Design of Optimal Fixed-rate Unrestricted Polar Quantizer for Bivariate Circularly Symmetric Sources

Huihui Wu, *Student Member, IEEE* and Sorina Dumitrescu, *Senior Member, IEEE*

***Abstract*—This letter presents an algorithm for the design of fixed-rate unrestricted polar quantizer (FUPQ) for bivariate circularly symmetric sources. The proposed algorithm is globally optimal for the class of FUPQs with the magnitude quantizer thresholds restricted to some predefined finite set. The solution algorithm is based on dynamic programming, which is further accelerated by exploiting a monotonicity property of the cost function. The time complexity of the accelerated algorithm is $O(KN^2)$, where N is the number of target qunatizer levels and K is the size of the predefined set of possible thresholds. The experimental results show that our approach outperforms the previous tractable designs when the total number of quantizer levels ranges between 25 and 256.**

***Index Terms*—Fixed-rate quantizer, unrestricted polar quantization, globally optimal algorithm, dynamic programming.**

I. INTRODUCTION

A polar quantizer quantizes the magnitude and the phase of a two dimensional source vector represented in polar coordinates. The phase quantizer is uniform, while the magnitude quantizer may be nonuniform. This paper considers the unrestricted polar quantizer (UPQ), where the phase quantizer depends on the magnitude level. More specifically, we address the design of fixed-rate UPQ (FUPQ).

A UPQ can be characterized by the configuration $(N, M, \mathbf{r}, \mathbf{P}, \boldsymbol{\theta})$ defined as follows. M is the number of magnitude levels and $\mathbf{r} = (r_0, r_1, \cdots, r_M)$ is the vector of thresholds of the magnitude quantizer, where $r_0 = 0 < r_1 < r_2 < \cdots < r_{M-1} < r_M = \infty$. Further, $\mathbf{P} = (P_1, P_2, \cdots, P_M)$, where P_m denotes the number of phase levels of the phase quantizer corresponding to the m-th magnitude bin (i.e., to $[r_{m-1}, r_m)$), for $1 \leq m \leq M$. $\mathbf{A} = (A_1, \cdots, A_M)$, where A_m is the m-th magnitude reconstruction value. Finally, $\boldsymbol{\theta} = (\theta_{m,s})_{1\leq m\leq M, 1\leq s\leq P_m}$, where $\theta_{m,s}$ is the phase reconstruction for the s-th cell of the phase quantizer corresponding to the m-th magnitude bin. Note that the total number of quantization bins of the UPQ is $N = \sum_{m=1}^{M} P_m$.

The goal in optimal FUPQ design is to minimize the distortion for a fixed number N of quantization bins. Design methods based on the asymptotic quantization theory are addressed in [1]–[3]. We point out that such techniques guarantee the optimality of the design only as the rate approaches infinity.

The authors are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada (Emails: wuh58@mcmaster.ca;sorina@mail.ece.mcmaster.ca).

The design of optimal FUPQ at finite rates is investigated in [4] and [5]. The approach taken in [4] is to solve iteratively the necessary conditions for optimal $\mathbf{r}$ and $\mathbf{A}$ when M and $\mathbf{P}$ are fixed. However, Wilson [4] relies on exhaustive search to optimize the rate allocation between the magnitude and phase quantizers, i.e., to find the optimum configuration $(M, \mathbf{P})$ satisfying the constraint $N = \sum_{m=1}^{M} P_m$. The algorithm of [4] is globally optimal, but it becomes intractable as N increases since the total number of possible configurations $(M, \mathbf{P})$ increases exponentially with N.

The authors of [5] propose a nearly optimal algorithm which iteratively optimizes the values of the vector $\mathbf{r}$, respectively, $\mathbf{P}$ and $\mathbf{A}$, while the other two vectors are kept fixed. The drawbacks of the method in [5] are slow convergence and lack of guarantee of optimality.

