TURBO QUANTIZATION OF FRAMES

Mohamed F. Mansour

Texas Instruments Inc.
Dallas, Texas, USA

ABSTRACT

We describe an iterative algorithm for quantizing overcomplete frames that achieves more than 18 dB/Octave improvement in the reconstruction error due to quantization. It starts with the trellis quantization path then successively refines the estimation with monotonically improving quantization through turbo iterations. We call the procedure *turbo quantization* because it resembles turbo coding in the error control coding literature.

***Index Terms*—** quantization, frames, oversampling, optimization, integer programming.

1. INTRODUCTION

Overcomplete frames provide a flexible representation for signals in $L^2(\mathbb{R})$ (the space of finite-energy functions or signals) with a stable reconstruction operator [1, 2]. They are encountered in numerous signal processing applications, e.g., discrete wavelet transform, filter banks, Gabor transform, and oversampled data converters. Frames exploit the duality between $L^2(\mathbb{R})$ and $l^2(\mathbb{Z})$ (that is the space of square summable sequences) to provide one-to-many discrete equivalents of signals in $L^2(\mathbb{R})$. This mapping provides a rigorous generalization to orthonormal basis of $L^2(\mathbb{R})$. The redundancy in the mapping, that is enabled by oversampling in the overcomplete frame representation, enables the deployment of optimization techniques to acheive desired criteria in the frame representation. The quantization of the coefficients of the frame representation has many important applications in the context of signal coding and analog-to-digital conversion. It has been considered in many earlier works, e.g., [3]-[7]. The common factor among all quantization algorithms is to exploit the null-space of the frame operator to minimize the quantization or reconstruction error. In [3], constraints on the quantized coefficients were set to satisfy consistent reconstruction conditions that was shown to achieve $O(1/r^2)$ performance, where r denotes the logarithm of the frame oversampling factor. Noise shaping of the quantization noise, that mimics sigma-delta converters [8], is used in [5] and [6]. In [5], the quantization noise at time τ is projected onto the following sample in an approximation to first-order sigma-delta converters. This was generalized in [6] where higher order noise shaping filters are used. The variance of the quantization noise acts as $O(1/r^2)$. A different approach for frame quantization was proposed in [7], where the quantization is modeled as a quadratic integer programming problem and a greedy procedure for trellis quantization was developed to solve the problem. The reported quantization performance was better than $O(1/r^2)$. The trellis quantization procedure could be combined with other quantization algorithms, e.g., noise shaping, for improved performance.

In this work, we extend the trellis quantization procedure in [7] to *turbo* quantization. Turbo quantization refers to an iterative quantization scheme, with monotonically decreasing quantization error. The turbo quantization procedure has three main components:

1. The *forward-backward* algorithm that is run at each iteration. This algorithm is a generalization of the trellis quantization procedure in [7].
2. *Reordering* of the frame coefficients, where at each iteration we reorder the frame coefficients to randomize the quantization error of successive coefficients.
3. *Extrinsic* quantization information that is exchanged between iterations to improve the quantization error.

The proposed turbo quantization procedure resembles turbo coding in the context of error control coding (and hence it gets this name). It is shown to provide significant quantization improvement and the quantization noise variance is shown to be better than $O(1/r^5)$ with guaranteed stability. This quantization behavior is (to our knowledge) the best reported results in the literature. In our analysis, we focus on finite-dimensional frames but this could be extended to infinite frames using sliding windows.

The paper is organized as follows. In section 2 we provide necessary background of the optimization model of the quantization problem and the trellis quantization procedure. Then, in section 3, we introduce the forward-backward quantization algorithm, followed by the turbo quantization algorithm in section 4. Finally in section 5, the proposed algorithm is evaluated and compared with prior art using finite tight frames. Throughout the paper we will use the following notation. Bold lower-case letters denote column vectors, and c_i denotes the i-th element of $\mathbf{c}$. $\mathbf{c}'$ denotes the transpose of $\mathbf{c}$. $\langle .,. \rangle$ denotes the inner product defined on the corresponding Hilbert space. We assume all vectors and matrices are real-valued. Additional notations are introduced when needed.

2. BACKGROUND

2.1. Frames

A set $\{\phi_j\}_{j=1}^N$ constitutes a frame in the finite-dimensional Hilbert space $\mathcal{H}$ if there exist two positive real numbers A and B such that for any $\mathbf{x} \in \mathcal{H}$

$$A\,\|\mathbf{x}\|^2 \leq \sum_{j\in J} |\langle \mathbf{x}, \phi_j \rangle| \leq B\,\|\mathbf{x}\|^2 \tag{1}$$

The frame is overcomplete if the dimension of $\mathcal{H}$ (denote it by K) is less than N. and we have

$$\mathbf{x} = \sum_{j=1}^{N} c_j \phi_j \tag{2}$$

where $\mathbf{c}$ is in general not unique if $K < N$. This redundancy could be exploited to optimize the quantization noise of overcomplete frames. In the following discussion, we assume finite-dimensional frames, i.e., $N < \infty$.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

