

# A New Encoder for Continuous-Time Gaussian Signals With Fixed Rate and Reconstruction Delay

Damián Marelli, Kaushik Mahata, and Minyue Fu, *Fellow, IEEE*

**Abstract**—In this paper, we propose a method for encoding continuous-time Gaussian signals subject to a usual data rate constraint and, more importantly, a reconstruction delay constraint. We first apply a Karhunen-Loève decomposition to reparameterize the continuous-time signal as a discrete sequence of vectors. We then study the optimal recursive quantization of this sequence of vectors. Since the optimal scheme turns out to have a very cumbersome design, we consider a simplified method, for which a numerical example suggests that the incurred performance loss is negligible. In this simplified method, we first build a state space model for the vector sequence and then use Bayesian tracking to sequentially encode each vector. The tracking task is performed using particle filtering. Numerical experiments show that the proposed approach offers visible advantages over other available approaches, especially when the reconstruction delay is small.

**Index Terms**—Bayesian methods, continuous-time signals, particle filters, predictive coding, quantization, state-space methods, transform coding.

## I. INTRODUCTION

**M**OST digital data transmitted over communication channels originate from continuous-time signals with the purpose of having the original continuous-time signals reconstructed. While many applications do not have a strict requirement on reconstruction delays, many other applications are of real-time nature. With the rapid growth in high-speed communication networks, real-time applications such as smart electricity grids [1], [2], and networked control systems [3], [4] are made possible. In both cases, fast changing continuous-time signals need to be digitally encoded and transmitted over a communication network, and the reconstruction of the signal needs to be made with minimal time delay to facilitate fast decision making. Such applications impose a new challenge on the encoding technology.

In principle, the encoding problem considered in this paper could be roughly stated as follows: *Given a maximum reconstruction delay  $T$  (in seconds) and a fixed (average) bit rate*

*$R$  (number of bits per  $T$  seconds), we need to encode a continuous-time signal  $y(t)$  in a given class (to be specified later) so that, at any time  $t$ , a reconstructed version  $\hat{y}(\tau)$  of  $y(\tau)$  is available for all  $\tau \leq t - T$ , and the reconstruction error is minimized in some sense (to be specified later).* However, this problem is too general. In particular, notice that, since only the average bit rate is specified, this problem allows using variable rate codes, which are not suitable for our intended real-time network applications. Hence, we consider a particular case of this problem instead. More precisely, we further assume that  $R$  bits are transmitted at every  $t = kT$  ( $k$  being an integer). From this problem statement, it follows that we only need to transmit data once every  $T$  seconds<sup>1</sup>. However, we differentiate the data transmission rate from the sampling rate. More precisely, a rate much higher than  $1/T$  can be chosen to sample the continuous-time signal, provided that we can encode the sampled signal within the given bit rate  $R$ . With the current advances in digital electronics, very fast sampling devices are easily implementable. Hence, we can realistically assume that, at any time  $t$ , the encoder knows the whole continuous-time signal up to time  $t$ . On the other hand, the constraint on the data transmission rate is often unavoidable, especially for large communication networks or wireless links. With the above thinking, our encoding problem can be restated as follows: *Given that the transmitter knows the whole continuous-time signal  $y(t)$  up to time  $kT$ , and that the receiver knows the bits transmitted up to time  $(k-1)T$ , which  $R$  bits of digital information (i.e., update) need to be transmitted, so that the receiver can reconstruct the continuous-time signal up to time  $kT$  with minimal distortion?*

Traditionally, continuous-time signals are encoded by first sampling the signal and then quantizing the samples. Guided by the Nyquist-Shannon sampling theorem, the sampling frequency is typically chosen to be higher than, but close to, the minimum sampling frequency which is twice of the signal bandwidth [5]. This sample-and-quantize approach is popular because of its simplicity and is adequate if the purpose is to inform on the sampled signal. However, when the purpose is to reconstruct the original continuous-time signal, reconstruction time delay is inevitable with this approach. A natural way to reduce the time delay is to sample faster. But this results in a higher transmission rate (number of transmissions per second) and a higher data rate (number of bits per second)<sup>2</sup>. Another drawback of the sample-and-quantize approach is that, in many

Manuscript received July 07, 2011; revised October 16, 2011 and January 18, 2012; accepted February 19, 2012. Date of publication March 06, 2012; date of current version May 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Olgica Milenkovic.

D. Marelli and K. Mahata are with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: Damian.Marelli@newcastle.edu.au; Kaushik.Mahata@newcastle.edu.au).

M. Fu is with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia. He is also with the Department of Control Science and Engineering, Zhejiang University, China (e-mail: Minyue.Fu@newcastle.edu.au).

Digital Object Identifier 10.1109/TSP.2012.2190064

<sup>1</sup>We assume that transmission delay and computational delay are negligible. But this assumption can be relaxed without adversely affecting our approach.

<sup>2</sup>The possibility of using entropy coding to reduce the high data rate resulting from a high sampling rate was studied in [6]. However, this introduces extra reconstruction delay.

cases it may not be realistic to assume that the continuous-time signal is band-limited.

The drawbacks of the sample-and-quantize approach are avoided by using transform coding [7]. This technique uses a linear transformation to obtain a vector of real coefficients which represents the signal over each time-interval of length  $T$ . Hence, the sequence of such vectors can be considered as an alternative representation of the continuous-time signal. The essential difference between this linear transformation and the sampling operation used in the sample-and-quantize approach, is that each coefficient vector provides a complete representation of each signal segment, whereas samples from infinite past and future are needed to represent each segment. Once the coefficient vector is computed, it needs to be encoded. In classical transform coding, this is typically done using scalar quantization on each vector entry. However, it is also possible to use vector quantization techniques (e.g., generalized Lloyd's algorithm or linear predictive vector quantization) [7].

In this paper we propose a coding method to address the delay-constrained problem described above, under the assumption that the signal  $y(t)$  to be encoded is Gaussian. The proposed method uses transform coding, more precisely the Karhunen-Loève (KL) decomposition [8], to obtain a sequence of vectors, as described above. We then study the optimal recursive strategy for quantizing these vectors. Unfortunately, it turns out that the optimal quantizer has a very cumbersome design. However, a simplified version of it leads to an accessible design. In this simplified version, each vector is encoded using its joint probability density function (pdf) conditioned on the previously transmitted quantized data. This is done recursively, every  $T$  seconds, and this process involves updating the joint pdf and the quantization dictionary (code book). In our approach, these are done with the aid of a state-space model of the KL-decomposed sequence and a particle filter. To support our choice of this simplified scheme, we present a numerical example suggesting that the distortion increase resulting from this simplification is indeed negligible. In addition, we present simulation results showing that the proposed approach leads to a smaller reconstruction error, when compared with other available methods, especially for small reconstruction delays. While the complexity of our approach is significantly higher than that of other methods, it is affordable when the reconstruction delays is small. Hence, the proposed method is a valid alternative for coding under small reconstruction delays, provided that the extra computational complexity is affordable.

We have explained above our strategy for quantizing the sequence of vectors resulting from a KL decomposition. A technique related to this strategy is known as *sequential quantization*, in the context of vector quantization [9]. In this technique, the (scalar) components of a vector are sequentially quantized using their pdf conditioned on the previously quantized samples. In contrast, in our problem we need to quantize an infinite sequence of vectors, rather than a finite sequence of scalars. Hence, we cannot express the conditional pdfs analytically or using a training sequence, but we need to resort to numerical methods (particle filtering) for pdf tracking.

We note that delay-constrained encoding has been an active research topic in coding theory for a long time. First, it is well

known that, if no limitation is imposed on the reconstruction delay, the theoretical minimum reconstruction distortion, for a given bit rate, is given by the *distortion-rate function* of the continuous-time signal to be encoded [8]. When time delay is constrained, the problem becomes more difficult. For the case of zero-delay reconstruction, it was shown in [10] that, for a discrete-time signal whose samples are statistically independent, the theoretical minimum distortion is achieved using scalar Lloyd-Max quantization. This means that optimal quantization is achieved by considering the knowledge of each sample independently, rather than jointly with its previous samples. For Markov sources, it was shown in [11] that if the source is  $k$ th-order Markov, the minimum distortion is achieved by forming each coded symbol using the last  $k$  source symbols, and the current state of the receiver (which is built using the past coded symbols). For the case of first-order Markov sources, this result was extended in [12] for the scenario where code symbols are transmitted through a noisy channel with noiseless feedback. The same problem, but without feedback, was studied in [13], concluding that the minimum distortion can be achieved considering the current source symbol, and the probability distribution (according to the encoder) of the decoder's state. A number of works study optimal coding structures where the performance measure is not simply given by the distortion. In this line, the authors of [14] studied the optimal coding of Markov sources, in the sense of minimizing a weighted sum of the distortion and the conditional entropy of the coded sequence. Also, optimal variable-rate coding, where the cost function is a weighted sum of distortion and rate, is studied in [15].

In [16] and [17], a variant of the zero-delay coding scheme called *causal coding* is studied. In this scheme, the reproduction value of each output depends on the present and past outputs. However, there is no constraint on the reconstruction delay, and therefore, it permits the placement of an entropy encoder after the quantizer. Also, theoretical minimum bounds for zero- and limited-delay coding were studied in [18]–[20], [17], for the individual sequence setting, where the signal to be coded is not assumed to be a random process but a deterministic bounded function. It is unfortunate that no simple expression is available for the distortion-rate function of a general stationary random process with correlated samples under zero or limited reconstruction delay.

The rest of the paper is organized as follows. In Section II we describe the limited-delay coding problem. In Section III we study some properties of optimal limited-delay coding, and we propose a sub-optimal coding strategy which is suitable for practical implementation. In Section IV we derive a numerical algorithm for implementing the proposed coding strategy. In Section V we present numerical experiments comparing the performance of the proposed coding algorithm with that of other available methods, and we give concluding remarks in Section VI.

## II. PROBLEM DESCRIPTION

Let  $y(t)$ ,  $t \in \mathbb{R}$  be a continuous-time stationary Gaussian random process, with zero mean and known autocorrelation  $r_y(\tau) = \mathcal{E}\{y(t)y(t+\tau)\}$ . The problem to be addressed is how

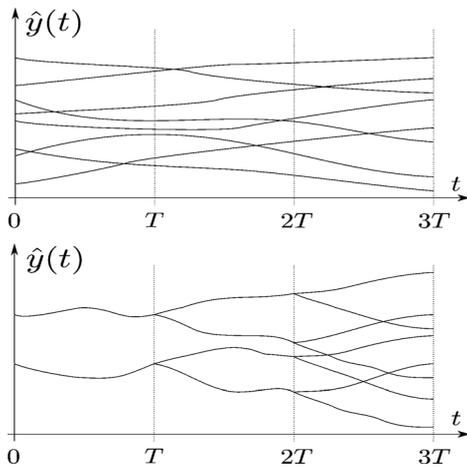


Fig. 1. Comparison between UDC (top) and LDC (bottom) dictionaries.

to code the signal  $y(t)$ , assuming that  $R$  bits are transmitted at every  $t = kT$ , for  $k \in \mathbb{N}$ , so that the distortion

$$D = \lim_{L \rightarrow \infty} \mathcal{E} \left\{ \frac{1}{L} \int_0^L (y(t) - \hat{y}(t))^2 dt \right\} \quad (1)$$

is minimized.

