A Simple Family of Non-Linear Analog Codes

Mariano Eduardo Burich and Javier Garcia-Frias, *Senior Member, IEEE*

***Abstract*— We propose novel non-linear graph-based analog codes that directly encode k real-valued source samples into n real-valued samples by using (non-linear) sample-by-sample soft quantization of the input samples followed by a linear transformation on the soft-quantized values. Different from existing analog coding schemes, the proposed analog codes are able to produce additional output symbols in a rateless manner and can be decoded utilizing message passing algorithms dealing with real-valued nodes, achieving a performance close to the theoretical limits.**

***Index Terms*— Analog coding, non-linear codes, sparse codes, iterative decoding.**

I. Introduction

DIGITAL systems based on separation between source and channel coding have been the cornerstone of modern communications. However, digital systems are not very robust to changes in channel conditions and require long block lengths to approach the theoretical limits, which means substantial delays and high encoding/decoding complexity. These problems can be alleviated by the use of analog joint source-channel coding, where the concatenation of the (vector) quantizer, source encoder and channel encoder characteristic of digital systems is substituted by a simple end-to-end analog encoder. Indeed, the concatenation of quantizer, source encoder, channel encoder and modulation in a digital communication system can be seen as an analog code implementing a non-linear transformation, but this transformation is quite complex to be described as a single block. When designing analog codes, the objective is to define them as discrete-time, continuous-amplitude systems, which directly process k real-valued samples[1] and, using a non-linear transformation, produce n real-valued samples which after proper modulation are transmitted directly through the channel (i.e., bits are never used in the encoding process.)

The origin of analog coding goes back to Shannon in 1949, who in [1] discussed the idea of designing noise-resilient analog transformations, introducing the idea of using space-filling curves for analog coding. In 1976, [2] developed bounds for analog linear systems, proving that linear transformations have mediocre performance when used for bandwidth expansion ($n > k$) and very bad performance when bandwidth reduction is sought ($n < k$). This motivated the development of new non-linear mappings, including space-filing curves [3], [4] and other techniques, which have been successfully used in applications such as image transmission [5], [6]. While many of the analog codes proposed in the literature have low delay and good performance [6], [7], [8], they must be completely redesigned if (k, n) changes. Moreover, as k or n increases, searching for good mappings becomes intractable due to the computational complexity of the optimization algorithms.

II. System Overview

In a digital system, the concatenation of quantizer, source encoder, channel encoder and modulation can be simplified by substituting the last three blocks by a block performing coded modulation, such as Rate Compatible Modulation (RCM) [9] or an Analog Fountain Code (AFC) [10]. Then, we can see the end-to-end system as an analog code consisting of the concatenation of a quantizer and a coded modulation block. Since coded modulation schemes are flexible in terms of rate (rateless codes), this flexibility also holds for the whole system. However, the end-to-end performance would still be limited by the quantizer, which means that, contrary to what occurs in well-designed analog codes existing in the literature, performance will not improve when the channel quality gets better. The use of a quantizer leads to this lack of robustness, which is typical of digital systems.

Inspired by the scheme described in the previous paragraph, we propose novel non-linear graph-based analog codes which, different from existing schemes in the literature, i) are flexible in terms of rate, and additional redundant symbols can be added in a rateless manner, ii) do not need to be redesigned for different values of k and n, and iii) can be decoded utilizing message passing algorithms dealing with real-valued nodes.

Specifically, the proposed analog codes generate the transmitted symbols by using n linear combinations of shifted versions of the k real-valued source samples. Therefore, they consist of two serially concatenated stages. In the first stage, the quantizer is substituted by a non-linearity (shifting) that acts directly on the k real-valued source samples, producing a vector of k intermediate real numbers. The way this is done is to partition the input space into regions, similar to what a quantizer does. The difference is that rather than assigning a different centroid to each one of the regions, as in quantization, all the input source samples belonging to the same region are shifted by the same value, and different regions are shifted by different amounts so that after shifting the regions are

Manuscript received 17 August 2023; accepted 7 September 2023. Date of publication 18 September 2023; date of current version 9 November 2023. This material is based upon work supported by the National Science Foundation under Grant CCF-2007754 and was performed when Mariano Eduardo Burich was a graduate student at the University of Delaware. The associate editor coordinating the review of this letter and approving it for publication was L. Li. *(Corresponding author: Javier Garcia-Frias.)*

Mariano Eduardo Burich is with Micron Technology Inc., Irvine, CA 92612 USA (e-mail: burich@udel.edu).

