December 8, 2006

# Computation of Information Rates by Particle Methods

Justin Dauwels and Hans-Andrea Loeliger

**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").

## 1 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] [2] [3]; an alternative approach is presented in [4]. In [5], the approach of [1] [2] [3] was extended to 2-D channels (using generalized belief propagation [6]). In [7], the method of [1] [2] [3] was described in greater generality and extended to the computation of upper and lower bounds on the information rate of very general channels (see also [8] [9]). Another method to compute an upper bound was presented in [10].

J. Dauwels was with the Dept. of Information Technology and Electrical Engineering, ETH, CH-8092 Zürich, Switzerland. He is now with the RIKEN Brain Science Institute, Hirosawa 2-1, Wako-shi, 351-0106, Saitama, Japan. Email: justin@dauwels.com. His work was supported in part by Swiss National Science Foundation grant 200021-101955. He is currently supported by a Post-Doctoral Fellowship (No. PE05060) from the Japanese Society for the Promotion of Science (JSPS).

H.-A. Loeliger is with the Dept. of Information Technology and Electrical Engineering, ETH, CH-8092 Zürich, Switzerland. Email: loeliger@isi.ee.ethz.ch.

The material in this paper was presented in part at the 2004 IEEE International Symposium on Information Theory, Chicago, USA, June 27–July 2, 2004.

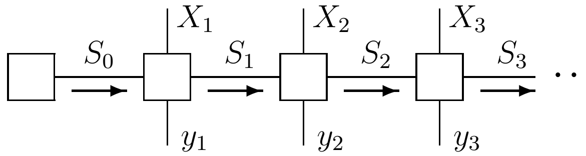


Figure 1: Computation of $p(y^n)$ by message passing through the factor graph of (2).

In this paper, we extend the methods of [1] and [7] 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 $\nu$-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 paper is structured as follows. In Section 2, we review the basic idea of [1] as presented in [9]. In Section 3, we show how particle methods allow to deal with a continuous state space; in Section 4, we present a numerical example, i.e., a communications channel with phase noise. In Section 5, we offer some concluding remarks.

# 2 Review of Basic Method

We briefly review the basic idea of [1] as presented in [9]. We first note that, as a consequence of the Shannon-McMillan-Breiman theorem, 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

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

   or, if $h(Y|X)$ is known analytically, $\hat{I}(X;Y) \triangleq -\frac{1}{n}\log p(y^n) - h(Y|X)$.

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

$$p(y^n) = \int_{x^n}\int_{s_0^n} p(x^n, y^n, s_0^n)\,dx^n ds_0^n. \tag{4}$$

(In [1] and [9], the integral (4) is actually a finite sum.) Define the state metric $\mu_k(s_k) \triangleq p(s_k, y^k)$. By straightforward application of the sum-product algorithm [13] we recursively compute the messages (state metrics)

$$\begin{aligned} \mu_k(s_k) &= \int_{x_k}\int_{s_{k-1}} \mu_{k-1}(s_{k-1})\, p(x_k, y_k, s_k|s_{k-1}) dx_k ds_{k-1} && (5) \\ &= \int_{x^k}\int_{s_0^{k-1}} p(x^k, y^k, s_0^k) dx^k ds_0^{k-1} && (6) \end{aligned}$$

for $k = 1, 2, 3, \ldots$ with $\mu_0(s_0) \triangleq p(s_0)$. The desired quantity (4) is then obtained as

$$p(y^n) = \int_{s_n} \mu_n(s_n) ds_n, \tag{7}$$

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

For large $k$, the state metrics $\mu_k$ computed according to (5) quickly tend to zero. In practice, the recursion (5) is therefore changed to

$$\mu_k(s_k) = \lambda_k \int_{x_k}\int_{s_{k-1}} \mu_{k-1}(s_{k-1})\, p(x_k, y_k, s_k|s_{k-1}) dx_k ds_{k-1}, \tag{8}$$

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

$$\int_{s_k} \mu_k(s_k) ds_k = 1 \tag{9}$$

holds for all $k$, i.e.,

$$\begin{aligned} \lambda_k^{-1} &= \int_{s_k}\int_{x_k}\int_{s_{k-1}} \mu_{k-1}(s_{k-1})\, p(x_k, y_k, s_k|s_{k-1}) dx_k ds_{k-1}, && (10) \\ &= \int_{s_k}\int_{x_k}\int_{s_{k-1}} \mu_{k-1}(s_{k-1})\, p(x_k, s_k|s_{k-1})\, p(y_k|x_k, s_k, s_{k-1}) ds_k dx_k ds_{k-1} && (11) \\ &= \mathrm{E}\big[p(y_k|X_k, S_k, S_{k-1})|Y^{k-1}\big], && (12) \end{aligned}$$

where the expectation is w.r.t. the probability density function (pdf)

$$p(s_{k-1}, s_k, x_k|y^{k-1}) = \mu_{k-1}(s_{k-1}) p(x_k, s_k|s_{k-1}). \tag{13}$$

It then follows that

$$\frac{1}{n}\sum_{k=1}^{n} \log \lambda_k = -\frac{1}{n} \log p(y^n). \tag{14}$$

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 [9].