In order to give some insight into the problem, we introduce the concepts of *unlimited-delay coding* (UDC) and *limited-delay coding* (LDC). Suppose that we want to code the signal  $y(t)$ , on the interval  $t \in [0, KT]$ , using  $KR$  bits (i.e., using  $R/T$  bits per second). Using UDC, the coding is done by determining a dictionary of *codeword* signals  $\hat{y}_i(t)$ ,  $i = 1, \dots, 2^{KR}$ , which are chosen to minimize the distortion

$$D = \mathcal{E} \left\{ \frac{1}{KT} \int_0^{KT} (y(t) - \hat{y}(t))^2 dt \right\} \quad (2)$$

when coding is done by choosing the codeword  $\hat{y}(t)$  of the dictionary which is closest to  $y(t)$ , i.e.,

$$\hat{y}(t) = \arg \min_{\hat{y}_i(t)} \int_0^{KT} (y(t) - \hat{y}_i(t))^2 dt.$$

For a given rate  $R$ , the distortion  $D$  is minimized when  $K$  tends to infinity, in which case it is given by the *distortion-rate function*  $D(R)$  of  $y(t)$  [8]. In this approach, the decoding side is only able to recover  $\hat{y}(t)$  with a delay of  $KT$ . If instead, the decoder needs to recover  $\hat{y}(t)$  with a maximum delay of  $T$ , we use LDC. More precisely, we define the codewords  $\hat{y}_i(t)$ ,  $i = 1, \dots, 2^{KR}$ ,  $t \in [0, KT]$ , so that the decoder can recover  $\hat{y}(t)$  on the interval  $t \in [(k-1)T, kT]$ , after  $t = kT$ , for each  $k = 1, \dots, K$ . To illustrate this idea, we compare in Fig. 1 an UDC dictionary and a LDC dictionary for  $K = 3$  and  $R = 1$ .

### III. LIMITED-DELAY CODING

In this section, we assume that the autocorrelation  $r_y(\tau)$  of  $y(t)$ , the maximum delay  $T$ , and the number of bits  $R$  per time interval of length  $T$  are given. We then propose a suboptimal LDC strategy for  $y(t)$ , in the sense of minimizing the distortion

(1). For convenience in the presentation, we constrain the domain of  $y(t)$  to the interval  $t \in [0, KT]$ , for a given  $K \in \mathbb{N}$ , and we minimize (2) instead of (1). However, the resulting coding strategy, which is stated in Section III-C, is independent of  $K$ . Therefore, it is readily applicable in a practical scenario, where the domain of  $y(t)$  is  $[0, \infty)$ .

#### A. Order Reduction Using the Karhunen-Loève Decomposition

Our coding problem can be restated as that of coding the sequence of segments  $y(t)$ ,  $t \in [(k-1)T, kT]$ , for  $k = 1, \dots, K$ . As explained in Section I, in practice the continuous-time signal  $y(t)$  is sampled using a very dense grid of points. Hence, the coding of the sequence of segments mentioned above turns into a very high dimensional problem. To avoid this, we resort to transform coding for reducing the dimension of the problem, as explained in [7, Sec. 12.6] in the context of vector quantization. More precisely, we use the KL decomposition [8].

Using the KL decomposition, we can expand  $y(t)$ , on each interval  $[(k-1)T, kT]$ , as follows:

$$y(t) = \sum_{m=1}^{\infty} y_m(k) \psi_m(t - (k-1)T) \quad (3)$$

where the functions  $\psi_m : [0, T] \rightarrow \mathbb{R}$ ,  $m \in \mathbb{N}$  are orthonormal (i.e.,  $\int_0^T \psi_m(t) \psi_n(t) dt = \delta(m-n)$ ), and the coefficients  $y_m = \int_0^T y(t) \psi_m(t) dt$  are uncorrelated random variables with  $\mathcal{E}\{y_m^2\} = \lambda_m$ , for some  $\lambda_1 \geq \lambda_2 \geq \dots$ . Doing the same expansion with the reconstructed version  $\hat{y}(t)$  of  $y(t)$  we obtain

$$\hat{y}(t) = \sum_{m=1}^{\infty} \hat{y}_m(k) \psi_m(t - (k-1)T). \quad (4)$$

Let  $D_k$ ,  $k = 1, \dots, K$  denote the distortion on the segment  $t \in [(k-1)T, kT]$  (i.e.,  $D_k = \mathcal{E}\{\frac{1}{T} \int_{(k-1)T}^{kT} (y(t) - \hat{y}(t))^2 dt\}$ ). Since the functions  $\psi_m$ ,  $m \in \mathbb{N}$  are orthonormal, it follows from Parseval's identity [21] that:

$$TD_k = \mathcal{E} \left\{ \sum_{m=1}^{\infty} (y_m(k) - \hat{y}_m(k))^2 \right\}. \quad (5)$$

Let  $M \in \mathbb{N}$  and suppose that we use a vector quantizer that acts only on the first  $M$  coefficients  $\mathbf{y}(k) = [y_1(k), \dots, y_M(k)]^T$ , and ignores the remaining coefficients. Then, we have

$$TD_k = \mathcal{E}\{\|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2\} + \sum_{m=M+1}^{\infty} \lambda_m^2$$

where  $\hat{\mathbf{y}}(k) = [\hat{y}_1(k), \dots, \hat{y}_M(k)]^T$ . Hence, the distortion  $D_k$  is given by the sum of the quantization error  $\mathcal{E}\{\|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2\}$  and the truncation error  $\sum_{m=M+1}^{\infty} \lambda_m^2$ . Also, as shown in Appendix B,  $D_k$  decreases monotonically with an increase in  $M$ . Hence, if  $M$  is chosen large enough to guarantee that the truncation error is negligible in comparison with the quantization error, then, from a practical perspective, the quantization of  $y(t)$  is equivalent to that of the  $M$ -dimensional vector process  $\mathbf{y}(k)$ . We will address this problem in Section III-B below.

*Remark 1:* The dimension  $M$  of the vector  $\mathbf{y}(k)$  is chosen such that the truncation error  $\sum_{m=M+1}^{\infty} \lambda_m^2$  is negligible in

comparison with the distortion  $D$ . Also,  $M$  needs to be kept small such that the coding complexity is not unnecessarily increased. Hence, finding the optimal value of  $M$  requires an iterative procedure.

*Remark 2:* In the presentation above we consider  $y(t)$  and  $\psi_m(t)$ ,  $m \in \mathbb{N}$  to be continuous-time functions. This implies that the basis functions  $\psi_m(t)$ , as well as the coefficients  $y_m$ ,  $m \in \mathbb{N}$ , are difficult to compute in practice. However, recall that we assume that in practice, the knowledge of  $y(t)$  is given by its samples obtained at a much higher sampling rate than the transmission rate. Under this assumption, the functions  $\psi_m(t)$  are obtained as the eigenvectors of the covariance matrix of the vector of samples of  $y(t)$ , in the interval  $t \in [(k-1)T, kT]$ , and the coefficients  $y_m$  as inner products between these basis functions and the same vector of samples.

### B. Recursive Quantization of the Vector Process $\mathbf{y}(k)$

1) *Characterization of Quantization Cells for Recursive LDC:* A coding strategy, either UDC or LDC, of the  $K$  (vector) samples  $\mathbf{Y}^{(K)} = [\mathbf{y}^T(1), \dots, \mathbf{y}^T(K)]^T$  induces a partition of the space  $\mathbb{R}^{MK}$  into  $2^{RK}$  cells, each corresponding to a codeword (i.e., a vector in  $\mathbb{R}^{MK}$ ) of some quantization dictionary. In this section, we study the structure of these cells for a recursive LDC scheme [7, Sec. 14.1], (i.e., when  $\hat{\mathbf{y}}(k)$  is computed from the vector  $\mathbf{y}(k)$ , and the previously quantized values  $\hat{\mathbf{y}}(1), \dots, \hat{\mathbf{y}}(k-1)$ ). The difference between the partitions induced by UDC and recursive LDC is illustrated in Fig. 2, using a two-dimensional example, where  $R = 2$ ,  $K = 2$  and  $M = 1$ , i.e.,  $\mathbf{y}(1)$  and  $\mathbf{y}(2)$  are both scalars. Fig. 2(top) shows the  $2^{RK} = 16$  quantization cells and codewords for the UDC design. They are those of an optimal joint quantizer for the vector  $[\mathbf{y}(1), \mathbf{y}(2)]$ . Notice that using this design, both samples  $\mathbf{y}(1)$  and  $\mathbf{y}(2)$  need to be known in order to choose the appropriate codeword  $[\hat{\mathbf{y}}(1), \hat{\mathbf{y}}(2)]$  from the quantization dictionary. On the other hand, Fig. 2(bottom) shows a recursive LDC design. Notice that in this case, the arrangement of quantization cells and dictionary permits choosing the appropriate codeword  $\hat{\mathbf{y}}(1)$  (from a marginal dictionary of  $2^R = 4$  codewords) as soon as  $\mathbf{y}(1)$  becomes available. Furthermore, this choice determines the  $2^R = 4$  quantization cells and codewords for  $\mathbf{y}(2)$ .

Extending the idea above to general values of  $R$ ,  $K$  and  $M$ , it follows that the partitions induced by recursive LDC coding have the following structure:

- For interval  $k = 1$ : A partition of  $\mathbb{R}^M$  is done into  $2^R$  cells  $\mathcal{T}_{i_1}$ ,  $i_1 = 1, \dots, 2^R$ , each corresponding to a quantized (vector) value  $\mathbf{v}_{i_1}$  of the first (vector) sample  $\mathbf{y}(1)$ .
- For interval  $k = 2$ : For each  $i_1 = 1, \dots, 2^R$ , a partition of  $\mathbb{R}^M$  is done into  $2^R$  cells  $\mathcal{T}_{i_1, i_2}$ ,  $i_2 = 1, \dots, 2^R$ , each corresponding to a quantized (vector) value  $\mathbf{v}_{i_1, i_2}$  of the second (vector) sample  $\mathbf{y}(2)$ , given that  $\mathbf{y}(1) \in \mathcal{T}_{i_1}$ .
- The procedure continues so that, at time interval  $k$ , for each  $i_1, \dots, i_{k-1} = 1, \dots, 2^R$ , a partition of  $\mathbb{R}^M$  is done into  $2^R$  cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ , each corresponding to a quantized (vector) value  $\mathbf{v}_{i_1, \dots, i_k}$  of the  $k$ -th (vector) sample  $\mathbf{y}(k)$ , given that  $\mathbf{y}(1) \in \mathcal{T}_{i_1}, \dots, \mathbf{y}(k-1) \in \mathcal{T}_{i_1, \dots, i_{k-1}}$ .