Javier Garcia-Frias is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: jgf@udel.edu).

Digital Object Identifier 10.1109/LCOMM.2023.3316611

[1] The samples can proceed directly from a discrete-time continuous-amplitude source or being the result of sampling a continuous-time continuous-amplitude source.

separated from each other in what we can consider a "soft" quantization. In the second stage, the transmitted symbols are produced by generating n linear combinations of the k intermediate real numbers obtained in the first stage.

We consider normally distributed samples with mean zero and unit variance that are sequentially grouped into blocks of k samples, vector $\mathbf{m}^k$. The analog encoder transforms the k source samples into n output symbols, which are denoted by the vector $\mathbf{x}^n = [x_1, \ldots, x_n]$, $x_i \in \mathbb{R}$. The output symbols are normalized to have an average energy per symbol of E_s, and transmitted through an Additive White Gaussian Noise (AWGN) channel of mean zero and variance σ_n^2. Finally, the decoder processes the received noisy sequence and generates the estimate vector $\hat{\mathbf{m}}^k$.

The mean square error (MSE) is used as the distortion metric, expressed as signal-to-distortion rate $\text{SDR}_{\text{dB}} = 10\log_{10}\frac{1}{\text{MSE}}$. The signal-to-noise ratio, SNR_{dB}, is defined as $\text{SNR}_{\text{dB}} = 10\log_{10}\frac{E_s}{\sigma_n^2}$, where $E_s = \mathbb{E}[\|\mathbf{x}^n\|^2]/n$ is the average transmitted energy per symbol. The end-to-end system performance is assessed by the SDR vs SNR curve. Since $kR(D) < nC$, where $R(D)$ is the rate distortion function and C is the channel capacity, for a given $R_c = k/n$ we can easily obtain the theoretical limit (OPTA) in terms of SDR vs SNR.

III. Encoder

As explained before, the encoder consists of two sequential stages: Shifting and analog linear coding.

A. Stage 1: Shifting

The first stage in the proposed scheme is to perform a sample-by-sample shifting of the source samples. In order to do this, we partition the real line, $\mathbb{R}$, in Q segments or regions, similar to what a quantizer would do. Different from quantization, where a different centroid is assigned to each one of the regions, in the proposed scheme all the input source samples belonging to the same region are shifted by a fixed value, and different regions are shifted by different amounts so that after shifting the transformed regions are separated from each other in what we can consider a "soft" quantization. In this letter, where we focus on Gaussian sources, we will use the partitions that are optimal for the scalar quantization of a Gaussian random variable, as described in [11]. Since we are working in a joint source-channel coding framework, this does not guarantee optimality. However, it considerably simplifies the design.

Figure 1 depicts the distribution of the output of the aforementioned shifting transformation, $S'(m_i)$, for $Q = 4$. Notice that since for any value of Q the interval lengths are fixed (optimal quantization for Gaussian sources), the transformation S' is uniquely defined by providing the separation between the transformed regions. Since the input distribution (Gaussian) as well as the partition used as starting point are symmetric, optimal transformations S' will also be symmetric. When Q is odd, S' will be defined by a vector of $q = \frac{Q-1}{2}$ real numbers, $[l_0, l_1, \ldots, l_{q-1}]$ specifying the separation between the transformed regions in $\mathbb{R}+$. For an even Q, S' will be defined by a vector of $q = \frac{Q}{2}$ real numbers, with l_0 specifying the separation between the two transformed regions that are closer to 0 and $[l_1, \ldots, l_{q-1}]$ indicating the separation between the transformed regions in $\mathbb{R}+$. The final shifting transformation, S, is constructed by normalizing S', so that the variance of the prior for nodes $g_i = S(m_i)$ is the same as that of source nodes m_k (unit variance in this letter).

