $\sigma_W = 0.5$, and $\sigma_W = 1$.

where X_k is the complex channel input symbol at time k, Y_k is the corresponding channel output symbol, and N_k is white Gaussian noise with known variance σ_N^2. For the sake of definiteness, we will assume, first, that the channel input alphabet $\mathcal{X}$ is a 4-PSK constellation, and second, that the channel input symbols $X_k, k = 1, 2, \ldots$, are independent and uniformly distributed. The phase Θ_k (which takes the role of the channel state S_k) is unknown to the receiver. We consider two dynamical models for the phase:

Random-walk phase model

$$\Theta_k = (\Theta_{k-1} + W_k) \bmod 2\pi \tag{25}$$

where W_k is white Gaussian noise with known variance σ_W^2.

ARMA phase model

$$Z_k = \sum_{\ell=1}^{m_a} a_\ell Z_{k-\ell} + \sum_{\ell=0}^{m_b} b_\ell W_{k-\ell}, \tag{26}$$

$$\Theta_k = Z_k \bmod 2\pi \tag{27}$$

with known real coefficients a_ℓ and b_ℓ and where W_k is white Gaussian noise with known variance σ_W^2.

This channel models a single-carrier communications system with phase jitter and perfect symbol timing knowledge [23]. The two phase noise models (random-walk (25) and ARMA (26)) correspond to a free running clock and a phase-locked loop respectively [24] (see also [19, Ch. 2]

For this channel (with both phase noise models), the application of the method of Section III is straightforward. Some numerical results are shown in Figs. 3 and 4. For the example in Fig. 4, the parameters of the ARMA model (26) are $m_a = 1, m_b = 2, a_1 = 0.4$ and $(b_0, b_1, b_2) = (0.3, 0.2, 0.1)$. In both Figs. 3 and Fig. 4, we simulated channel input/output sequences of length n between 10^5 and 10^6, and we used $N = 10^4$ particles.

The numerical results of Fig. 3 were also checked with the auxiliary-channel method of [7], and the results agree up to the accuracy of the plot. The auxiliary channel is in this case a quantized version of (25) where Θ_k is quantized into 5000 bins. Note that quantization of the state space is not practical for the ARMA noise model.

The convergence of the proposed method is illustrated by Fig. 5, which shows the estimates $\hat{I}(X;Y)$ of 10 different simulation runs as a function of the sequence length n (for the random-walk model).

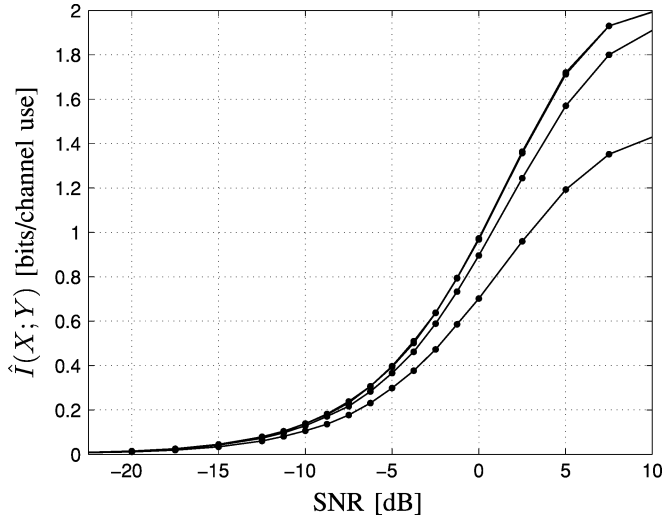

Fig. 4. Information rates for the ARMA phase noise channel (26) with $m_a = 1, m_b = 2, a_1 = 0.4$, and $(b_0, b_1, b_2) = (0.3, 0.2, 0.1)$. From top to bottom: $\sigma_W = 0, \sigma_W = 0.01$, and $\sigma_W = 0.1$ (all on top of each other), $\sigma_W = 0.5$, and $\sigma_W = 1$.

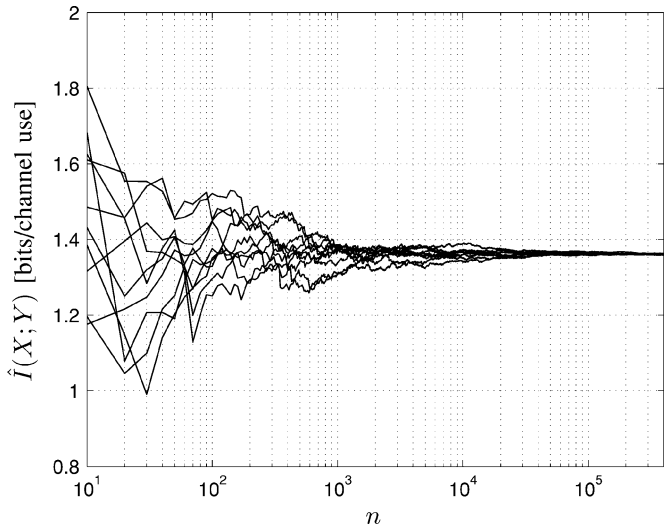

Fig. 5. Estimated information rate (for the random-walk phase noise channel) as a function of the sequence length n, for 10 simulation runs of the particle method, for $\mathrm{SNR} = 10$ dB and $\sigma_W = 0.5$.