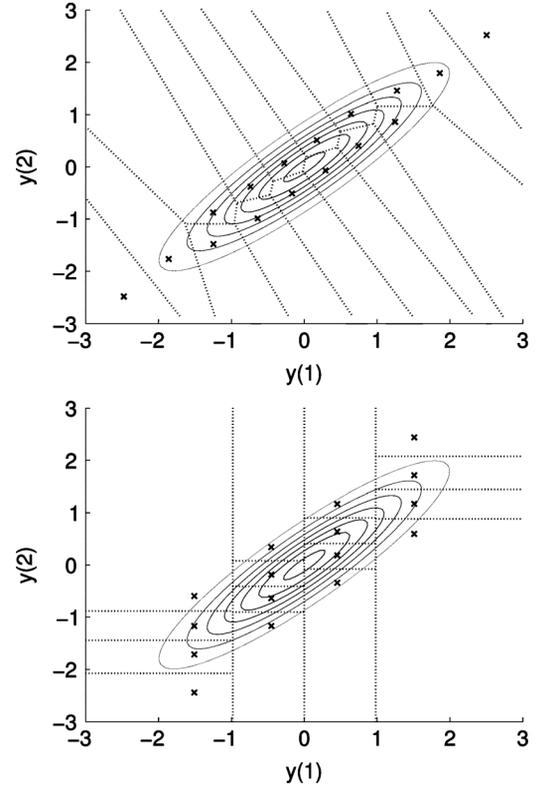


Fig. 2. Quantization cells and dictionary for a Gaussian random vector  $[\mathbf{y}(1), \mathbf{y}(2)]$  using UDC (top) and recursive LDC (bottom).

2) *Design of the Quantization Cells and Codewords:* In order to design the optimal recursive LDC, we need to design, for each  $k$ , and each combination of the indexes  $i_1, \dots, i_{k-1} = 1, \dots, 2^R$ , the quantization cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  and the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  of  $\mathbb{R}^M$ . Let  $\hat{\mathbf{Y}}^{(k)} = \{\hat{\mathbf{y}}(1), \hat{\mathbf{y}}(2), \dots, \hat{\mathbf{y}}(k)\}$  and  $\mathcal{S}^{(i_1, \dots, i_k)} = \mathcal{T}_{i_1} \times \mathcal{T}_{i_1, i_2} \times \dots \times \mathcal{T}_{i_1, \dots, i_{k-1}}$ . Then, as shown in Appendix A, we have that

$$\begin{aligned} \mathcal{E}\{\|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2\} &= \sum_{\hat{\mathbf{Y}}^{(k-1)}} p(\hat{\mathbf{Y}}^{(k-1)}) \\ &\quad \times \int_{\mathbb{R}^M} \|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2 p \\ &\quad \times \left( \mathbf{y}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) d\mathbf{y}(k) \quad (6) \\ &= \sum_{i_1, \dots, i_{k-1}=1}^{2^R} p(\hat{\mathbf{Y}}^{(k-1)}) \\ &\quad \times \sum_{i_k=1}^{2^R} \int_{\mathcal{T}_{i_1, \dots, i_k}} \|\mathbf{y}(k) - \mathbf{v}_{i_1, \dots, i_k}\|^2 p \\ &\quad \times \left( \mathbf{y}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) d\mathbf{y}(k) \quad (7) \end{aligned}$$

where

$$p(\hat{\mathbf{Y}}^{(k-1)}) = \int_{\mathcal{S}^{(i_1, \dots, i_{k-1})}} p(\mathbf{Y}^{(k-1)}) d\mathbf{Y}^{(k-1)}, \quad (8)$$

$$p(\mathbf{y}(k) | \hat{\mathbf{Y}}^{(k-1)}) = \frac{\int_{\mathcal{S}^{(i_1, \dots, i_{k-1})}} p(\mathbf{Y}^{(k)}) d\mathbf{Y}^{(k-1)}}{p(\hat{\mathbf{Y}}^{(k-1)})}. \quad (9)$$

In view of (6), the distortion in the time interval  $k$  (i.e., in the continuous-time interval  $[(k-1)T, kT]$ ) depends on the conditional probability of the sample  $y(k)$  given the past quantization values  $\hat{\mathbf{Y}}^{(k-1)}$ . It is clear from (7) that, if the quantization cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ , for the interval  $k$  are given, the optimal quantization codeword  $\mathbf{v}_{i_1, \dots, i_k}$  is the centroid of each cell under the conditional probability  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$ <sup>3</sup>. Hence, only the quantization cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  need to be designed.

Form (7)–(9), we see that the distortion at time interval  $k$  depends not only on the quantization cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  for that interval (via the integration regions), but also on the choice of the cells at previous intervals (via  $p(\hat{\mathbf{Y}}^{(k-1)})$  and  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$ ). A natural question then is whether, at each  $k$ , we can design the  $2^R$  cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  simply considering the pdf  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  of  $\mathbf{y}(t)$  conditioned to the past quantized values (we call this a *greedy* design), or instead, we need to consider the joint pdf  $p(\mathbf{y}(1), \dots, \mathbf{y}(K))$  of all  $K$  samples, to jointly design all  $2^{KR}$  cells  $\mathcal{T}_{i_1}, \mathcal{T}_{i_1, i_2}, \dots, \mathcal{T}_{i_1, \dots, i_K}$ ,  $i_k = 1, \dots, 2^R$ ,  $k = 1, \dots, K$  (we call this a *joint* design). The main advantage of the greedy design is that, if it is optimal, we can devise a simple method for quantizing  $\mathbf{y}(k)$  recursively. Unfortunately, it turns out that a greedy design is not optimal<sup>4</sup>. This is shown by Example 1 below.

*Example 1:* Consider the LDC design depicted in Fig. 2(right), where  $K = 2$ ,  $R = 2$  and  $M = 1$ , i.e.,  $\mathbf{y}(k)$ ,  $k = 1, 2$ , are scalar discrete-time samples. Let  $\mathbf{y}(1) \sim \mathcal{N}(0, 1)$  (i.e., have a Gaussian distribution with zero mean and unit variance), and consider the particular case where  $\mathbf{y}(1) = \mathbf{y}(2)$ . We consider the greedy and joint designs described above. For the greedy design, the quantization cell boundaries for  $\mathbf{y}(1)$  are computed using Lloyd's algorithm [7]. Then, for each value of  $\hat{\mathbf{y}}(1) \in \{\mathbf{v}_1, \dots, \mathbf{v}_4\}$ , the boundaries for  $\mathbf{y}(2)$  are computed using the same algorithm, considering the pdf  $p(\mathbf{y}(2)|\hat{\mathbf{y}}(1))$  of  $\mathbf{y}(2)$  conditioned on the previously coded value  $\hat{\mathbf{y}}(1)$  of  $\mathbf{y}(1)$ . The quantization boundaries so obtained are shown in Fig. 3 (the figure only shows the boundaries on the positive axis, since those on the negative axis are symmetric). The figure also shows the distortions obtained for  $\mathbf{y}(1)$  and  $\mathbf{y}(2)$ . Notice that this design minimizes the distortion of the first sample, but not necessarily the total distortion. This is the goal of the joint design, in which the quantization cell boundaries of both samples are jointly optimized to minimize the total distortion. The optimization is carried out using a quasi-Newton (BFGS) procedure. In order to evaluate whether the optimization procedure gets stuck into a local minimum, we carry out ten quasi-Newton parallel searches, which are initialized using boundaries with randomly chosen locations. It turns out that all quasi-Newton searches yield the same result. The boundaries and distortions resulting from the joint design are shown in Fig. 4. We see that while the distortion of the first sample is higher than the one obtained with the greedy design, the total distortion of the joint design is smaller.

<sup>3</sup>This follows from the same argument used in the centroid optimality condition for vector quantization [7], by replacing the pdf  $p(\mathbf{y}(k))$  of  $\mathbf{y}(k)$  by the conditional pdf  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$ .

<sup>4</sup>The theoretical possibility of this being the case was pointed out in [7, p. 524], in the context of recursive vector quantization.

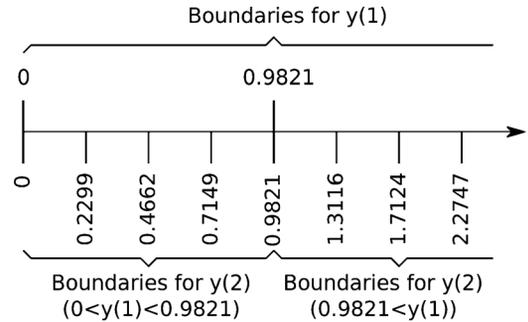


Fig. 3. Quantization cell boundaries and distortion for the greedy encoder.

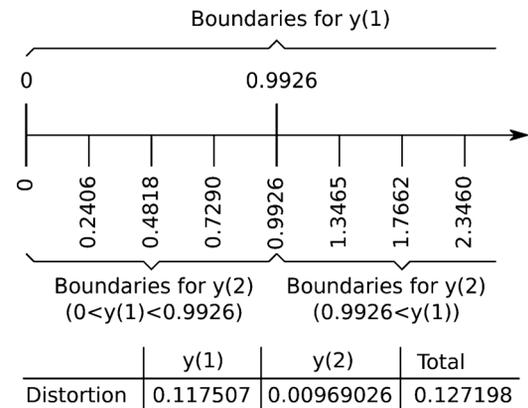


Fig. 4. Quantization cell boundaries and distortion for the joint encoder.

Example 1 indicates that the greedy method is not optimal. Hence, the quantization cells for all time intervals  $k = 1, \dots, K$  need to be jointly designed using  $p(\mathbf{y}(1), \dots, \mathbf{y}(K))$ . This can be a very cumbersome task. Luckily, numerical studies demonstrate that, when the samples have a jointly Gaussian distribution, the advantage obtained by the joint method is very small in comparison with the greedy method. This is illustrated in the following simple example.

*Example 2:* Consider the coding of  $K$  samples taken from a discrete-time scalar random process (i.e.,  $M = 1$ )  $\mathbf{y}(t)$ , generated by filtering discrete-time Gaussian white noise using the filter  $g(z) = 1/(1 - az^{-1})$ . We compare the distortion  $D_g$  obtained using the greedy design, with the one  $D_j$  resulting from the joint design, as described in Example 1. Fig. 5 shows the distortion per sample (i.e.,  $D_g/K$  and  $D_j/K$ ) resulting from both methods, for different values of the pole  $a$  of the filter  $g(z)$ . (Notice that the value of  $a$  determines the level of correlation between consecutive samples. Notice also that, when samples are not correlated both, the greedy and joint design, yield the same result). The figure also shows the difference  $(D_g - D_j)/K$  per sample, which measures the improvement offered by the joint design. We see that this improvement is indeed negligible for all values of  $a$ . Fig. 6 shown the same comparison for different number of quantization levels. (Notice that we do not constraint the plot to only those values which are powers of two). Also in this case we see that the improvement given by the joint design is negligible.