2.2. Optimization Model

The quantized frame coefficients $\widetilde{\mathbf{c}}$ could be expressed as:

$$\widetilde{\mathbf{c}} = \lfloor \mathbf{c} \rfloor + \boldsymbol{\alpha} \quad (3)$$

where $\lfloor \cdot \rfloor$ denotes the floor integer operator, and $\boldsymbol{\alpha}$ is an $N \times 1$ vector whose entries $\{\alpha_i\}_{i=1}^{N}$ belong to a finite set of integers whose size depends on the number of least significant bits that could be varied in the optimization problem. In the simplest case (which we use in the remainder of the paper), $\alpha_i \in \{0, 1\}$, which corresponds to varying only the least significant bit of the quantized coefficients. The objective of the quantization optimization is to find $\boldsymbol{\alpha}$ that minimizes the reconstruction error after quantization. It was shown in [7] that, the quantization optimization problem is a quadratic integer programming problem of the form

$$\Psi = \langle L(\mathbf{e} - \boldsymbol{\alpha}), (\mathbf{e} - \boldsymbol{\alpha}) \rangle \quad (4)$$

where L is a self-adjoint operator (which, in case of minimizing the mean reconstruction error, equals the frame operator), and

$$\mathbf{e} \triangleq \mathbf{c} - \lfloor \mathbf{c} \rfloor \quad (5)$$

2.3. Trellis Quantization

The trellis quantization procedure [7] is a greedy solution to the optimization problem in (4). The objective function could be expressed recursively as [7]:

$$J_\tau = J_{\tau-1} + |\varepsilon_\tau|^2 \langle \phi_\tau, \phi_\tau \rangle + 2 \sum_{t<\tau} \varepsilon_t \varepsilon_\tau \langle \phi_t, \phi_\tau \rangle \quad (6)$$

where $\Psi = J_N$ and $\boldsymbol{\varepsilon}$ is the quantization error vector that is defined as

$$\boldsymbol{\varepsilon} \triangleq \mathbf{e} - \boldsymbol{\alpha} \quad (7)$$

Note that, J_τ could be regarded as the reconstruction error when the coefficients from $\tau + 1$ to N are not quantized. The trellis quantization procedure uses a trellis of N steps with the number of states at step τ equals the possible quantization values of α_τ. In the binary case, which we consider here, we have two states that represent 0 and 1. Note that, in the more general case we may combine more than one coefficient in a single step. If l coefficients are combined, then we have N/l steps with 2^l states per step. The optimal path through the trellis yields the quantization values of all coefficients. The cost function that is evaluated on the trellis is an approximation of (6). Each state at time τ keeps track of the best path that leads to it. The recursive metric for the k-th state at step τ is computed as:

$$J_\tau^{(k)} = \min_i [J_{\tau-1}^{(i)} + \sum_{t<\tau} 2\varepsilon_t^{(i)} \varepsilon_\tau^{(k)} \langle \phi_t, \phi_\tau \rangle] + |\varepsilon_\tau^{(k)}|^2 \langle \phi_\tau, \phi_\tau \rangle] \quad (8)$$

Note that, $\{\varepsilon_t^{(i)}\}_{t<\tau}$ specifies the trellis path up to state i at time $\tau - 1$. If the best previous state is p, then the updated path for state k at τ is specified by:

$$\varepsilon_t^{(k)} = \varepsilon_t^{(p)}, \qquad \text{for } t = 1, 2, ..., \tau - 1 \quad (9)$$

and $\varepsilon_\tau^{(k)}$ is specified by the quantization value of the k-th state at step τ. The algorithm is suboptimal because the relation between the quantization of the τ-th coefficient and the quantization of coefficients of index less than $\tau - 1$ is evaluated only through the trellis path ending at step $\tau - 1$. Therefore not all possible combinations of this quantizations are tested.

2.4. Comments

The trellis expansion can be viewed as a special case of the common branch-and-bound algorithm for discrete optimization [9]. It could be slightly modified to accommodate the conventional branch-and-bound algorithm by investigating all possible paths at each step and allowing the best two to survive (rather than allowing one path per state). In this case, the number of survived paths could be increased to more than two.

The trellis expansion as described in the previous section has three problems:

1. Some potentially good paths are pruned prematurely if they do not satisfy (8).
2. The best path is dependent on the ordering of the frame coefficients.
3. The expansion assumes we have no quantization errors in future frame coefficients.

In the following, we describe two algorithms: the forward-backward algorithm and the turbo quantization algorithm that efficiently address these issues and provide significant improvement in frame quantization.

3. THE FORWARD-BACKWARD ALGORITHM

The forward-backward algorithm was introduced in [10] in the context of Maximum-a-Posterior (MAP) decoding of convolutional codes. It has also been used in the expansion of Hidden Markov Models (HMM) in the speech recognition context [11]. In an abstract way, the algorithm has three components: forward expansion, backward expansion, and combining forward and backward metrics. For our quantization purposes the proposed forward-backward algorithm follows the same line.