In this work we propose an efficient globally optimal FUPQ design algorithm, for the class of FUPQs with magnitude quantizer thresholds restricted to a predefined finite set. The solution algorithm is based on dynamic programming sped up with the aid of a fast matrix search technique in totally monotone matrices [6]. Its time complexity is $O(KN^2)$, where K is the size of the set from which the magnitude thresholds are selected.

The main contribution of this work over prior work on FUPQ design, resides in proposing the first algorithm which handles efficiently the problem of rate allocation between the magnitude and phase quantizers, while still guaranteeing the globally optimal solution (under certain constraints) at finite rates.

Very recently, an efficient algorithm for the design of optimal entropy-constrained (EC) UPQ was proposed in [7]. It is worth pointing out that the present work has significant differences versus [7]. Specifically, the current work minimizes the distortion with a constraint on the number of levels, while the problem solved in [7] is formulated as the unconstrained minimization of a weighted sum of distortion and entropy. These different formulations call for different solution approaches, with distinct time complexities. Additionally, in the current work we solve the problem for any possible number N of quantizer levels, while the algorithm in [7] can find only the ECUPQs corresponding to points on the lower boundary of the convex hull of the set of entropy-distortion pairs.

This letter is structured as follows. The next section contains the problem formulation. Section III presents a dynamic programming solution algorithm achieving $O(K^2N^2)$ time complexity. Section IV establishes a monotonicity property of the objective function, which is further exploited to reduce the time complexity to $O(KN^2)$. Section V presents experimental results and the comparison with prior FUPQ designs. Finally, Section VI concludes the letter.

II. Problem Formulation

Consider a bivariate random variable with the following circularly symmetric density, as a function of the polar coordinates r and θ,

$$p(r,\theta) = \frac{1}{2\pi} g(r),\ 0 \le r < \infty,\ 0 \le \theta < 2\pi.$$

Note that $g(r)$ is the marginal probability density function (pdf) of the magnitude variable, while the phase variable is uniformly distributed over the interval $[0, 2\pi)$. Each quantization bin of the UPQ can be represented as

$$\mathcal{R}(m,s) = \left\{ re^{j\theta} | r_{m-1} \le r < r_m, (s-1)\frac{2\pi}{P_m} \le \theta < s\frac{2\pi}{P_m} \right\},$$

with $A_m e^{j\theta_{m,s}}$ being the reconstructed magnitude-phase pair, for $1 \le m \le M$, $1 \le s \le P_m$. According to [4] the reconstruction values which minimize the squared error distortion are given by $\theta_{m,s} = (2s-1)\pi/P_m$ and

$$A_m = sinc\left(\frac{1}{P_m}\right) \frac{\int_{r_{m-1}}^{r_m} rg(r)dr}{\int_{r_{m-1}}^{r_m} g(r)dr},$$

where $sinc(\frac{1}{P_m}) = \frac{\sin(\pi/P_m)}{\pi/P_m}$. The expected distortion (per sample) of the UPQ is [4]

$$D = \frac{1}{2}\left(\int_0^\infty r^2 g(r)dr - \sum_{m=1}^{M} A_m^2 \int_{r_{m-1}}^{r_m} g(r)dr \right). \quad (1)$$

We will assume that the thresholds of the magnitude quantizer of the UPQ take values in some set $\mathcal{A} = \{a_0, a_1, \cdots, a_{K-1}\}$, where $a_0 = 0$, $a_{K-1} = \infty$ and $a_i < a_{i+1}$, for $0 \le i \le K-2$. This set can be obtained by finely discretizing some interval $[0, B]$, chosen such that the probability that the magnitude level is larger than B, to be sufficiently small.

For each positive integer k and extended real number $\beta \in \mathcal{A} \cap (\mathbb{R} \cup \{\infty\})$, denote by $\mathcal{T}_k(\beta)$ the set of all $(k+1)$-tuples $\mathbf{r} = (r_0, r_1, \cdots, r_k)$ such that $0 = r_0 < r_1 < r_2 < \cdots < r_k = \beta$ and $r_m \in \mathcal{A}$ for all $1 \le m \le k-1$.