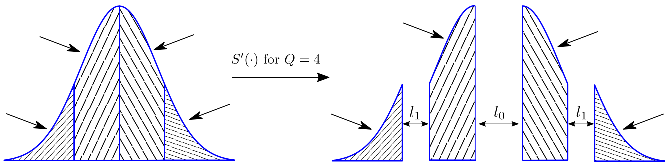


Fig. 1. Proposed non-linear shifting function, S', for the case of $Q = 4$.

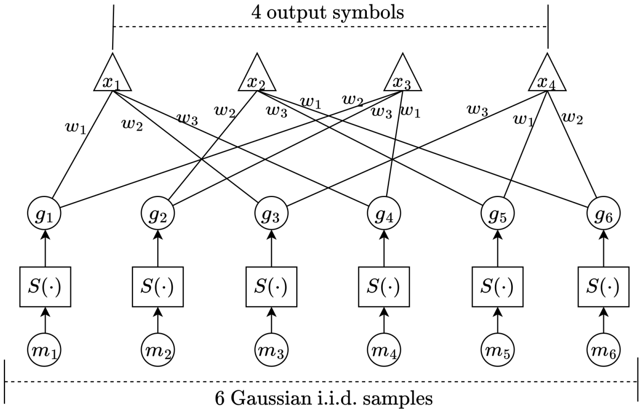


Fig. 2. Proposed scheme that encodes 6 Gaussian samples into 4 output symbols. Each symbol is generated by the linear combination of a subset of the source samples shifted by the function S.

As we will see in the sequel, the design of the function S is key in order to obtain good performance. Notice that the discontinuities introduced by S are an extension of what a quantizer does. Discontinuities also appear in a different manner in existing analog codes, such as in space-filling curves. As indicated before, linear analog codes have a very poor performance, and the discontinuities introduced in the shifting stage can be seen as very simple non-linearities that allow for improved performance over linear schemes.

B. Stage 2: Analog Linear Coding

Each transmitted symbol, $x_j \in \mathbb{R}$, is generated as a linear combination of a random subset of nodes, $g_i = S(m_i)$, proceeding from the first stage. Specifically,

$$x_j =< \mathbf{w}, \mathbf{g_j} >= \sum_{k=1}^{d_r} w_k g_{i|w(e_{ij})=w_k}, \qquad (1)$$

where d_r is the number of connections of each output symbol x_j, and its value is small so that the resulting graph is sparse and message passing can be applied; $w(e_{ij})$ is the weight of the edge that connects g_i and x_j, and $\mathbf{g}_j = [g_{i|w(e_{ij})=w_1}, \cdots, g_{i|w(e_{ij})=w_{d_r}}]$ is the set of nodes connected to the output symbol x_j, which are ordered according to the weight vector $\mathbf{w} = [w_1, \ldots, w_{d_r}]$.[2] Figure 2 depicts an example of the proposed non-linear analog code when $k = 6$ and $n = 4$ ($R_c = 1.5$).

[2] To facilitate the description of the decoding process, $\mathbf{w}$ is chosen so that the average transmitted energy per symbol is the desired value E_s.

IV. Decoder

As typically done for sparse graph-based digital channel codes, we use the belief propagation algorithm or message passing [12] to decode the received symbols, $\mathbf{y}^n$, resulting from the transmission of $\mathbf{x}^n$ through the AWGN channel ($y_j = x_j + n_j$, where $n_j \sim N(0, \sigma_n^2)$). The objective is to obtain the estimates of the source symbols, $\hat{m}_i$, that minimize the mean square error. This is achieved by estimating the posterior probability density function of the source samples, $p_{m_i}^p$, in an iterative manner and then choosing $\hat{m}_i$ as the mean of $p_{m_i}^p$. The iterative decoder for linear analog codes can be found in [13] and [14]. The proposed decoder, which is presented next, follows [13], [14] closely with two differences: First, the non-linearity introduced by the shifting function S. Second, [13], [14] use check nodes instead of the projections we consider here. In the sequel, the iteration number is denoted by t, where $t = 0$ in the first iteration.