# 3 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 (8) and (12) 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 pdf $f(x)$ with $x \in \mathcal{X}$ is formally defined as a list of pairs

$$\mathcal{L}_f \triangleq \big\{(\hat{x}^{(1)}, w^{(1)}), (\hat{x}^{(2)}, w^{(2)}), \ldots, (\hat{x}^{(N)}, w^{(N)})\big\} \triangleq \big\{(\hat{x}^{(\ell)}, w^{(\ell)})\big\}_{\ell=1}^{N}, \tag{15}$$

where $\hat{x}^{(\ell)} \in \mathcal{X}$, the weights $w^{(\ell)}$ are positive real numbers and $\sum_{\ell=1}^{N} w^{(\ell)} = 1$. In partic-

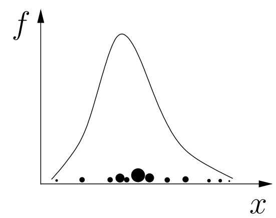


Figure 2: A probability density function $f : \mathbb{R} \to \mathbb{R}^+$ and its representation as a list of particles.

ular, we will represent the message $\mu_k$ by a list of $N$ particles $\{\hat{s}_k^{(\ell)}, w_k^{(\ell)}\}_{\ell=1}^{N}$, and we will represent the distribution $p(s_{k-1}, s_k, x_k | y^{k-1})$ (13) 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 (12) is then approximately computed as an average over those $N$ (weighted) three-tuples:

$$\lambda_k^{-1} \approx \sum_{\ell=1}^{N} w_{k-1}^{(\ell)}\, p(y_k | \hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)}). \tag{16}$$

The recursive computation of (8) is accomplished as follows.

1. Begin with a particle list $\{\hat{s}_{k-1}^{(\ell)}, w_{k-1}^{(\ell)}\}_{\ell=1}^{N}$ that represents $\mu_{k-1}$.
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 $\lambda_k$ using (16).
4. Compute the weights $w_k$:

$$w_k^{(\ell)} = \lambda_k\, w_{k-1}^{(\ell)}\, p(y_k | \hat{s}_{k-1}^{(\ell)}, \hat{s}_k^{(\ell)}, \hat{x}_k^{(\ell)}). \tag{17}$$

(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 $\mu_k$.
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

$$N_{k,\text{eff}} \triangleq \frac{1}{\sum_{\ell=1}^N \big(w_k^{(\ell)}\big)^2} < \varepsilon N, \tag{18}$$

where $\varepsilon$ 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 $\mu_k$.

Some remarks:

- 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. (14) and (16)) 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.

# 4 A Numerical Example

We consider the channel

$$Y_k = X_k\, e^{j\Theta_k} + N_k, \tag{19}$$

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 $\sigma_N^2$. 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 process is independently and uniformly distributed (i.u.d.). The phase $\Theta_k$ (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**

$$\Theta_k = (\Theta_{k-1} + W_k) \bmod 2\pi, \tag{20}$$

where $W_k$ is white Gaussian noise with known variance $\sigma_W^2$.

**ARMA phase model**

$$Z_k = \sum_{\ell=1}^{m_a} a_\ell \, Z_{k-\ell} + \sum_{\ell=0}^{m_b} b_\ell \, W_{k-\ell}, \tag{21}$$

$$\Theta_k = Z_k \bmod 2\pi, \tag{22}$$

where $W_k$ is white Gaussian noise with known variance $\sigma_W^2$.

This channel models a single-carrier communications system with phase jitter and perfect symbol timing knowledge [23]. The random-walk and ARMA phase noise model is a simple model for phase noise in a free running clock and a phase-locked loop respectively [24] (see also [19, Chapter 2]).

For this channel, the application of the method of Section 3 is straightforward; the results for the random-walk and ARMA phase noise model are shown in Figures 3 and 4 respectively. For these computations, 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 in Figure 3 (for the random-walk phase noise model) were also checked with the auxiliary-channel method of [9] based on a quantization of $\Theta$ (with 5000 uniform quantization bins). (Note that the state space in the random-walk phase model is one-dimensional and bounded, and therefore, quantization is feasible; the state space of the ARMA phase model is high-dimensional and quantization is in this case not practical.) The numerical results agree with those in Figure 3 up to the accuracy of the plot. Figure 5 depicts the estimate $\hat{I}(X;Y)$ (for the random-walk phase noise model) as a function of the sequence length $n$.