The problem of fixed-rate UPQ design can be formulated as the following level-constrained minimization problem

$$\begin{aligned} &\min_{M,\mathbf{r},\mathbf{P}} \quad D \\ &s.t. \ \sum_{m=1}^{M} P_m = N,\ P_m \in \mathbb{Z}^+,\ \mathbf{r} \in \mathcal{T}_M(\infty), \end{aligned} \quad (2)$$

where $\mathbb{Z}^+$ is the set of positive integers and N is the target value for the number of levels of the UPQ. In this work we propose a globally optimal solution to the above problem.

III. Dynamic Programming Solution

In this section we present a solution to problem (2) based on dynamic programming. First we will introduce a few more notations. For each $y \ge 1$, let $f(y) \triangleq -sinc^2(\frac{1}{y})$ and for $\alpha \le \beta$ and positive integer P denote

$$q(\alpha,\beta) \triangleq \int_\alpha^\beta g(r)dr, \quad x(\alpha,\beta) \triangleq \frac{\int_\alpha^\beta rg(r)dr}{\int_\alpha^\beta g(r)dr},$$

$$\omega_P(\alpha,\beta) \triangleq \frac{1}{2} f(P) \left(x(\alpha,\beta)\right)^2 q(\alpha,\beta). \quad (3)$$

Notice that the first term in (1) is constant, therefore it can be removed from the cost function of (2). After doing so the objective function of (2) becomes

$$\mathcal{O}(\mathbf{r},\mathbf{P}) \triangleq \sum_{m=1}^{M} \omega_{P_m}(r_{m-1}, r_m).$$

For each pair of positive integers (k,n) with $1 \le k \le K-1$ and $1 \le n \le N$, consider problem $\mathcal{P}(k,n)$ defined as

$$\begin{aligned} &\min_{M,\mathbf{r},\mathbf{P}} \quad \mathcal{O}(\mathbf{r},\mathbf{P}) \\ &s.t. \ \sum_{m=1}^{M} P_m = n,\ P_m \in \mathbb{Z}^+,\ \mathbf{r} \in \mathcal{T}_M(a_k). \end{aligned} \quad (4)$$

Additionally, denote by $\hat{\mathcal{O}}(k,n)$ the optimal value of the objective function in (4), for $1 \le k \le K-1$ and $1 \le n \le N$.

Intuitively, problem (4) can be interpreted as finding the optimal FUPQ with n levels, corresponding to the portion of the magnitude space ranging from 0 to a_k. It can be easily seen that problem (2) is equivalent to $\mathcal{P}(K-1, N)$. The dynamic programming solution consists of solving all sub-problems $\mathcal{P}(k,n)$, for $1 \le k \le K-1$ and $1 \le n \le N$, using the following recurrence relation

$$\hat{\mathcal{O}}(k,n) = \min_{0 \le t < n} \min_{0 \le j < k} \left(\hat{\mathcal{O}}(j,t) + \omega_{n-t}(a_j, a_k) \right), \quad (5)$$

where $\hat{\mathcal{O}}(0,0) = 0$ and $\hat{\mathcal{O}}(0,t) = \hat{\mathcal{O}}(j,0) = \infty$, for $t > 0$ and $j \ge 1$. The dynamic programming process evaluates (5) in increasing order of k and n. For each pair (k,n) the minimizations in (5) take $O(KN)$ operations if each quantity $\omega_{n-t}(a_j, a_k)$ can be evaluated in constant time. Since there are $O(KN)$ pairs (k,n) in total, the time complexity of the solution algorithm becomes $O(K^2N^2)$. It can be seen from (3) that for computing the values $\omega_{n-t}(a_j, a_k)$ the quantities $x(a_j, a_k)$ and $q(a_j, a_k)$ are needed. In order to enable the computation of each $x(a_j, a_k)$ and $q(a_j, a_k)$ in constant time, the cumulative probabilities and first moments are precomputed and stored in a preprocessing step as in [7], which only requires $O(K)$ operations.