V. Conclusion

Using particle methods, we have extended the basic idea of [1] and [7] to channels with a continuous state space. A closed-form expression of the state transition probability is not required. The accuracy of the proposed method depends not only on the length of the simulated sequence (as in [1], [7]), but also on the number of particles.

It should be noted that the proposed method can be used also to compute the auxiliary-channel bounds on the information rate of [7, Sec. VI].

Acknowledgment

The authors wish to thank Nicolas Cedraschi for assistance with the simulations and Daniel Hösli, Volker Koch, and Arnaud Doucet for helpful discussions and comments.

References

[1] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," in *Proc. 2001 IEEE Int. Conf. Commun.*, Helsinki, Finland, Jun. 11–14, 2001, pp. 2692–2695.

[2] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to Markov channels," in *Proc. 2001 IEEE Int. Symp. Inf. Theory*, Washington, DC, June 24–29, 2001, p. 283.

[3] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite-state ISI channels," in *Proc. 2001 IEEE GLOBECOM*, San Antonio, TX, Nov. 25–29, 2001, pp. 2992–2996.

[4] D. Arnold, H.-A. Loeliger, and P. O. Vontobel, "Computation of information rates from finite-state source/channel models," in *Proc. 40th Annu. Allerton Conf. Commun., Contr. Comput.*, Monticello, IL, Oct. 2–4, 2002, pp. 457–466, Allerton House.

[5] D. Arnold, A. Kavčić, H.-A. Loeliger, P. O. Vontobel, and W. Zeng, "Simulation-based computation of information rates: upper and lower bounds," in *Proc. 2003 IEEE Int. Symp. Inf. Theory*, Yokohama, Japan, Jun./Jul. 29–4, 2002, p. 119.

[6] D. Arnold, "Computing Information Rates of Finite-State Models With Application to Magnetic Recording," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2002, ETH 14760.

[7] D. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavčić, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug. 2007.

[8] S. Egner, V. B. Balakirsky, L. Tolhuizen, S. Baggen, and H. Hollmann, "On the entropy rate of a hidden Markov model," in *Proc. 2004 IEEE Int. Symp. Inf. Theory*, Chicago, IL, Jun./Jul. 27–2, 2004, p. 12.

[9] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[10] O. Shental, N. Shental, and S. Shamai Shitz, "On the achievable information rates of finite-state input two-dimensional channels with memory," in *Proc. 2005 IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 4–9, 2005, pp. 2354–2358.

[11] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[12] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[13] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, pp. 28–41, Jan. 2004.

[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[15] M. Isard, "Pampas: Real-valued graphical models for computer vision," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, Jun. 16–22, 2003, vol. 1, pp. 613–620.

[16] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Non-parametric belief propagation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, USA, Jun. 16–22, 2003.

[17] M. Briers, A. Doucet, and S. S. Singh, "Sequential auxiliary particle belief propagation," in *Proc. 7th Int. Conf. Inf. Fusion*, 2005.

[18] J. Dauwels, S. Korl, and H.-A. Loeliger, "Particle methods as message passing," in *Proc. 2006 IEEE Int. Symp. Inf. Theory*, Seattle, WA, July 9–14, 2006, pp. 2052–2056.

[19] J. Dauwels, "On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, Dec. 2005, ETH 16365.

[20] P. Del Moral and L. Miclo, J. Azéma, M. Eméry, M. Ledoux, and M. Yor, Eds., "Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to nonlinear filtering," *Séminaire de Probabilités XXXIV;* , ser. Lecture Notes in Mathematics 1729, pp. 1–145, 2000.

[21] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer, 2004.

[22] P. Del Moral and A. Doucet, "Particle motions in absorbing medium with hard and soft obstacles," *Stoch. Anal. Appl.*, vol. 22, no. 5, pp. 1175–1207, 2004.

[23] M. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing*. New York: Wiley, 1998, pp. 246–249.

[24] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: A unifying theory and numerical methods for characterization," *IEEE Trans. Circuits Syst. I*, vol. 47, no. 5, pp. 655–674, May 2000.