A. Initialization

The only information at the nodes g_i during the first iteration comes from the *a priori* knowledge of the source, which is the probability density function $p_m^a(x) \sim \mathcal{N}(0,1)$. Thus, the *a priori* probability density function of g_i, $p_g^a(x) = S(p_m^a(x))$, is obtained by transforming p_m^a through the non-linear shifting function S, as follows

$$p_g^a(x) \propto \begin{cases} e^{\frac{-y^2}{2}}, & \text{for } x = S(y),\ y \in \mathbb{R} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

B. Symbol Nodes

The message passed from the transmitted symbol node, x_j, to the shifted node, g_i, is denoted as $s_{ij}^{(t)}(x)$, $x \in \mathbb{R}$. As previously mentioned, the equation of the received symbol, y_j, is expressed as

$$y_j = < \mathbf{w}, \mathbf{g}_j > + n_j = \sum_{k=1}^{d_r} w_k g_{v|w(e_{vj})=w_k} + n_j, \tag{3}$$

where $\mathbf{g}_j = [g_{v|w(e_{vj})=w_1}, \ldots, g_{v|w(e_{vj})=w_{d_r}}]$ is the vector of shifted nodes connected to x_j ordered according to their weight index.

The auxiliary variable, $z_{j\backslash i}$, is defined as

$$z_{j\backslash i} = \sum_{\substack{k=1 \\ v \neq i}}^{d_r} w_k g_{v|w(e_{vj})=w_k} + n_j, \tag{4}$$

and has probability density function

$$p_{z_{j\backslash i}} \propto \left(\underset{\substack{k \text{ s.t. } e_{kj} \in E_j \\ k \neq i}}{\circledast} q_{kj}^{\prime(t-1)} \right) \circledast \mathcal{N}(0, \sigma_n^2), \tag{5}$$

where $\circledast$ is the convolution operator, $q_{kj}^{\prime(t-1)}(x) \propto q_{kj}^{(t-1)}(x/w_k)$ is the properly scaled message (probability density function) passed from shifted node g_k to symbol node x_j as calculated in (10), E_j is the set of edges of the graph departing from symbol node x_j, and e_{kj} is the edge connecting g_k and x_j.

TABLE I

Partition of the Input Space as a Function of Q

Number of segments (Q)	Partition
2	$[-\infty, 0.0, \infty]$
3	$[-\infty, -0.6120, 0.6120, \infty]$
4	$[-\infty, -0.9816, 0.0, 0.9816, \infty]$
5	$[-\infty, -1.244, -0.3823, 0.3823, 1.244, \infty]$

Substituting (4) in (3) we get $y_j = z_{j\backslash i} + w(e_{ij})g_i$, and then we calculate

$$p(y_j|g_i) \propto \int_{\infty}^{\infty} p(y_j|g_i, z_{j\backslash i}) p(z_{j\backslash i}) dz_{j\backslash i} \tag{6}$$

$$\propto \int_{\infty}^{\infty} \delta(y_j - w(e_{ij})g_i - z_{j\backslash i}) p(z_{j\backslash i}) dz_{j\backslash i} \tag{7}$$

$$\propto p_{z_{j\backslash i}}(y_j - w(e_{ij})g_i), \tag{8}$$

where $\delta(\cdot)$ is the delta function and $p(z_{j\backslash i})$ denotes the probability density function, $p_{z_{j\backslash i}}(\cdot)$, calculated by (5).