In the next section we will show that the algorithm can be sped up by exploiting a certain monotonicity property of the objective function.

IV. Complexity Reduction

For each pair of integers (n,t) with $1 \leq t < n \leq N$, consider the upper triangular matrix $G_{n,t}$ with elements $G_{n,t}(j,k)$, $1 \leq j < k \leq K-1$,

$$G_{n,t}(j,k) \triangleq \hat{\mathcal{O}}(j,t) + \omega_{n-t}(a_j, a_k). \tag{6}$$

Clearly, the minimization over j in (5) is equivalent to finding the smallest element on column k of matrix $G_{n,t}$, i.e., finding

$$\hat{G}_{n,t}(k) \triangleq \min_{1 \leq j \leq k-1} G_{n,t}(j,k). \tag{7}$$

Then relation (5) is equivalent to

$$\hat{\mathcal{O}}(k,n) = \min\left(\omega_n(0,a_k), \min_{1\leq t<n} \hat{G}_{n,t}(k)\right). \tag{8}$$

Determining all column minima takes $O(K^2)$ time in a general $O(K)$-by-$O(K)$ matrix. However, when the matrix is *totally monotone* this task can be accomplished in $O(K)$ time using the algorithm nicknamed SMAWK [6]. According to [6] matrix $G_{n,t}$ is said to be totally monotone (with respect to the column minima problem[1]) if for all $j < j'$ and $k < k'$ the following implication holds

$$G_{n,t}(j',k) < G_{n,t}(j,k) \Rightarrow G_{n,t}(j',k') < G_{n,t}(j,k').$$

A sufficient condition for the total monotonicity to hold is the following, known as the *Monge* condition [8]

$$G_{n,t}(j,k) + G_{n,t}(j',k') \leq G_{n,t}(j,k') + G_{n,t}(j',k) \tag{9}$$

for all $1 \leq j < j' < k < k' \leq K-1$.

Proposition 1: Matrix $G_{n,t}$ satisfies the Monge condition.

Proof: By replacing (6) in (9) and performing the cancellation of the like terms, (9) becomes equivalent to

$$\omega_{n-t}(a_j,a_k)+\omega_{n-t}(a_{j'},a_{k'}) \leq \omega_{n-t}(a_j,a_{k'})+\omega_{n-t}(a_{j'},a_k). \tag{10}$$

Define now, for $1 \leq j < k \leq K-1$,

$$d(j,k) \triangleq \int_{a_j}^{a_k} r^2 g(r)dr - (x(a_j,a_k))^2 q(a_j,a_k).$$

It was shown in [9] that $d(j,k)$ satisfies the Monge condition, i.e., the following holds

$$d(j,k) + d(j',k') \leq d(j,k') + d(j',k), \tag{11}$$

for all $1 \leq j < j' < k < k' \leq K-1$. Note from (3) that

$$d(j,k) = \int_{a_j}^{a_k} r^2 g(r)dr + \frac{2}{sinc^2(\frac{1}{n-t})}\omega_{n-t}(a_j,a_k).$$

By applying the above in (11) and performing some algebraic manipulations, relation (10) follows. ■

The fast solution algorithm proceeds as follows. It iterates over n in increasing order from 1 to N. For each n, problem $\mathcal{P}(k,n)$ is solved for all k, as follows. We increase t from 1 to $n-1$ and for each t all column minima in matrix $G_{n,t}$ are determined using SMAWK. This requires $O(K)$ time for each matrix. Over all values of t, this amounts to $O(KN)$ operations. After that the minimization over t in (5) is performed, for each k, requiring a total of $O(KN)$ operations. Performing the above for all n leads to $O(KN^2)$ time complexity for the solution algorithm.