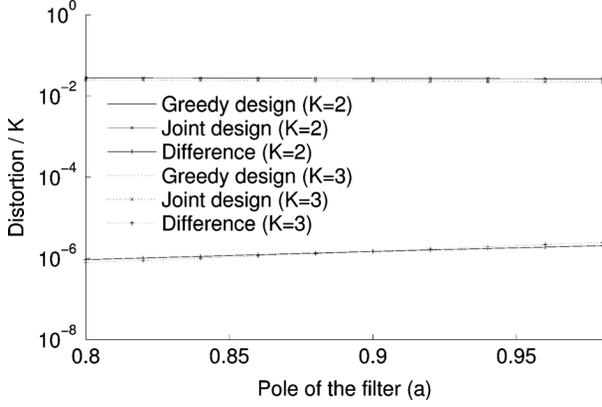


Fig. 5. Distortion comparison between greedy and joint encoders, for different values of  $a$ .

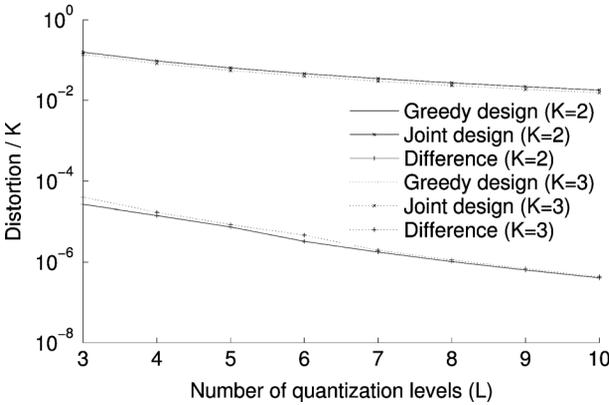


Fig. 6. Distortion comparison between greedy and joint encoders, for different quantization levels.

In view of Example 2, and since we are only concerned about coding of Gaussian signals, we adopt the greedy approach to design the quantization cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  for each time interval  $k$ . Notice that in this case, designing the cells  $\mathcal{T}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  is equivalent to designing the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ . This equivalence follows from the nearest neighbor optimality condition for vector quantization [7], which states that the quantization cell  $\mathcal{T}_{i_1, \dots, i_k}$  of each  $\mathbf{v}_{i_1, \dots, i_k}$  is formed by those vectors  $\mathbf{y}(k)$  which are closer to  $\mathbf{v}_{i_1, \dots, i_k}$  than to any other codeword. With this in mind, we state the proposed LDC strategy in Section III-C.

### C. Resulting LDC Procedure

Following the analysis above, a suboptimal recursive LDC strategy is obtained by carrying out, at the  $k$ -th time interval, the following four steps:

(S1) Use a KL decomposition to obtain the vector of coefficients  $\mathbf{y}(k)$ , representing the signal  $y(t)$  in the interval  $[(k-1)T, kT]$ ;

(S2) Compute the conditional pdf  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  using the past quantized values  $\hat{\mathbf{Y}}^{(k-1)}$ ;

(S3) Use  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  to compute the quantization codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ ;

(S4) Quantize  $\mathbf{y}(k)$  by choosing  $\hat{\mathbf{y}}(k)$  as the codeword which is the closest to  $\mathbf{y}(k)$ .

The main burden in the strategy above lies in Steps (S2) and (S3). These are to be studied next.

*Remark 3:* The strategy described above redesigns the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  at each time interval  $k$ . A natural question is whether the decoder is able to reproduce these codewords for reconstructing  $\hat{\mathbf{y}}(k)$ . Notice that the codewords depend on conditional probabilities given the past quantized values  $\hat{\mathbf{Y}}^{(k-1)}$ . Since  $\hat{\mathbf{Y}}^{(k-1)}$  is available at the decoder, it is able to reproduce the codewords, making the reconstruction of  $\hat{\mathbf{y}}(k)$  possible.

*Remark 4:* As mentioned in Section I, our strategy for coding the sequence of vectors  $\mathbf{y}(k)$  [i.e., steps (S2) to (S4)], is equivalent to a sequential quantization scheme [9], where instead of the scalar components of a finite dimensional vector, an infinite sequence of vectors is quantized. This prevents us from expressing the conditional pdf  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  either analytically, or using a training sequence. Instead, we use a Bayesian tracking procedure, which we describe in the next section.

## IV. PROPOSED LDC ALGORITHM

In this section, we describe the numerical implementation of the steps (S2) and (S3) mentioned above.

### A. State Space Realization for pdf Tracking

Step (S2) requires the computation of the conditional pdf  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  at each  $k$ , given the past quantized values. We introduce below a recursive scheme for doing so, using a state space realization of the vector process  $\mathbf{y}(k)$ .

Recall (3) and let  $\boldsymbol{\psi}(k) = [\psi_1(k), \dots, \psi_M(k)]^T$ . Then, it is easy to show that the autocorrelation  $\mathbf{R}(k)$  of  $\mathbf{y}(k)$  is given by

$$\mathbf{R}(k) = (\boldsymbol{\psi} * r_x * \boldsymbol{\psi}^*)(kT) \quad (10)$$

where  $*$  denotes convolution and the superscript  $*$  denotes the transpose, time-reversal operation (i.e.,  $\boldsymbol{\psi}^*(t) = \boldsymbol{\psi}^T(-t)$ ). Now, since  $y(t)$  is Gaussian, using some spectral realization method [22] we can build a state space model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (11)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (12)$$

$$\hat{\mathbf{y}}(k) = \mathcal{Q}_k(\mathbf{y}(k)) \quad (13)$$

so that the autocorrelation of  $\mathbf{y}(k)$  equals (10). In (11)–(13),  $\mathbf{x}(k) \in \mathbb{R}^N$  denotes the state vector at time interval  $k$ ,  $\mathbf{A} \in \mathbb{R}^{N \times N}$ ,  $\mathbf{B} \in \mathbb{R}^{N \times M}$ ,  $\mathbf{C} \in \mathbb{R}^{M \times N}$ ,  $\mathcal{Q}_k$  denotes the quantizer at  $k$ , and  $\mathbf{u}(k)$  is a sequence of independent,  $M$ -dimensional random vectors with distribution  $\mathcal{N}(0, I)$ . Also, to satisfy the stationarity condition, we assume that the initial state  $\mathbf{x}(1)$  has distribution  $\mathcal{N}(0, \Sigma)$ , where  $\Sigma$  is the solution of  $\Sigma = \mathbf{A}\Sigma\mathbf{A}^T + \mathbf{B}\mathbf{B}^T$ .

Using the state space model (11)–(13), we can recursively compute the conditional pdf  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  using a Bayesian tracking procedure [23], [24]. More precisely, we first compute  $p(\mathbf{x}(k)|\hat{\mathbf{Y}}^{(k-1)})$  using the following recursive formulas:

$$\begin{aligned} & p(\mathbf{x}(k)|\hat{\mathbf{Y}}^{(k-1)}) \\ &= \int_{\mathbb{R}^M} p(\mathbf{x}(k)|\mathbf{x}(k-1))p \end{aligned} \quad (14)$$

$$\begin{aligned}
& \times \left( \mathbf{x}(k-1) | \hat{\mathbf{Y}}^{(k-1)} \right) d\mathbf{x}(k-1) \\
p \left( \mathbf{x}(k) | \hat{\mathbf{Y}}^{(k)} \right) &= \frac{p \left( \hat{\mathbf{y}}(k) | \mathbf{x}(k) \right) p \left( \mathbf{x}(k) | \hat{\mathbf{Y}}^{(k-1)} \right)}{\int_{\mathbb{R}^M} p \left( \hat{\mathbf{y}}(k) | \mathbf{x}(k) \right) p \left( \mathbf{x}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) d\mathbf{x}(k)} \quad (15)
\end{aligned}$$

where  $p(\mathbf{x}(k)|\mathbf{x}(k-1)) \sim \mathcal{N}(\mathbf{A}\mathbf{x}(k-1), \mathbf{B}\mathbf{B}^T)$  and

$$p(\hat{\mathbf{y}}(k)|\mathbf{x}(k)) = \begin{cases} 1, & \mathcal{Q}_k(\mathbf{C}\mathbf{x}(k)) = \hat{\mathbf{y}}(k) \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

Then,  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  is computed from  $p(\mathbf{x}(k)|\hat{\mathbf{Y}}^{(k-1)})$  as follows:

$$\begin{aligned}
p \left( \mathbf{y}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) &= \int_{\mathbb{R}^M} p(\mathbf{y}(k)|\mathbf{x}(k)) p \left( \mathbf{x}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) d\mathbf{x}(k) \\
&= \int_{\mathbf{x}(k):\mathbf{y}(k)=\mathbf{C}\mathbf{x}(k)} p \left( \mathbf{x}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) d\mathbf{x}(k). \quad (17)
\end{aligned}$$

### B. Implementation Using Particle Filtering

The implementation of the recursive formulas (14)–(17) required for step (S2), and the design of the optimal codewords in (S3) are both numerically complex. We derive below a numerically tractable algorithm for approximately implementing these tasks using particle filtering [24].

For a given  $k \in \mathbb{N}$ , let  $\mathbf{X}^{(k)} = [\mathbf{x}^T(1), \dots, \mathbf{x}^T(k)]^T \in \mathbb{R}^{kN}$  and  $\mathbf{Y}^{(k)} = [\mathbf{y}^T(1), \dots, \mathbf{y}^T(k)]^T \in \mathbb{R}^{kM}$ . Fix  $k$ , and suppose that the conditional distribution  $p(\mathbf{X}^{(k)}|\hat{\mathbf{Y}}^{(k-1)})$  is known. We describe below an iterative algorithm which uses the (approximate) knowledge of  $p(\mathbf{X}^{(k)}|\hat{\mathbf{Y}}^{(k-1)})$  to build an approximation of  $p(\mathbf{X}^{(k+1)}|\hat{\mathbf{Y}}^{(k)})$ . Each iteration is formed by a number of steps which are detailed below. The initialization of the iterations is addressed subsequently.

**Computing (17):** The idea is to approximate the distribution  $p(\mathbf{X}^{(k)}|\hat{\mathbf{Y}}^{(k-1)})$  by a sum of  $I$  impulses (*particles*) located at some points  $\mathbf{X}_i^{(k)} = [\mathbf{x}_i^T(1), \dots, \mathbf{x}_i^T(k)]^T \in \mathbb{R}^{kN}$ ,  $i = 1, \dots, I$ , i.e.

$$p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k-1)} \right) \simeq \frac{1}{I} \sum_{i=1}^I \delta \left( \mathbf{X}^{(k)} - \mathbf{X}_i^{(k)} \right). \quad (18)$$

The particle locations  $\mathbf{X}_i^{(k)}$ ,  $i = 1, \dots, I$  are obtained from  $I$  random samples of the distribution  $p(\mathbf{X}^{(k)}|\hat{\mathbf{Y}}^{(k-1)})$ . Now, from (18), (17) becomes

$$p \left( \mathbf{y}(k) | \hat{\mathbf{Y}}^{(k-1)} \right) \simeq \frac{1}{I} \sum_{i=1}^I \delta \left( \mathbf{y}(k) - \mathbf{C}\mathbf{x}_i(k) \right). \quad (19)$$

**Computing (S3):** In view of (19), the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ , for the time interval  $k$ , can be obtained by running the k-means algorithm [7] on the samples  $\mathbf{C}\mathbf{x}_i(k)$ .