Finally, the message from the symbol node, x_j, to the shifted node, g_i, is given by the probability density function

$$s_{ij}^{(t)}(x) \propto \left(\underset{\substack{k \text{ s.t. } e_{kj} \in E_j \\ k \neq i}}{\circledast} q_{kj}^{\prime(t-1)} \right) \circledast \mathcal{N}(0, \sigma_n^2)(y_j - w(e_{ij})x). \tag{9}$$

Compared with (5), $s_{ij}^{(t)}$ has been shifted leftwards by the value of the received observation, y_j, and scaled by $-w(e_{ij})$.

C. Shifted Nodes

The message passed from the shifted node g_i to the symbol node x_j at the t-th iteration is denoted as $q_{ij}^{(t)}(x)$. As in [13] and [14], the outgoing message $q_{ij}^{(t)}$ is equal to the point-wise product of the incoming messages, which includes the prior p_g^a, leading to the probability density function

$$q_{ij}^{(t)}(x) \propto p_g^a(x) \times \left(\prod_{v \text{ s.t. } e_{iv} \in E^i, v \neq j} s_{iv}^{(t-1)}(x) \right), \tag{10}$$

where E^i is the set of edges departing from node g_i, and e_{iv} is the edge between nodes g_i and x_v.

D. Decision

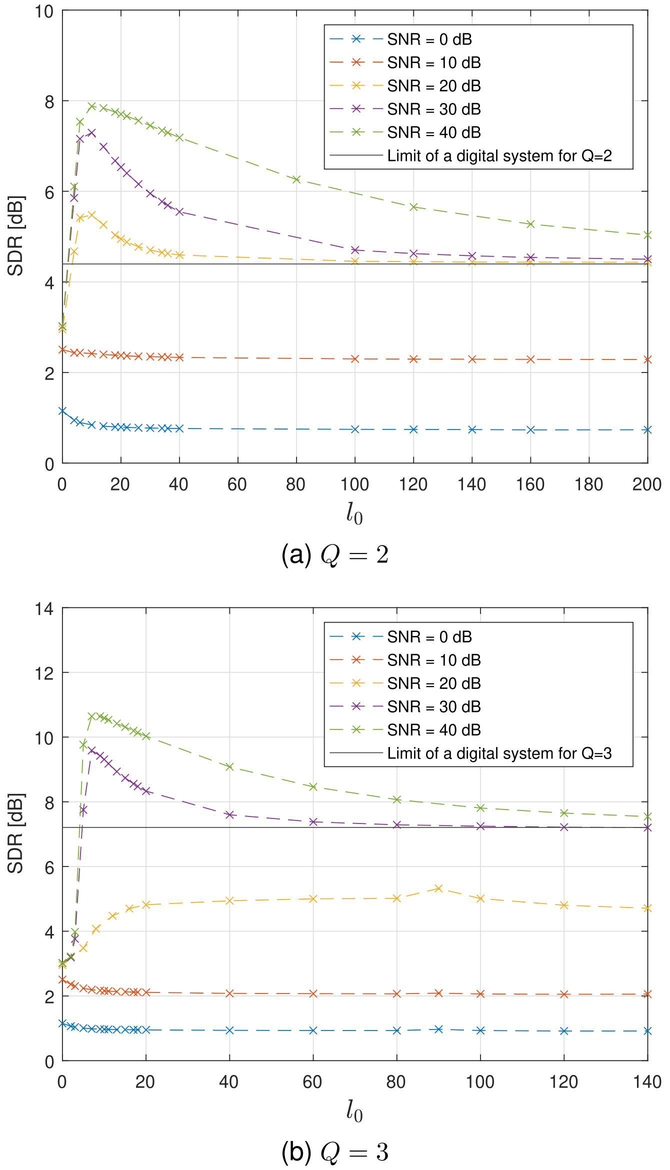
In order to make the final decision, we first estimate the posterior probability density function of shifted node g_i, $p_{g_i}^p$, by using all the incoming messages at node g_i. This results in

$$p_{g_i}^p(x) \propto p_g^a(x) \times \left(\prod_{v \text{ s.t. } e_{iv} \in E^i} s_{iv}^{(t-1)}(x) \right). \tag{11}$$

Then, we obtain the posterior probability density function of input node m_i by applying the inverse to transformation S, so that

$$p_{m_i}^p(x) \propto p_{g_i}^p(y) \text{ for } y = S(x), \quad x \in \mathbb{R}. \tag{12}$$

The source sample estimate, $\hat{m}_i$, is obtained by computing the expectation of this posterior probability.