Note that in order to apply SMAWK, the matrix $G_{n,t}$ has to be extended to a full matrix. This can be done by setting to ∞ all elements below the main diagonal. This extension does not change the column minima, and the full matrix still satisfies the total monotonicity [8].

The following pseudocode (Algorithm 1) describes the algorithm to solve problem (2). We use the notation $\hat{j}_{n,t}(k)$ for the value of j achieving optimality in (7), and $\hat{t}(n,k)$ for the optimal t in (5).

Algorithm 1: Solution algorithm to problem (2).

Preprocessing Stage
begin
  for $k = 1$ **to** $K-1$ **do**
    $\hat{\mathcal{O}}(k,1) = \omega_1(0,a_k)$ /* $n = 1$ */
    $\hat{j}_{1,0}(k) = 0$
    $\hat{t}(1,k) = 0$
  for $n = 1$ **to** N **do**
    $\hat{\mathcal{O}}(1,n) = \omega_n(0,a_1)$ /* $k = 1$ */
    $\hat{j}_{n,0}(1) = 0$
    $\hat{t}(n,1) = 0$
  for $n = 2$ **to** N **do**
    for $t = 1$ **to** $n-1$ **do**
      Evaluate $\hat{G}_{n,t}(k)$ for all k using SMAWK
      Record $\hat{j}_{n,t}(k)$ for all k
    for $k = 2$ **to** K **do**
      Compute $\hat{\mathcal{O}}(k,n)$ using (8)
      Record $\hat{t}(n,k)$
  Restore the vectors r and P

V. Experimental Results

This section assesses the practical performance of the proposed FUPQ design algorithm and compares it with the designs of [3], [4] and [5]. The experiments are conducted for a two-dimensional random vector (X_1, X_2), where X_1 and X_2 are i.i.d. Gaussian variables with zero-mean and unit-variance, with the following joint pdf in polar coordinates

$$p(r,\theta) = \frac{r}{2\pi}\exp\left(-\frac{r^2}{2}\right),\ 0 \leq r < \infty,\ 0 \leq \theta < 2\pi,$$

where $r = \sqrt{x_1^2 + x_2^2}$, and $\theta = \tan^{-1}(x_2/x_1)$. It then follows that $g(r) = r\exp(-r^2/2)$.

The set $\mathcal{A}$ consists of elements $a_i = 0.002i$, for $0 \leq i \leq K-2$, and $a_{K-1} = \infty$, where $K = 3001$. We applied the dynamic programming algorithm to construct the optimal FUPQ with $N = 256$. The FUPQs for all $N < 256$ were also generated during the dynamic programming process.

[1]The total monotonicity is defined in [6] for the problem of row maxima, which can be converted to the column minima problem by transposing the matrix and multiplying all entries by -1. Here we adapt the definition of total monotonicity to the column minima problem.

TABLE I
PERFORMANCE COMPARISON WITH THE FUPQ OF [4] AND THE CORRESPONDING OPTIMAL CONFIGURATION, FOR $N = 25$ AND 36.

N	$(M, P_1, \cdots, P_M)$	$(r_1, \cdots, r_{M-1})$	$10\log_{10} D$	$(M, P_1, \cdots, P_M)^{[4]}$	$(r_1, \cdots, r_{M-1})^{[4]}$	$10\log_{10} D^{[4]}$	$10\log_{10}\frac{D^{[4]}}{D}$
25	$(3, 5, 9, 11)$	$(0.856, 1.682)$	-11.3188	$(3, 4, 10, 11)$	$(0.798, 1.674)$	-11.3181	0.0007
36	$(4, 3, 8, 12, 13)$	$(0.524, 1.130, 1.898)$	-12.7805	$(4, 1, 8, 13, 14)$	$(0.369, 1.051, 1.848)$	-12.7772	0.0033

TABLE II
PERFORMANCE COMPARISON WITH THE FUPQ OF [5] AND THE CORRESPONDING OPTIMAL CONFIGURATION.