**Computing (15):** We have

$$\begin{aligned}
p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k)} \right) &= \frac{p \left( \hat{\mathbf{y}}(k), \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k-1)} \right)}{p \left( \hat{\mathbf{y}}(k) | \hat{\mathbf{Y}}^{(k-1)} \right)} \\
&= \frac{p \left( \hat{\mathbf{y}}(k) | \mathbf{X}^{(k)}, \hat{\mathbf{Y}}^{(k-1)} \right) p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k-1)} \right)}{p \left( \hat{\mathbf{y}}(k) | \hat{\mathbf{Y}}^{(k-1)} \right)} \\
&= \frac{p \left( \hat{\mathbf{y}}(k) | \mathbf{x}(k) \right)}{p \left( \hat{\mathbf{y}}(k) | \hat{\mathbf{Y}}^{(k-1)} \right)} p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k-1)} \right). \quad (20)
\end{aligned}$$

Now, using (16) and the approximation (18), we have that  $p(\mathbf{X}^{(k)}|\hat{\mathbf{Y}}^{(k)})$  could in principle be approximated by constraining the sum in (18) to only those impulses such that  $\mathcal{Q}_k(\mathbf{C}\mathbf{x}_i(k)) = \hat{\mathbf{y}}(k)$ , i.e.

$$p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k)} \right) \simeq \frac{1}{I_k} \sum_{\substack{i=1: \\ \mathcal{Q}_k(\mathbf{C}\mathbf{x}_i(k))=\hat{\mathbf{y}}(k)}}^I \delta \left( \mathbf{X}^{(k)} - \mathbf{X}_i^{(k)} \right) \quad (21)$$

where  $I_k = \sum_{i=1}^I p(\hat{\mathbf{y}}(k)|\mathbf{x}_i(k))$  equals the number of such impulses. However, notice that doing so reduces the number of particles from  $I$  to  $I_k$ . This would cause that the iterations would have eventually no particle left. To prevent this, we obtain a new set of  $I$  samples  $\mathbf{X}_i^{(k)}$ ,  $i = 1, \dots, I$ , by drawing them randomly from the discrete distribution in (21). These samples would obviously have repetitions if  $I_k < I$ . Doing so we obtain

$$p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k)} \right) \simeq \frac{1}{I} \sum_{i=1}^I \delta \left( \mathbf{X}^{(k)} - \mathbf{X}_i^{(k)} \right). \quad (22)$$

**Computing (14):** The last step consists in using the points  $\mathbf{X}_i^{(k)} \in \mathbb{R}^{kN}$ ,  $i = 1, \dots, I$ , to obtain points  $\mathbf{X}_i^{(k+1)} \in \mathbb{R}^{(k+1)N}$ ,  $i = 1, \dots, I$  for building an approximation of  $p(\mathbf{X}^{(k+1)}|\hat{\mathbf{Y}}^{(k)})$  as in (14). We have that

$$\begin{aligned}
p \left( \mathbf{X}^{(k+1)} | \hat{\mathbf{Y}}^{(k)} \right) &= p \left( \mathbf{x}(k+1) | \mathbf{X}^{(k)}, \hat{\mathbf{Y}}^{(k)} \right) p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k)} \right) \\
&= p \left( \mathbf{x}(k+1) | \mathbf{x}(k) \right) p \left( \mathbf{X}^{(k)} | \hat{\mathbf{Y}}^{(k)} \right). \quad (23)
\end{aligned}$$

Hence, for each  $i = 1, \dots, I$ , we can obtain a points  $\mathbf{X}_i^{(k+1)}$  by simply adding to the point  $\mathbf{X}_i^{(k)}$  an extra component  $\mathbf{x}_i(k+1)$ . This component is obtained by a random sampling of the distribution  $p(\mathbf{x}(k+1)|\mathbf{x}_i(k)) \sim \mathcal{N}(\mathbf{A}\mathbf{x}_i(k), \mathbf{B}\mathbf{B}^T)$ .

**Initialization:** For  $k = 1$ , we have that  $p(\mathbf{X}^{(k+1)}|\hat{\mathbf{Y}}^{(k)}) = p(\mathbf{x}(1))$ . Hence, we approximate it by the following sum of impulses

$$p(\mathbf{x}(1)) \simeq \frac{1}{I} \sum_{i=1}^I \delta(\mathbf{x}(1) - \mathbf{x}_i(1)),$$

where the impulse locations  $\mathbf{x}_i(1)$ ,  $i = 1, \dots, I$  are obtained as random samples of the distribution  $\mathcal{N}(0, \Sigma)$  of  $\mathbf{x}(1)$ .

In the explanation above we have used particles to approximate joint conditional pdfs. Each particle  $\mathbf{X}_i^{(k)}$  is represented by

a vector in  $\mathbb{R}^{kN}$ . Hence, there seems to be a dimension increase with  $k$ . However, this is not the case. Notice that, due to the Markov property of the state space model (11)–(13), we only need to keep track of the last component  $\mathbf{x}_i(k)$  of each  $\mathbf{X}_i^{(k)}$ ,  $i = 1, \dots, I$ , avoiding a memory growth of the algorithm. With this consideration, the resulting algorithm is summarized here.

---

### Algorithm 1

---

Start by drawing  $\mathbf{x}_i(1)$ ,  $i = 1, \dots, I$  from the distribution  $\mathcal{N}(0, \Sigma)$  of the initial state  $\mathbf{x}(0)$ . Then, for each  $k \in \mathbb{N}$ ,

- 1) Compute the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$  using the k-means algorithm and the points  $\mathbf{C}\mathbf{x}_i(k)$ .
- 2) Choose  $\hat{\mathbf{y}}(k) = \mathbf{v}_{i_1, \dots, i_k}$ , where  $i_k = \arg \min_j \|\mathbf{y}(k) - \mathbf{v}_{i_1, \dots, j}\|_2$  (i.e., choose the codeword closest to  $\mathbf{y}(k)$ ).
- 3) Keep only the set of  $I_k$  points  $\mathbf{x}_{i_n}(k)$ ,  $n = 1, \dots, I_k$  satisfying

$$i_k = \arg \min_j \|\mathbf{C}\mathbf{x}_{i_n}(k) - \mathbf{v}_{i_1, \dots, j}\|_2,$$

i.e., such that  $\mathbf{C}\mathbf{x}_{i_n}(k)$  is closer to  $\mathbf{v}_{i_1, \dots, i_k}$  than to any other codeword.

- 4) Obtain  $\mathbf{x}_i(k+1)$ ,  $i = 1, \dots, I$  by doing  $I$  random choices (with possible repetitions) from the set  $\mathbf{x}_{i_n}(k)$ ,  $n = 1, \dots, I_k$ , defined in Step 3.
  - 5) For each  $i = 1, \dots, I$ , draw  $\mathbf{x}_i(k+1)$  from the distribution  $\mathcal{N}(\mathbf{A}\mathbf{x}_i(k), \mathbf{B}\mathbf{B}^T)$ .
- 

*Remark 5:* Fig. 1(right) shows a dictionary of continuous signals for LDC. Notice that recursively designing the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ , for each time interval  $k$ , using the k-means algorithm, does not guarantee that the signals of the resulting dictionary are continuous on the boundaries of the intervals  $[(k-1)T, kT]$ ,  $k = 1, 2, \dots$ . To guarantee this, the codewords need to be chosen under the constraints  $[\psi_1(0), \dots, \psi_M(0)]\mathbf{v}_{i_1, \dots, i_k} = \hat{\mathbf{y}}^-(k-1)T$ , for all  $i_k = 1, \dots, 2^R$ , where  $\hat{\mathbf{y}}^-(k-1)T$  denotes the value of the reconstructed signal segment  $\hat{\mathbf{y}}((k-1)T)$ ,  $t \in [(k-2)T, (k-1)T]$ , at the boundary  $(k-1)T$ . Similar constraints can be imposed to guarantee the continuity of the derivatives of  $\hat{\mathbf{y}}(t)$  at the interval boundaries.

*Remark 6:* As mentioned in Remark 3, the decoder needs to reproduce the codewords  $\mathbf{v}_{i_1, \dots, i_k}$ ,  $i_k = 1, \dots, 2^R$ , at time interval  $k$ , in order to reconstruct  $\hat{\mathbf{y}}(k)$ . In Algorithm 1, these codewords are derived from the set of particles  $\mathbf{x}_i(k)$ ,  $i = 1, \dots, I$ , which are randomly generated. In practice, a pseudo-random generator needs to be used so that the same particles can be generated at both encoder and decoder ends.

### C. Complexity Analysis

In this section we study the numerical complexity of Algorithm 1, proposed in Section IV-B. We summarize below the operations requiring floating point multiplications. These operations need to be carried out at both the encoder and the decoder, unless explicitly stated, and at each time interval (i.e., once every  $T$  seconds).

- 1) The KL transform at the encoder and the KL inverse transform at the decoder require each  $MF_iT$  multiplications, where  $F_i$  denotes the (super-Nyquist) frequency used for sampling  $y(t)$  before processing.
- 2) The k-means algorithm with  $I$  points of  $M$  dimensions and  $2^R$  clusters, requires  $IJM(2^R + 1)$  multiplications, where  $J$  denotes the average number of iterations required for convergence.
- 3) Coding the vector  $\mathbf{y}(k)$  requires  $M2^R$  multiplications (this is only done at the encoder).
- 4) The random particle choices carried out in step 4 of Algorithm 1 requires generating  $I$  uniform random variables, and the generation of  $\mathbf{u}(k)$  requires generating  $IM$  Gaussian variables.
- 5) Computing  $\mathbf{x}_i(k+1) = \mathbf{A}\mathbf{x}_i(k) + \mathbf{B}u(k)$  and  $\mathbf{y}_i(k) = \mathbf{C}\mathbf{x}_i(k)$  requires  $IN(N + 2M)$  multiplications.

From the tasks above, the complexity of tasks 2 and 5 are dominant. Hence, at each time interval, the number of multiplication  $\mu$  at both the encoder and the decoder is approximately given by

$$\mu \simeq I[N(N + 2M) + JM(2^R + 1)]. \quad (24)$$

## V. NUMERICAL EXPERIMENTS

In order to evaluate the performance of the proposed LDC algorithm, we compare it with a number of standard quantization techniques. For the comparison we use a random process generated by filtering Gaussian white noise using a fifth-order Butterworth filter with cutoff frequency  $F_c = 0.25$  Hz. The power spectral density (PSD) of the resulting signal is shown in Fig. 7 (left). While in theory, the signal  $y(t)$  is of continuous-time, we implement it as a discrete-time signal with sampling frequency  $F_i = 100$  Hz, which is much higher than the Nyquist rate of  $y(t)$ . We generate samples spanning 1000 seconds. We describe below the quantization methods used in the comparison. Within the description of each method, we include the complexity associated with the decoding task. We consider this complexity instead of that of the encoding task, because the former is higher than or equal to the latter, in all cases.