# 5 Conclusion

Using particle methods, we have extended the basic idea of [1] and [9] 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] [9]), 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 [9, Section VI].

# Acknowledgement

The authors wish to thank Nicolas Cedraschi for assistance with the simulations. The helpful discussions with Daniel Hösli and Volker Koch are also gratefully acknowledged.

The first author is grateful to Arnaud Doucet for helpful comments.

# References

[1] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," *Proc. 2001 IEEE Int. Conf. on Communications,* Helsinki, Finland, June 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", *Proc. 2001 IEEE Int. Symp. Information Theory,* Washington, DC, USA, 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," *Proc. 2001 IEEE Globecom,* San Antonio, TX, pp. 2992–2996, Nov. 25–29, 2001.

[4] S. Egner, V. B. Balakirsky, L. Tolhuizen, S. Baggen, and H. Hollmann, "On the entropy rate of a hidden Markov model," *Proc. 2004 IEEE International Symposium on Information Theory,* p. 12, Chicago, USA , June 27–July 2, 2004.

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

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

[7] D. Arnold, H.-A. Loeliger, and P. O. Vontobel, "Computation of information rates from finite-state source/channel models," *Proc. 40th Annual Allerton Conference on Communication, Control, and Computing,* (Allerton House, Monticello, Illinois), October 2–4, 2002, pp. 457–466. Available from `http://www.isi.ee.ethz.ch/~loeliger`.

[8] D. Arnold, *Computing Information Rates of Finite-State Models with Application to Magnetic Recording.* ETH Dissertation no. 14760, ETH Zürich, Switzerland, 2002.

[9] 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. Information Theory,* vol. 52, no. 8, pp. 3498–3508, August 2006.

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

[11] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, eds., *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 non-linear/non-Gaussian Bayesian tracking", *IEEE Trans. Signal Proc.*, vol. 50, February 2002, pp. 174–188.

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

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

[15] M. Isard, "Pampas: Real-Valued Graphical Models for Computer Vision," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03),* Madison, Wisconsin, USA, June 16–22, 2003, vol. 1, pp. 613–620.

[16] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky "Non-Parametric Belief Propagation," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03),* Madison, Wisconsin, USA, June 16–22, 2003.

[17] M. Briers, A. Doucet, and S. S. Singh, "Sequential Auxiliary Particle Belief Propagation," *Proc. 7th Intern. Conf. on Information Fusion,* 2005.

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

[19] J. Dauwels, *On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation,* PhD. Thesis at ETH Zurich, Diss. ETH No 16365, December 2005. Available from `www.dauwels.com/PhD.htm`.

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

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

[22] P. Del Moral and A. Doucet, "Particle motions in absorbing medium with hard and soft obstacles," *Stochastic Analysis and Applications,* 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, John Wiley & Sons, 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, May 2000, pp. 655–674.

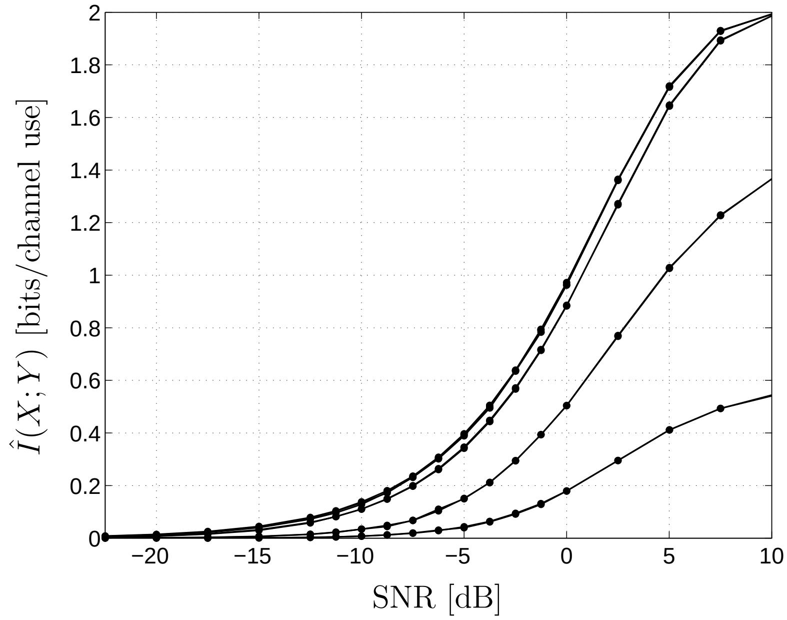


Figure 3: Information rates for the random-walk phase noise channel. From top to bottom: $\sigma_W = 0$ and $\sigma_W = 0.01$ (on top of each other), $\sigma_W = 0.1$, $\sigma_W = 0.5$, and $\sigma_W = 1$.