N	$(M, P_1, \cdots, P_M)$	$(r_1, \cdots, r_{M-1})$	$10\log_{10} D$	$(M, P_1, \cdots, P_M)^{[5]}$	$(r_1, \cdots, r_{M-1})^{[5]}$	$10\log_{10} D^{[5]}$	$10\log_{10}\frac{D^{[5]}}{D}$
64	$(5, 5, 10, 15, 18, 16)$	$(0.536, 0.998, 1.534, 2.234)$	-15.150	$(6, 2, 6, 11, 15, 16, 14)$	$(0.277, 0.663, 1.120, 1.655, 2.345)$	-15.082	0.068
128	$(8, 1, 7, 13, 18, 22, 24, 24, 19)$	$(0.180, 0.498, 0.826, 1.174, 1.564, 2.026, 2.650)$	-18.053	$(8, 4, 9, 14, 18, 22, 23, 22, 16)$	$(0.324, 0.623, 0.943, 1.290, 1.688, 2.161, 2.806)$	-17.991	0.062
256	$(11, 1, 8, 14, 20, 25, 29, 33, 35, 35, 32, 24)$	$(0.138, 0.378, 0.610, 0.848, 1.098, 1.364, 1.660, 1.998, 2.408, 2.972)$	-20.985	$-$	$-$	-20.907	0.078

TABLE III
PERFORMANCE COMPARISON OF THE PROPOSED FUPQ WITH ASY AND PASY OF [3], FOR $N \geq 16$.

N	$10\log_{10} D$	$10\log_{10} D_{ASY}$	$10\log_{10} D_{PASY}$	$10\log_{10}\frac{D_{PASY}}{D}$
16	-9.614	-9.572	-9.324	0.290
32	-12.340	-12.297	-12.206	0.134
64	-15.150	-15.125	-15.075	0.075
128	-18.053	-18.022	-17.969	0.084
256	-20.985	-20.963	-20.945	0.040

Wilson [4] constructed the optimal FUPQs for all N between 1 and 16, and for $25, 32$ and 36, and reported the optimal configuration $M, \mathbf{P}, \mathbf{r}$. Our approach generated the same FUPQs as in [4] for all N, except for $N = 25$ and 36. The results for the latter values and the comparison with [4], are presented in Table I. We see that our design exhibits an improvement in distortion of 0.0033 dB for $N = 36$, respectively 0.0007 dB for $N = 25$. It is worth pointing out that, while the performance of the FUPQ of [4] is identical or very close to our scheme for small values of N, the algorithm of [4] is not tractable for larger values of N, because of the exponential growth of the space of all configurations $(M, \mathbf{P})$ satisfying $N = \sum_{m=1}^{M} P_m$. On the other hand, the time complexity of our proposed solution grows only quadratically with N, therefore it is tractable for much larger values.

Table II illustrates the comparison with the FUPQ of [5]. Note that the authors of [5] only report the distortions for $N = 64, 128, 256$, and the optimal FUPQ parameters $M, \mathbf{P}, \mathbf{r}$ for $N = 64, 128$. It can be seen that our algorithm always outperforms the design of [5] with gains higher than 0.06 dB, and reaching a peak improvement of 0.078 dB when $N = 256$.

Next we compare the performance of the proposed design with the FUPQ of [3], using the results reported in [3]. We use the acronym ASY to refer to the asymptotical performance derived in [3], and the acronym PASY to refer to the practical design counterpart. Table III illustrates the performance of the proposed algorithm in comparison with ASY and PASY, for N taking as values the powers of 2 from 16 to 256. We see that the proposed algorithm is superior to both ASY and PASY for all values of N examined. Specifically, the gains over ASY are always higher than 0.01 dB, with a peak of 0.043 dB at $N = 32$. The performance improvement over PASY ranges between 0.29 dB and 0.075 dB for N between 16 and 128. Additionally, we observe that the gap between PASY and the proposed scheme tends to decrease as N increases. This is expected since PASY is globally optimal as $N \to \infty$, therefore its accuracy is expected to improve as N increases. On the other hand, since the proposed approach is globally optimal at finite rates (subject to the confined set of thresholds), it can serve as a benchmark to establish the accuracy of PASY and ASY at finite rates.