**Sampling and scalar quantization (SMP + SQ):** This method samples the continuous-time signal  $y(t)$  using a super-Nyquist rate of  $F_s = 1$  Hz. Then, each sample is quantized using a (non-uniform) optimal scalar quantizer. This quantizer is designed using Lloyd's algorithm ([7], Section 6.4). The reconstruction  $\hat{\mathbf{y}}(t)$  of the continuous-time signal from the quantized samples is done using a filter derived from a non-causal infinite impulse response (IIR) filter  $f(z)$ . Hence, to achieve a prescribed reconstruction delay  $T$ , we truncate the non-causal component of the impulse response  $f(t)$  of  $f(z)$  so that  $f(t) = 0$ , for  $t < -T$ . In order to reduce the error introduced by such truncation, we choose  $f(z)$  to be a raised cosine filter with cutoff frequency 0.5 Hz and roll-off factor  $\beta = 0.3$ . The frequency and impulse responses of this filter are shown in Fig. 7. Finally, to reduce computations, we also truncate the causal component of  $f(t)$  so that  $f(t) = 0$ , for  $t > 10$  sec. The decoding complexity of this method is determined by the reconstruction process, and

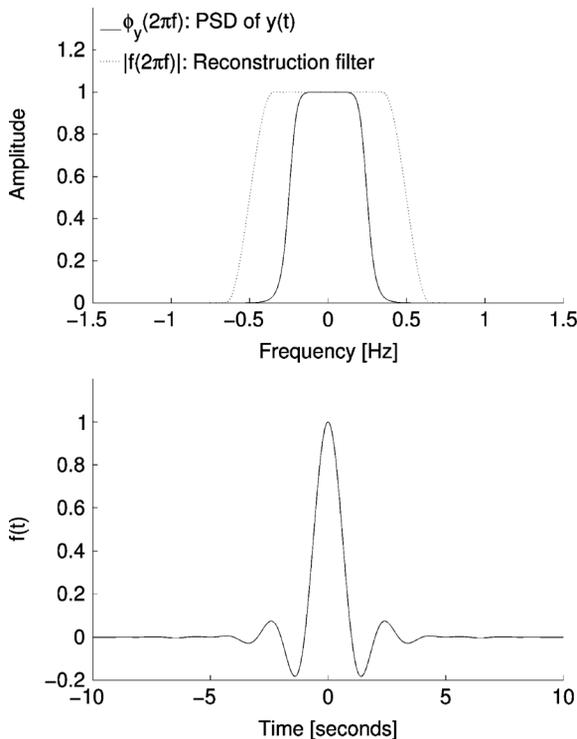


Fig. 7. (Top) PSD of  $y(t)$  and frequency response of the reconstruction filter. (Bottom) Impulse response of the reconstruction filter.

is given by  $100T + 1000$  multiplications per sample (i.e., per second).

**Sampling and linear predictive quantization (SMP + LPQ):** This method is similar to the SMP+SQ method described above, with the only difference in the quantization stage. This consists of a linear predictive quantizer, with a linear predictor of 10th order. The design of the quantizer is done using the method described in [25], which is summarized in Appendix C. To do so we use a training signal with  $2^{9+R/T}$  samples. The decoding complexity of this method is due to the reconstruction task and the prediction, i.e.,  $100T + 1010$  multiplications per sample (i.e., per second).

**KL decomposition and scalar quantization (KL + SQ):** This method uses a KL decomposition to obtain a sequence  $\mathbf{y}(k)$  of vector coefficients of  $y(t)$ , as explained in Section III-A. The number  $M$  of components of the vectors  $\mathbf{y}(k)$  is chosen as the smallest number of components yielding a truncation error at least 50 dB smaller than the total power. Then, each component  $y_m(k)$ ,  $m = 1, \dots, M$  of  $\mathbf{y}(k)$  is quantized using the scalar scheme described in SMP+SQ. To do so, the  $R$  bits available to quantize the vector  $\mathbf{y}(k)$  need to be allocated over its  $M$  components. We do so using the greedy bit allocation algorithm described in ([7], Section 8.4). In this method, the  $R$  bits are sequentially allocated by assigning, at each step, one additional bit to the component having the highest reconstruction error. The decoding complexity of this method is due to the KL transform, and is given by  $MF_iT$  multiplications every  $T$  seconds.

**KL decomposition and linear predictive quantization (KL + LPQ):** This method is similar to the KL+SQ method, with the only difference in how the components  $y_m(k)$ ,

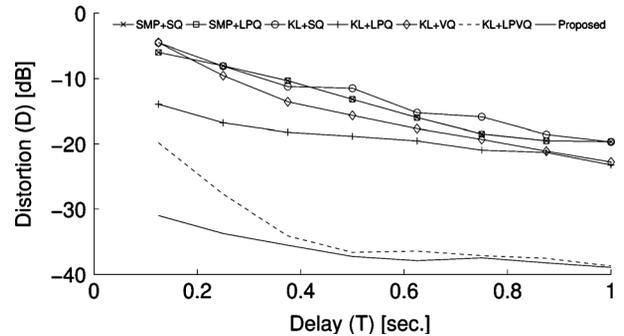


Fig. 8. Distortion comparison for different reconstruction delays  $T$ , and fixed bit rate  $\bar{R} = 8$ .

$m = 1, \dots, M$  are quantized. This is done using the linear predictive method described in SMP + LPQ. The decoding complexity is given by the KL transform and the prediction, i.e.,  $M(F_iT + 10)$  multiplications every  $T$  seconds.

**KL decomposition and vector quantization (KL + VQ):** This method uses a KL decomposition as described for the KL + SQ method. Then, each vector  $\mathbf{y}(k)$  is jointly quantized using an optimal vector quantizer. This quantizer is designed using the generalized Lloyd's algorithm [7, Sec. 11.3] and a training signal of  $2^{9+R}$  samples. The decoding complexity of this method is given by the KL transform, i.e.,  $MF_iT$  multiplications every  $T$  seconds.

**KL decomposition and linear predictive vector quantization (KL + LPVQ):** This method is similar to the KL+VQ method, with the only difference in how the vectors  $\mathbf{y}(k)$  are quantized. This is done using a linear predictive vector quantizer. The quantizer is designed using the method in [25], and summarized in Appendix C. Following [25], we use a first-order vector linear predictor in KL-LPVQ, since higher orders lead to instability during the design procedure. The decoding complexity of this method is given by the KL transform and linear prediction, i.e.,  $M(F_iT + M)$  multiplications every  $T$  seconds.

**Proposed method:** The proposed method uses the KL transform described for the KL + SQ method. For obtaining the state-space model (11)–(12) we use the algorithm described in Appendix D. Also, for the pdf tracking and codeword design algorithm described in Section IV.B, we use  $I = 2^{7+R}$  particles.

In the first simulation, we compare the distortion of the proposed method to those of the methods listed above. To do so, we use a fixed rate of  $\bar{R} = R/T = 8$  bits per second, and we vary the reconstruction delay  $T$  from 0.125 to 1 s (so that the quantization of  $\mathbf{y}(k)$  is done using 1 to 8 bits). The result of this comparison is shown in Fig. 8. For comparison purposes, we point out that the rate distortion function of the continuous-time signal  $y(t)$ , evaluated at  $\bar{R} = 8$ , equals  $-45.55$  dB. We see that the distortion of the KL + LPVQ and the proposed methods are noticeably smaller than those of the other methods. Also, the proposed method outperforms the KL + LPVQ method for low reconstruction delays. The reason for this is discussed in the next paragraph.

As shown in Section III-B2, a nearly optimal quantization strategy is achieved by designing the vector quantizer  $Q_k$  for each vector sample  $\mathbf{y}(k)$  using the conditional pdf

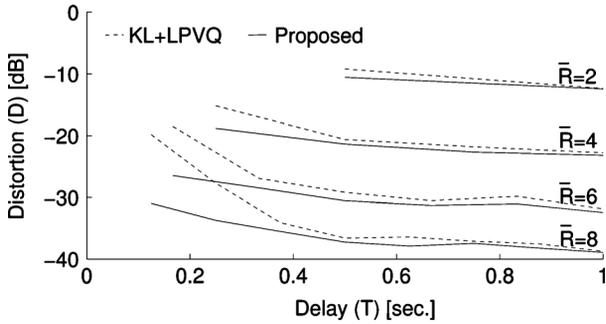


Fig. 9. Comparison between the KL + LPVQ and the proposed methods, for different reconstruction delays  $T$  and bit rates  $\bar{R}$ .

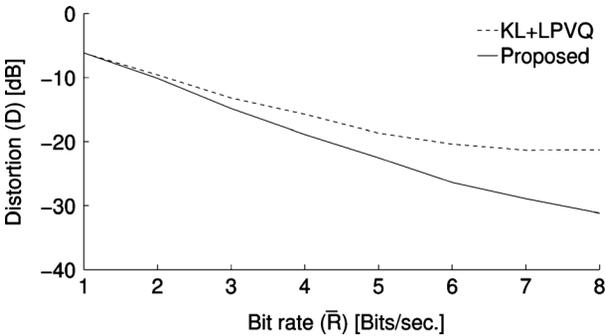


Fig. 10. Comparison between the KL + LPVQ and the proposed methods, for different bit rates  $\bar{R}$  and minimum delay  $T = 1/\bar{R}$ .

$p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  of that sample, given the previous quantization values. This requires  $\mathcal{Q}_k$  to be redesigned at each  $k$ , as done in the proposed method. Instead of doing so, the KL-LPVQ method uses  $\mathcal{Q}_k(\mathbf{y}(k)) = \bar{\mathcal{Q}}(\mathbf{y}(k) - \mathbf{y}_p(k)) + \mathbf{y}_p(k)$ , where  $\mathbf{y}_p(k)$  is the predicted value of  $\mathbf{y}(k)$ , i.e., at each  $k$  it uses the same quantizer  $\bar{\mathcal{Q}}$  (which we design using the method described in Appendix C) whose center is shifted by  $\mathbf{y}_p(k)$ . Now, if there is no quantization (or equivalently, as the number of quantization bits  $R$  tends to infinity),  $p(\mathbf{y}(k)|\hat{\mathbf{Y}}^{(k-1)})$  becomes a Gaussian distribution with mean  $\mathbf{y}_p(k)$  and a covariance matrix which is independent of  $k$ . Hence, the KL+LPVQ method becomes optimal as  $R$  increases; and therefore, the proposed coding method is advantageous when  $R$  is small. When the coding bit rate  $\bar{R}$  is fixed, this (i.e., a small value of  $R$ ) corresponds to a small delay  $T$ .

To see this point in more detail, in the second simulation we repeat the experiment, only involving the KL+LPVQ method and the proposed method, and considering different bit rates  $\bar{R}$ . The result is shown in Fig. 9, showing the advantage of the proposed method for low reconstruction delays.

Finally, in Fig. 10 we compare the distortions of the KL+LPVQ and the proposed methods, for different bit rates  $\bar{R}$ , and for each rate, we use the minimum delay, i.e.,  $T = 1/\bar{R}$ , so that  $\mathbf{y}(k)$  is always quantized with one bit (i.e.,  $R = 1$ ). We see that the advantage of the proposed method becomes more clear at high rates, where the distortion becomes smaller.