(a) $Q = 2$

(b) $Q = 3$

Fig. 3. Simulation results for $d_r = 4$ and $R_c = 2$ as a function of l_0.

V. Simulation Results

We perform Monte Carlo simulations to evaluate the performance of the proposed scheme over the AWGN channel. As explained before, the partition of the input space is the same as for optimal scalar quantization and is presented in Table I for $Q = 2 \dots 5$. The coefficients for the linear mapping (stage 2) of the proposed scheme, $\mathbf{w}$, are the same as in [10], consisting of the d_r leftmost components of the vector $K \times [1/2, 1/3, 1/5, 1/7, 1/11, 1/13]$, where K is chosen so that the average transmitted energy per symbol is the desired value E_s. The vector $\mathbf{l} = [l_0, \dots, l_{q-1}]$ indicating the separation among the transformed regions is optimized by simulations. In order to calculate the messages (probability density functions) exchanged in the iterative decoder, we quantize them in the interval $[-50, 50]$ with 50001 values, which results in a resolution step of 0.002. The decoder performs 15 iterations per block.

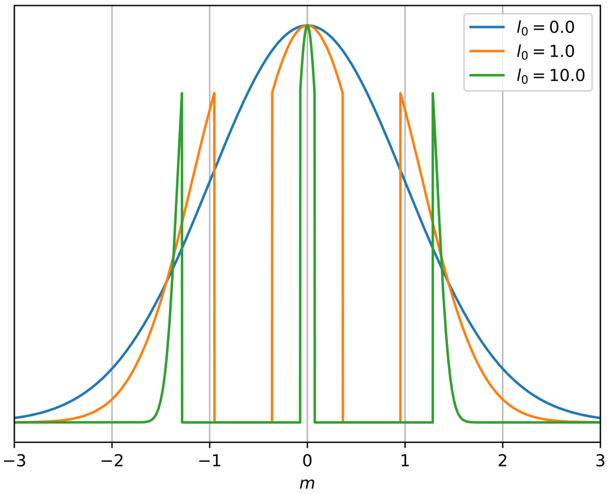


Fig. 4. Distribution of the prior for nodes g_i when $Q = 3$. As explained in Section III, normalization is performed so that the variance for nodes g_i is equal to that of the Gaussian source (unit variance in this letter). The increase of l_0 leads to a shrinkage of the lobes, which in the limit converge to deltas (scalar quantization).

We first evaluate the proposed scheme when $k = 300$ and $n = 150$ ($R_c = 2$) by simulating 80 blocks. Figures 3a and 3b depict the performance, in terms of SDR, of the proposed codes for $d_r = 4$ and several values of the SNR as a function of l_0 when $Q = 2$ and $Q = 3$, respectively. We also depict the limit of a system where optimal scalar quantization with Q levels is performed for each source sample, as in [11]. Notice that when $l_0 = 0$ the system is indeed a linear analog code. It is remarkable that in both subfigures the SDR begins to improve when l_0 increases, provided that the SNR is high enough.[3] When l_0 continues to increase, SDR achieves a maximum and then it decreases up to the theoretical limit for optimal scalar quantization. The reason is that, as depicted in Fig. 4, when l_0 increases the proposed non-linear analog code approximates a digital system performing optimal scalar quantization followed by coded modulation. It is remarkable that by choosing the appropriate value of l_0, the proposed non-linear analog code is able to easily outperform linear codes and it is not bounded by the SDR limit of scalar quantization. A similar behavior can be observed when both l_0 and l_1 increase and $Q = 4$ or $Q = 5$.