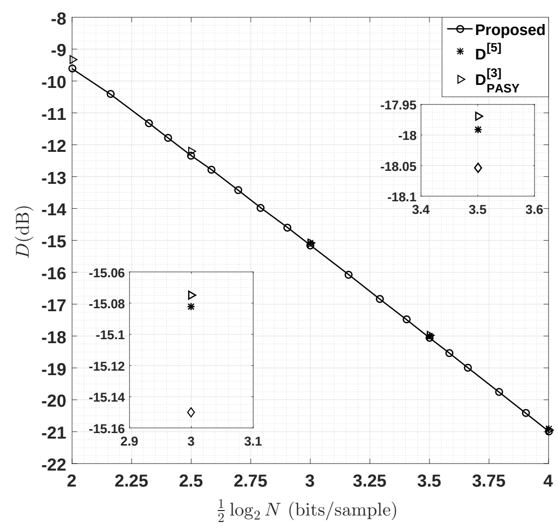

Fig. 1. Performance comparison with PASY [3] and with [5].

Finally, Figure 1 plots the distortion in dB (i.e., $10\log_{10} D$) versus rate, computed as $R = \frac{1}{2}\log_2 N$, for the proposed FUPQ, the PASY scheme in [3] and the design of [5], where the plots of (R, D) pairs at $R = 3$ and 3.5 are magnified.

VI. CONCLUSION

This letter presents a globally optimal algorithm for fixed-rate unrestricted polar quantizer design for bivariate circularly symmetric sources. The global optimality holds when the thresholds of the magnitude quantizer are selected from a predefined finite set. The proposed solution is a dynamic programming algorithm sped up based on a monotonicity property of the objective function. The experimental results show better performance than predicted by the high-rate quantization theory and than the prior tractable designs when the number of quantizer cells ranges between 25 and 256.

REFERENCES

[1] P. F. Swaszek and T.W. Ku, "Asymptotic performance of unrestricted polar quantizers," *IEEE Trans. Inform. Theory,* vol. IT-32, no. 2, pp. 330-333, Mar. 1986.

[2] D. L. Neuhoff, "Polar quantization revisited," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT 1997),* pp. 60, Ulm, Germany, Jun. 1997.

[3] Z. Peric and J. Nikolic, "Design of asymptotically optimal unrestricted polar quantizer for Gaussian source", *IEEE Signal Process. Lett.,* vol. 20, no. 10, pp. 980-983, Oct. 2013.

[4] S. G. Wilson, "Magnitude/phase quantization of independent Gaussian variates", *IEEE Trans. Commun.,* vol. COM-28, no. 11, pp. 1924-1929, Nov. 1980.

[5] M. D. Petkovic, Z. H. Peric, and A. Z. Jovanovic, "An iterative method for optimal resolution-constrained polar quantizer design," *COMPEL: Int. J. Comput. and Math. Elect. and Electron. Eng.,* vol. 30, no. 2, pp. 574-589, 2011.

[6] A. Aggarval, M. Klave, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix-searching algorithm", *Algorithmica*, vol. 2, no. 1-4, pp.195-208, Nov. 1987.

[7] H. Wu and S. Dumitrescu, "Design of Optimal Entropy-constrained Unrestricted Polar Quantizer for Bivariate Circularly Symmetric Sources", accepted for publication in *IEEE Trans. Commun.,* available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8245805.

[8] R. E. Burkard, B. Klinz, and R. Rudolf, "Perspectives of Monge properties in optimization", *Discrete Applied Mathematics,* vol. 70, no. 2, pp. 95-161, Sept. 1996,

[9] X. Wu, "Optimal Quantization by Matrix Searching", *Journal of Algorithms,* vol. 12, no. 4, pp. 663-673, Dec. 1991.