Table I shows the details for computing the complexity  $\mu$  of the proposed method, measured in number of multiplications per interval of  $T$  seconds. These details correspond to a rate of  $\bar{R} = 8$  bits per second, and reconstruction delays  $T$  ranging

TABLE I  
DETAILS OF THE COMPUTATION OF THE COMPLEXITY  $\mu$  MULT./ $T$  OF THE PROPOSED METHOD, FOR  $\bar{R} = 8$

T	R	I	J	M	N	$\mu$
0.125	1	256	5.853	2	12	$58.14 \times 10^3$
0.25	2	512	11.13	2	12	$155.3 \times 10^3$
0.375	3	1024	20.93	3	13	$831.6 \times 10^3$
0.5	4	2048	30.46	3	14	$3.755 \times 10^6$

TABLE II  
COMPLEXITY MULT./ $T$  OF ALL OTHER METHODS, FOR  $\bar{R} = 8$

T	SMP+SQ	SMP+LPQ	KL+SQ KL+VQ	KL+LPQ	KL+LPVQ
0.125	126.6	127.8	25	45	29
0.25	256.3	258.8	50	70	54
0.375	389.1	392.8	112.5	142.5	121.5
0.5	525	530	150	180	159

from 0.125 to 0.5 s. Table II shows the complexity of all other methods used in the comparison. We see that the complexity of the proposed method is significantly higher than those of the other methods. Hence, in view of the performance comparison presented above, we conclude that the proposed method is a valid option for encoding under low reconstruction delay (i.e., when transmitting one or two bits at a time), if the extra complexity can be afforded.

## VI. CONCLUSION

We have proposed a fixed-rate encoding method for continuous-time Gaussian signals to reduce the reconstruction distortion for given constraints on the reconstruction delay and data rate. The proposed approach uses a Karhunen-Loève decomposition to obtain a sequence of coefficient vectors, whose innovation is vector-quantized. This is recursively done by considering the conditional pdf of the current coefficient vector, given the past quantized data which is already available at the decoder. While the proposed method achieves a suboptimal reconstruction distortion, numerical experiments show that the difference with the optimal reconstruction is negligible in the Gaussian case, while offering advantages over other available approaches, especially when the reconstruction delay is small.

## APPENDIX A PROOF OF (6)

Since the quantized value  $\hat{\mathbf{y}}(k)$  does not depend on the future values  $\mathbf{y}(k+1), \dots, \mathbf{y}(K)$ , we have

$$\begin{aligned}
 & \mathcal{E} \{ \|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2 \} \\
 &= \int_{\mathbb{R}^{MK}} \|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2 p(\mathbf{Y}^{(K)}) d\mathbf{Y}^{(K)} \\
 &= \int_{\mathbb{R}^{Mk}} \|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2 p(\mathbf{Y}^{(k)}) d\mathbf{Y}^{(k)} \\
 & \quad \times \int_{\mathbb{R}^{M(K-k)}} p(\mathbf{y}(k+1), \dots, \mathbf{y}(K)) d\mathbf{y}(k+1) \cdots d\mathbf{y}(K) \\
 &= \int_{\mathbb{R}^{Mk}} \|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2 p(\mathbf{Y}^{(k)}) d\mathbf{Y}^{(k)}.
 \end{aligned}$$

Now, since  $\hat{\mathbf{y}}(k) = \mathbf{v}_{i_1, \dots, i_k}$  whenever  $\mathbf{y}(k) \in \mathcal{S}^{(i_1, \dots, i_k)}$ , it follows that:

$$\begin{aligned} & \mathcal{E}\{\|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2\} \\ &= \sum_{i_1, \dots, i_k=1}^{2^R} \int_{\mathcal{S}^{(i_1, \dots, i_k)}} \|\mathbf{y}(k) - \mathbf{v}_{i_1, \dots, i_k}\|^2 p(\mathbf{Y}^{(k)}) d\mathbf{Y}^{(k)} \\ &= \sum_{i_1, \dots, i_k=1}^{2^R} \int_{\mathcal{S}^{(i_1, \dots, i_{k-1})}} \int_{\mathcal{T}_{i_1, \dots, i_k}} \|\mathbf{y}(k) - \mathbf{v}_{i_1, \dots, i_k}\|^2 \\ & \quad \times p(\mathbf{Y}^{(k)}) d\mathbf{y}(k) d\mathbf{Y}^{(k-1)} \\ &= \sum_{i_1, \dots, i_k=1}^{2^R} \int_{\mathcal{T}_{i_1, \dots, i_k}} \|\mathbf{y}(k) - \mathbf{v}_{i_1, \dots, i_k}\|^2 \\ & \quad \times \int_{\mathcal{S}^{(i_1, \dots, i_{k-1})}} p(\mathbf{Y}^{(k)}) d\mathbf{Y}^{(k-1)} d\mathbf{y}(k) \end{aligned}$$

and since

$$p(\hat{\mathbf{Y}}^{(k-1)}, \mathbf{y}(k)) = \int_{\mathcal{S}^{(i_1, \dots, i_{k-1})}} p(\mathbf{Y}^{(k)}) d\mathbf{Y}^{(k-1)}$$

it follows that:

$$\begin{aligned} & \mathcal{E}\{\|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2\} \\ &= \sum_{i_1, \dots, i_k=1}^{2^R} \int_{\mathcal{T}_{i_1, \dots, i_k}} \|\mathbf{y}(k) - \mathbf{v}_{i_1, \dots, i_k}\|^2 \\ & \quad \times p(\hat{\mathbf{Y}}^{(k-1)}, \mathbf{y}(k)) d\mathbf{y}(k) \\ &= \sum_{\hat{\mathbf{Y}}^{(k-1)}} \int_{\mathbb{R}^M} \|\mathbf{y}(k) - \hat{\mathbf{y}}(k)\|^2 p(\hat{\mathbf{Y}}^{(k-1)}, \mathbf{y}(k)) d\mathbf{y}(k). \end{aligned}$$

Finally, (6) follows from:

$$p(\hat{\mathbf{Y}}^{(k-1)}, \mathbf{y}(k)) = p(\mathbf{y}(k) | \hat{\mathbf{Y}}^{(k-1)}) p(\hat{\mathbf{Y}}^{(k-1)}).$$

## APPENDIX B

### MONOTONIC DECREASE OF $D_K$ WITH $M$

For any  $M, R \in \mathbb{N}$ , define  $\mathbf{y}^{(M)}(k) = [y_1(k), \dots, y_M(k)]^T$ ,  $\hat{\mathbf{y}}^{(M)}(k) = [\hat{y}_1(k), \dots, \hat{y}_M(k)]^T$  and let  $\mathcal{Q}_k^{(M)} : \mathbf{y}^{(M)}(k) \mapsto \hat{\mathbf{y}}^{(M)}(k)$  denote the optimal quantizer for  $\mathbf{y}^{(M)}(k)$  of rate  $R$ . Let also  $D_k^{(M)}$  denote the distortion obtained by using the quantizer  $\mathcal{Q}_k^{(M)}$  on  $\mathbf{y}(k) = [y_1(k), \dots, y_M(k)]^T$ , and ignores the remaining coefficients of the expansion (5).

Let,  $M < N$ , and define the (not necessarily optimal) quantizer  $\mathcal{Q}_k^{(M, N)} : \mathbf{y}^{(N)}(k) \mapsto \hat{\mathbf{y}}^{(N)}(k)$  for  $\mathbf{y}^{(N)}(k)$ , obtained from the quantizer  $\mathcal{Q}_k^{(M)}$  as follows:

$$\left[ \mathcal{Q}_k^{(M, N)}(\mathbf{y}^{(N)}(k)) \right]_m = \begin{cases} \left[ \mathcal{Q}_k^{(M)}(\mathbf{y}^{(M)}(k)) \right]_m, & m \leq M \\ 0, & m > M \end{cases}$$

for all  $m = 1, \dots, N$ , where  $[x]_m$  denoted the  $m$ th component of  $x$ . That is,  $\mathcal{Q}_k^{(M, N)}$  quantizes the first  $M$  components of  $\mathbf{y}^{(N)}(k)$  according to the optimal quantizer  $\mathcal{Q}_k^{(M)}$  for  $\mathbf{y}^{(M)}(k)$ ,

and makes zero the remaining components. Since  $\mathcal{Q}_k^{(N)}$  is the optimal quantizer for  $\mathbf{y}^{(N)}(k)$  of rate  $R$ , it follows that

$$\begin{aligned} D_k^{(N)} &= \mathcal{E} \left\{ \left\| \mathbf{y}^{(N)}(k) - \mathcal{Q}_k^{(N)}(\mathbf{y}^{(N)}(k)) \right\|^2 \right\} + \sum_{m=N+1}^{\infty} \lambda_m^2 \\ &\leq \mathcal{E} \left\{ \left\| \mathbf{y}^{(N)}(k) - \mathcal{Q}_k^{(M, N)}(\mathbf{y}^{(N)}(k)) \right\|^2 \right\} + \sum_{m=N+1}^{\infty} \lambda_m^2 \\ &= \mathcal{E} \left\{ \left\| \mathbf{y}^{(M)}(k) - \mathcal{Q}_k^{(M)}(\mathbf{y}^{(M)}(k)) \right\|^2 \right\} \\ & \quad + \sum_{m=M+1}^{\infty} \lambda_m^2 = D_k^{(M)}. \end{aligned}$$

## APPENDIX C

### DESIGN OF A LPQ USING THE METHOD IN [25]

For simplicity, we describe the design of a scalar LPQ. The same procedure can be straightforwardly applied for designing a vector LPQ.

Let  $y(t)$ ,  $t \in \mathbb{N}$  be a scalar discrete-time signal,  $P = p_1 q^{-1} + p_2 q^{-2} + \dots + p_n q^{-n}$  be an  $n$ th-order predictor ( $q$  is the forward shift operation, i.e.,  $P(y)(t) = y(t+1)$ ), and  $\mathcal{Q}(\cdot)$  be a scalar quantizer. In the LPQ scheme, the reconstructed version  $\hat{y}(t)$  of  $y(t)$  is obtained as follows:

$$\hat{y}(t) = P(\hat{y})(t) + \mathcal{Q}(y - P(\hat{y}))(t).$$

Now, given the statistics (i.e., the autocorrelation) of the input signal, and the predictor  $P$  (which is straightforwardly designed using linear least-squares [7, Sec. 4.3], the problem is how to design  $\mathcal{Q}$ . To do so we use a realization  $y(t)$  as a training signal, and we use the following iterative procedure. At iteration  $i$ , let  $\hat{y}^{(i-1)}(t)$  denote the reconstructed version of  $y(t)$  obtained from the previous iteration. Then, the quantizer  $\mathcal{Q}^{(i)}$  is computed using the k-means algorithm on the residuals

$$\tilde{y}^{(i-1)}(t) = y(t) - P(\hat{y}^{(i-1)})(t).$$

After doing so, the reconstruction  $\hat{y}^{(i)}(t)$  is computed by

$$\hat{y}^{(i)}(t) = \mathcal{Q}^{(i)}(\tilde{y}^{(i-1)})(t).$$

The iterations stop when the reconstruction error stops decreasing. Also, the iterations are initialized by choosing

$$\hat{y}^{(1)}(t) = \mathcal{Q}^{(1)}(y - P(y))(t)$$

where the quantizer  $\mathcal{Q}^{(i)}$  is computed using the k-means algorithm on the residuals  $\tilde{y}^{(i-1)} = y - P(\hat{y}^{(i-1)})$ .