Figure 5 compares the performance of the proposed non-linear analog codes to the OPTA and to the theoretical limits when scalar quantization is used. For each value of SNR, the SDR was obtained by optimizing the vector $\mathbf{l} = [l_0, \dots, l_{q-1}]$. In the low SNR case, the best performance, already very close to the OPTA, is obtained by linear codes ($\mathbf{l} = \mathbf{0}$). As the SNR increases, the proposed non-linear analog codes rapidly begin to outperform the linear system, and they are also able to achieve SDRs higher than those of optimal digital systems based on scalar quantization. Although the performance of the proposed codes is several dB worse than that of the best analog codes in the literature [6], [7], [8] (e.g., for SNR = 30 dB, SDR = 10 dB vs SDR close to 15 dB in the aforementioned references), contrary to existing schemes in the literature the

[3] For low SNR, the performance of linear codes ($l_0 = 0$) is already close to the OPTA and using a non-linear code ($l_0 > 0$) will not lead to improved performance.

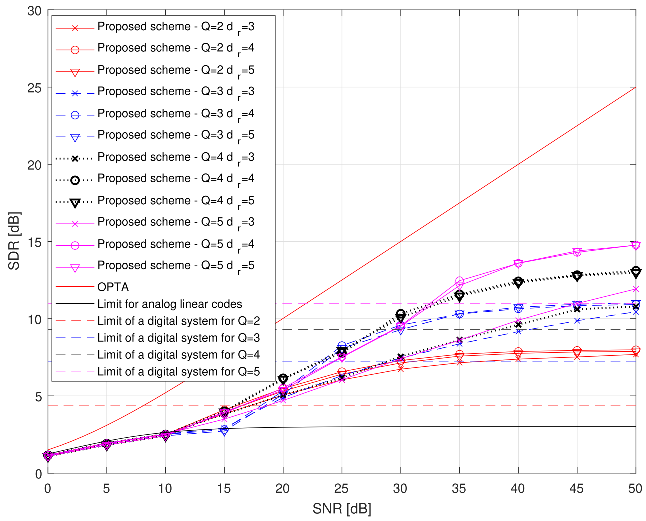


Fig. 5. Simulation results for the proposed non-linear analog codes with $R_c = 2$ for different values of Q and d_r. For comparison purposes, we also depict the performance of the optimal system when scalar quantization with Q levels is used [11], as well as the performance of an optimal linear code ($\mathbf{l} = \mathbf{0}$) [2]. Each data point for the non-linear codes corresponds to the best SDR obtained when the vector $\mathbf{l}$ is optimized.

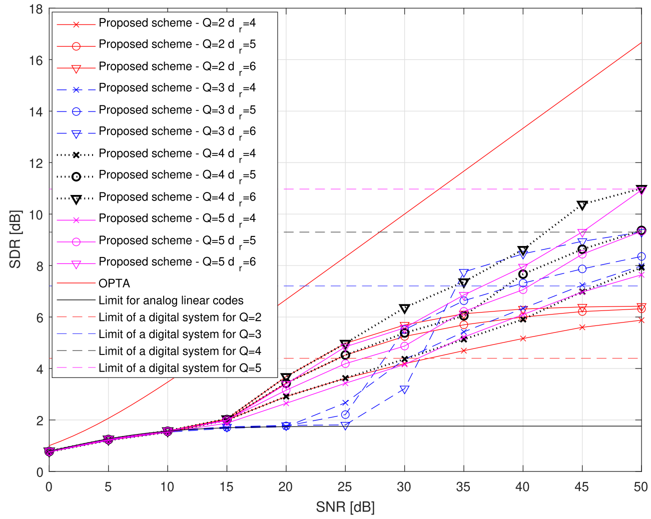


Fig. 6. Simulation results for the proposed non-linear analog codes with $R_c = 3$ for different values of Q and d_r. For comparison purposes, we also depict the performance of the optimal system when scalar quantization with Q levels is used [11], as well as the performance of an optimal linear code ($\mathbf{l} = \mathbf{0}$) [2]. Each data point for the non-linear codes corresponds to the best SDR obtained when the vector $\mathbf{l}$ is optimized.

proposed scheme does not have to be redesigned for different values of n and k and can be used in a rateless manner. Note that the envelope of the curves in Fig. 5 is parallel to the OPTA. This is an interesting result, as it is the first time it has been observed in the literature for graph-based analog codes decoded using message passing.