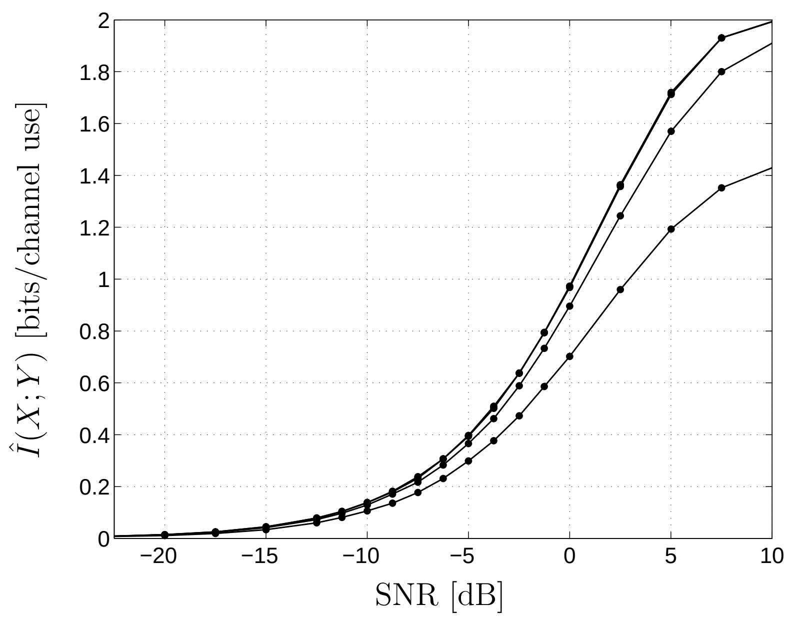


Figure 4: Information rates for the ARMA phase noise channel (with $m_a = 1$, $m_b = 2$, $a_1 = 0.4$, and $(b_0, b_1, b_2) = (0.3, 0.2, 0.1)$, cf. (21)). From top to bottom: $\sigma_W = 0$, $\sigma_W = 0.01$, and $\sigma_W = 0.1$ (on top of each other), $\sigma_W = 0.5$, and $\sigma_W = 1$.

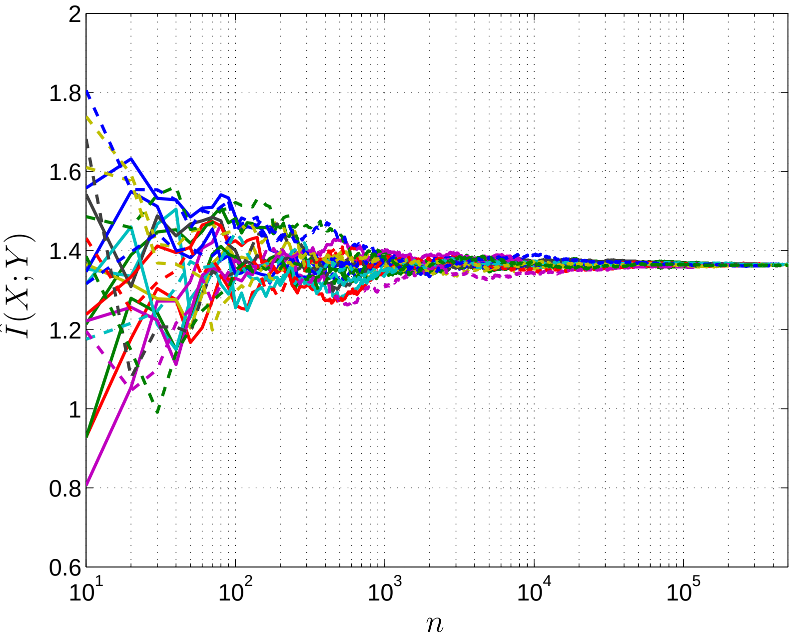


Figure 5: Estimated information rate (for the random-walk phase noise channel) as a function of the sequence length $n$, for 10 simulation runs of the particle method (dashed lines) and ten runs of the auxiliary-channel method (solid lines), for SNR = 10dB and $\sigma_W = 0.5$.