## APPENDIX D

### SPECTRAL REALIZATION METHOD USED FOR COMPUTING

#### (11)–(12)

In this Appendix we described the method we use to compute the state space model (11)–(12), from the auto-correlation (10).

Let  $K \in \mathbb{N}$  be such that the tail  $\mathbf{R}(k)$ ,  $k > K$ , is negligible, and let

$$\mathbf{R}_+(z) = \sum_{k=1}^K \mathbf{R}(k)z^{-k}. \quad (25)$$

Now, for a given  $N \in \mathbb{N}$ , we can find  $\mathbf{N}(z) = \sum_{n=1}^N \mathbf{N}_n z^{-n}$  and  $\mathbf{D}(z) = \sum_{n=1}^N \mathbf{D}_n z^{-n}$  such that

$$\mathbf{N}(z)(\mathbf{I} - \mathbf{D}(z))^{-1} \simeq \mathbf{R}_+(z).$$

This is done by solving the linear least-squares problem

$$[\mathbf{N}(z), \mathbf{D}(z)] = \arg \min_{\tilde{\mathbf{N}}(z), \tilde{\mathbf{D}}(z)} \|\mathbf{N}(z) - \tilde{\mathbf{N}}(z)(\mathbf{I} - \tilde{\mathbf{D}}(z))\|$$

where  $\|\cdot\|$  denotes the matrix Frobenius norm. For doing this approximation, the value of  $N$  is chosen using Akaike's criterion [26, p. 442].

The next step is to build a state-space model  $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$  [similar to the one in (11)–(12)] such that its transfer function  $\tilde{\mathbf{G}}(z) = \tilde{\mathbf{C}}(z\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}}$  approximates  $\mathbf{N}(z)(\mathbf{I} - \mathbf{D}(z))^{-1}$ . This is done by choosing [27, p. 101]

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \cdots & \mathbf{D}_N \\ \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}, \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix},$$

$$\tilde{\mathbf{C}} = [\mathbf{N}_1 \quad \cdots \quad \mathbf{N}_N].$$

The state-space model  $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$  is typically not minimal. Hence, we use balanced truncation ([27], p. 197) to obtain a model  $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$  of reduced order (i.e., with a matrix  $\tilde{\mathbf{A}}$  of smaller dimension than  $\tilde{\mathbf{A}}$ ), whose transfer matrix  $\tilde{\mathbf{G}}(z) = \tilde{\mathbf{C}}(z\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}}$  is approximately equal to  $\tilde{\mathbf{G}}(z)$ . Then, we have

$$\mathbf{R}_+(z) \simeq \tilde{\mathbf{G}}(z). \quad (26)$$

Now, from (25) and (26), it follows that the spectrum  $\mathbf{S}(z) = \sum_{k=-\infty}^{\infty} \mathbf{R}(k)z^{-k}$  of  $\mathbf{y}(k)$  is given by:

$$\begin{aligned} \mathbf{S}(z) &= \mathbf{R}_+^T(z^{-1}) + \mathbf{R}(0) + \mathbf{R}_+(z) \\ &\simeq \tilde{\mathbf{G}}^T(z^{-1}) + \mathbf{R}(0) + \tilde{\mathbf{G}}(z). \end{aligned}$$

Using the spectral factorization technique in [28, Sec. 8.5], it follows that:

$$\mathbf{S}(z) \simeq (\tilde{\mathbf{C}}(z\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}} + \check{\mathbf{D}})(\tilde{\mathbf{C}}(z^{-1}\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}} + \check{\mathbf{D}})^T \quad (27)$$

where  $\check{\mathbf{B}} = \mathbf{K}\mathbf{W}^{1/2}$  and  $\check{\mathbf{D}} = \mathbf{W}^{1/2}$ , with

$$\begin{aligned} \mathbf{K} &= \mathbf{V}(\mathbf{P})\mathbf{W}(\mathbf{P})^{-1} \\ \mathbf{V}(\mathbf{P}) &= \tilde{\mathbf{B}} - \tilde{\mathbf{A}}\mathbf{P}\tilde{\mathbf{C}}^T \\ \mathbf{W}(\mathbf{P}) &= \mathbf{R}(0) - \tilde{\mathbf{C}}\mathbf{P}\tilde{\mathbf{C}}^T \end{aligned}$$

and the matrix  $\mathbf{P}$  being the solution of the Riccati equation

$$\mathbf{P} = \mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{V}(\mathbf{P})\mathbf{W}(\mathbf{P})^{-1}\mathbf{V}(\mathbf{P})^T.$$

Equation (27) implies that a spectral realization of  $\mathbf{y}(k)$  is given by the following state-space model

$$\tilde{\mathbf{x}}(k+1) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}(k) + \check{\mathbf{B}}\mathbf{u}(k) \quad (28)$$

$$\mathbf{y}(k) = \check{\mathbf{C}}\tilde{\mathbf{x}}(k) + \check{\mathbf{D}}\mathbf{u}(k). \quad (29)$$

However, the model above includes an *output noise* component given by the matrix  $\check{\mathbf{D}}$ . To obtain the desired model (11)–(12) from (28)–(29), we use the transformation in [26, p. 178]. Doing so, we obtain

$$\mathbf{A} = \begin{bmatrix} \tilde{\mathbf{A}} & \check{\mathbf{B}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}, \mathbf{C} = [\check{\mathbf{C}} \quad \check{\mathbf{D}}].$$

## REFERENCES

- [1] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, Jan.–Feb. 2010.
- [2] K. Moslehi, "Intelligent infrastructure for coordinated control of a self-healing power grid," in *Proc. IEEE Electr. Power Syst. Conf. (EPEC'08)*, 2008.
- [3] G. Nair, F. Fagnani, S. Zampieri, and R. Evans, "Feedback control under data rate constraints: An overview," *Proc. IEEE*, vol. 95, no. 1, pp. 108–137, 2007.
- [4] M. Fu and C. de Souza, "State estimation for linear discrete-time systems using quantized measurements," *Automatica*, vol. 45, no. 12, pp. 2937–2945, 2009.
- [5] M. Unser, "Sampling-50 years after Shannon," *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, 2000.
- [6] R. Zamir and M. Feder, "Rate-distortion performance in coding band-limited sources by sampling and dithered quantization," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 141–154, 1995.
- [7] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, 1st ed. New York: Springer, 1991.
- [8] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
- [9] R. Balasubramanian, C. Bouman, and J. Allebach, "Sequential scalar quantization of vectors: An analysis," *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1282–1295, Sept. 1995.
- [10] N. Gaarder and D. Slepian, "On optimal finite-state digital transmission systems," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 167–186, 1982.
- [11] H. Witsenhausen, "On the structure of real-time source coders," *Bell Syst. Tech. J.*, vol. 58, no. 6, pp. 1437–1451, 1979.
- [12] J. Walrand and P. Varaiya, "Optimal causal coding-decoding problems," *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 814–820, 1983.
- [13] D. Teneketzis, "On the structure of optimal real-time encoders and decoders in noisy communication," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4017–4035, 2006.
- [14] V. Borkar, S. Mitter, and S. Tatikonda, "Optimal sequential vector quantization of Markov sources," *SIAM J. Contr. Optimiz.*, vol. 40, no. 1, pp. 135–148, 2001.
- [15] Y. Kaspi and N. Merhav, "Structure Theorems for Real-Time Variable-rate Coding With and Without Side Information" arXiv preprint arXiv:1108.2881, 2011.
- [16] D. Neuhoff and R. Gilbert, "Causal source codes," *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 701–713, 1982.
- [17] T. Linder and R. Zamir, "Causal coding of stationary sources and individual sequences with high resolution," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 662–680, 2006.
- [18] T. Linder and G. Lugosi, "A zero-delay sequential scheme for lossy coding of individual sequences," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2533–2538, 2001.
- [19] T. Weissman and N. Merhav, "On limited-delay lossy coding and filtering of individual sequences," *IEEE Trans. Inf. Theory*, vol. 48, no. 3, pp. 721–733, 2002.
- [20] A. Gyorgy, T. Linder, and G. Lugosi, "Efficient adaptive algorithms and minimax bounds for zero-delay lossy source coding," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2337–2347, 2004.
- [21] J. Conway, *A Course in Functional Analysis*. New York: Springer, 1990.

- [22] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control)*. Philadelphia, PA: SIAM, Jul. 2005.
- [23] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, ser. Springer Series in Statistics, 2nd ed. New York: Springer, Mar. 1999.
- [24] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [25] H. Khalil, K. Rose, and S. Regunathan, "The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 15–23, Jan. 2001.
- [26] T. Soderstrom and P. Stoica, *System Identification*, ser. Prentice-Hall International Series in Systems and Control Engineering. Englewood Cliffs, NJ: Prentice-Hall, Mar. 1994.
- [27] C.-T. Chen, *Linear System Theory and Design (Oxford Series in Electrical and Computer Engineering)*, 3rd ed. Oxford: Oxford Univ. Press, Sep. 1998.
- [28] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, Apr. 2000.



**Damián Marelli** received the Bachelor's degree in electronics engineering from the Universidad Nacional de Rosario, Argentina, in 1995, the Ph.D. degree in electrical engineering and a Bachelor (Honors) degree in mathematics from the University of Newcastle, Australia, in 2003.

From 2004 to 2005, he held a postdoctoral position at the Laboratoire d'Analyse Topologie et Probabilités, CNRS/Université de Provence, France. Since 2006, he is Research Academic at the School of Electrical Engineering and Computer Science,

University of Newcastle. In 2007, he received a Marie Curie Postdoctoral Fellowship, hosted at the Faculty of Mathematics, University of Vienna, Austria, and in 2010, he received a Lise Meitner Senior Fellowship, hosted at the Acoustics Research Institute of the Austrian Academy of Sciences. His main research interests include signal processing and communications.



**Kaushik Mahata** received the Ph.D. degree in electrical engineering from Uppsala University, Sweden, in 2003.

Since then he has been with the University of Newcastle, Australia. His research interest is in signal processing and its applications.



**Minyue Fu** (F'83) received the Bachelor's degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the University of Wisconsin-Madison, in 1983 and 1987, respectively.

From 1983 to 1987, he held a Teaching Assistantship and a Research Assistantship at the University of Wisconsin-Madison. He worked as a Computer Engineering Consultant at Nicolet Instruments, Inc., Madison, during 1987. From 1987 to 1989, he

served as an Assistant Professor in the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI. He joined the Department of Electrical and Computer Engineering, University of Newcastle, Australia, in 1989, where he is currently Chair Professor in Electrical Engineering and Head of School of Electrical Engineering and Computer Science. In addition, he was a Visiting Associate Professor at the University of Iowa during 1995–1996, and a Senior Fellow/Visiting Professor at Nanyang Technological University, Singapore, 2002. He holds a Qian-ren Professorship at Zhejiang University, China. His main research interests include control systems, signal processing, and communications.

Dr. Fu has been an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica*, and the *Journal of Optimization and Engineering*.