Next, we simulate 80 blocks for a $k = 300$, $n = 100$ analog system ($R_c = 3$). The system behavior when the vector $\mathbf{l}$ varies is similar to the case of $R_c = 2$ described before. Figure 6 compares the performance of the proposed non-linear analog codes to the OPTA and to the theoretical limits when scalar quantization is used. As before, the SDR was optimized for each value of SNR by optimizing the vector $\mathbf{l}$. Again, the performance is several dBs worse than that of the best analog codes in the literature.

VI. Conclusion

We have presented a novel analog coding scheme for the transmission of Gaussian samples through an AWGN channel by using a random-like sparse graph. The key idea is the insertion of non-linearities into each input sample prior to linear coding. These non-linearities are introduced by partitioning the input space into regions, similar to what a quantizer does. However, rather than assigning a different centroid to each one of the regions, all the input source samples belonging to the same region are shifted by the same value, and different regions are shifted by different amounts so that after shifting the regions are separated from each other in what we can consider as "soft" quantization. By optimizing the number of regions and the shifts applied to each region (e.g., more regions are required when the channel quality is better,) the proposed analog codes are able, for the first time in the literature, to achieve a performance close to the theoretical limits in a rateless fashion, easily outperforming analog linear codes and surpassing the theoretical limits that would be imposed by scalar quantization.

References

[1] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[2] K.-H. Lee and D. Petersen, "Optimal linear coding for vector channels," *IEEE Trans. Commun.*, vol. COM-24, no. 12, pp. 1283–1290, Dec. 1976.

[3] V. A. Vaishampayan, "Combined source-channel coding for bandlimited waveform channels," Ph.D. dissertation, Dept. Elect. Eng., Univ. Maryland, College Park, MD, USA, 1989.

[4] A. Fuldseth and T. A. Ramstad, "Bandwidth compression for continuous amplitude channels based on vector approximation to a continuous subset of the source signal space," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, Munich, Germany, Apr. 1997, pp. 3093–3096.

[5] Y. Hu, Z. Wang, J. Garcia-Frias, and G. R. Arce, "Non-linear coding for improved performance in compressive sensing," in *Proc. 43rd Annu. Conf. Inf. Sci. Syst.*, Baltimore, MD, USA, Mar. 2009, pp. 18–22.

[6] Y. M. Saidutta, A. Abdi, and F. Fekri, "Joint source-channel coding over additive noise analog channels using mixture of variational autoencoders," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2000–2013, Jul. 2021.

[7] Y. Hu, J. Garcia-Frias, and M. Lamarca, "Analog joint source-channel coding using non-linear curves and MMSE decoding," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 3016–3026, Nov. 2011.

[8] E. Akyol, K. B. Viswanatha, K. Rose, and T. A. Ramstad, "On zero-delay source-channel coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7473–7489, Dec. 2014.

[9] H. Cui, C. Luo, K. Tan, F. Wu, and C. W. Chen, "Seamless rate adaptation for wireless networking," in *Proc. 14th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*, Miami, FL, USA, Oct. 2011, pp. 437–446.

[10] M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Near-capacity adaptive analog fountain codes for wireless channels," *IEEE Commun. Lett.*, vol. 17, no. 12, pp. 2241–2244, Dec. 2013.

[11] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.

[12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Burlington, MA, USA: Morgan Kaufmann, 1988.

[13] F. Hu and W. Henkel, "An analysis of the sum-product decoding of analog compound codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 1886–1890.

[14] F. Hu, "Analog codes for analysis of iterative decoding and impulse noise correction," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Jacobs-Univ., Bremen, Germany, 2008.