The forward expansion of the forward-backward algorithm is the same as the trellis expansion. The backward algorithm operates similarly but with the coefficients in *reverse* order. The metric recursion (analogous to (8)) can be expressed as

$$R_\tau^{(k)} = \min_i [R_{\tau+1}^{(i)} + \sum_{t>\tau} 2\varepsilon_t^{(i)} \varepsilon_\tau^{(k)} \langle \phi_t, \phi_\tau \rangle] + |\varepsilon_\tau^{(k)}|^2 \langle \phi_\tau, \phi_\tau \rangle] \quad (10)$$

where $\varepsilon_t^{(i)}$ is defined as in (8) for the i-th state at $\tau + 1$. The k-th state at time τ keeps track of the path $\{\varepsilon_t^{(k)}\}_{t \geq \tau}$ that leads to the state. Note that, the final recursion value $R_1^{(k)}$ has the reconstruction error due to the quantization of the path leading to k-th quantization value of the first coefficient.

After the forward and backward expansions are completed, we have two sets $\{J_\tau^{(k)}, R_\tau^{(k)}\}$ for $1 \leq \tau \leq N$, along with the corresponding quantization paths. Note that, $J_\tau^{(k)}$ denotes the reconstruction error for the quantization path that leads to state k at step τ assuming that $\varepsilon_t = 0$ for $t > \tau$ (i.e., with no quantization error of future coefficients). Similarly, $R_\tau^{(k)}$ denotes the reconstruction error for the backward quantization path that leads to state k at step τ assuming that $\varepsilon_t = 0$ for $t < \tau$. Combining the forward and backward metrics is performed at *each* trellis step. The k-th state at step τ of the forward expansion is combined with the j-th state of step $\tau + 1$ of the backward expansion to have a complete quantization path. If $\{\varepsilon_t^{(k)}\}_{t \leq \tau}$ is the forward path and $\{\varepsilon_t^{(k)}\}_{t > \tau}$ is the backward path, then the forward-backward metric in the transition from state k at step τ to state j at step $\tau + 1$ is

$$\Psi_\tau^{(k,j)} = J_\tau^{(k)} + R_{\tau+1}^{(j)} + 2 \sum_{t \leq \tau} \sum_{l > \tau} \varepsilon_t^{(k)} \varepsilon_l^{(j)} \langle \phi_t, \phi_l \rangle \quad (11)$$

Note that $\Psi_\tau^{(k,j)}$ is the reconstruction error with the combined quantization path

$$\boldsymbol{\varepsilon}_\tau^{(k,j)} = \left[\varepsilon_1^{(k)}, \ldots, \varepsilon_\tau^{(k)}, \varepsilon_{\tau+1}^{(j)}, \ldots, \varepsilon_N^{(j)}\right]' \tag{12}$$

Hence, $\Psi_\tau^{(k,j)}$ could alternatively be computed from (4) as

$$\Psi_\tau^{(k,j)} = \langle L\boldsymbol{\varepsilon}_\tau^{(k,j)}, \boldsymbol{\varepsilon}_\tau^{(k,j)} \rangle \tag{13}$$

The optimal quantization path for the forward-backward algorithm is $\boldsymbol{\varepsilon}_{\hat{\tau}}^{(\hat{k},\hat{j})}$, where

$$(\hat{\tau}, \hat{k}, \hat{j}) = \underset{\tau,k,j}{\operatorname{argmin}}\ \Psi_\tau^{(k,j)} \tag{14}$$

Note that, the forward-backward algorithm investigates all paths at all steps in the trellis expansion even the ones that are pruned later in the forward and backward expansions. Therefore, it addresses the first problem of the trellis expansion that was discussed in the previous section. The other issues are addressed by the turbo quantization procedure in the following section.

The forward-backward algorithm comprises two trellis expansions plus the merging step. It has roughly three times the complexity of the trellis quantization procedure.

4. TURBO QUANTIZATION

The quantization path, and hence the reconstruction error, of the trellis expansion and the forward-backward algorithm depends on the *ordering* of the frame coefficients because of the continuous pruning of paths according to the value of the corresponding metric. This dependency could be reduced if the algorithms are run multiple times with different ordering of the frame coefficients. The chosen quantization path corresponds to the one with the minimum overall metric value. However, this ordering should be carefully performed to have reasonable correlation between successive coefficients. The choice of the first coefficient is totally arbitrary, but successive coefficients are sorted according to the correlation with earlier ordered coefficients. In particular, assume we have ordered coefficients up to step τ, with order $\pi_1, \pi_2, \ldots, \pi_\tau$, then the coefficient at $\tau+1$ is chosen from the remaining coefficients such that it has maximum correlation with earlier coefficients. Let

$$\Pi_\tau \triangleq \{\pi_1, \pi_2, \ldots, \pi_\tau\} \tag{15}$$

then the $(\tau+1)$-th coefficient is chosen according to

$$\pi_{\tau+1} = \underset{k \notin \Pi_\tau}{\operatorname{argmax}} \sum_{i \le \tau} W_i \|\langle \phi_k, \phi_{\pi_i} \rangle\|^2 \tag{16}$$

where $\{W_i\}$ are appropriate weighting to give more emphasis to the most recent coefficients. Note that, coefficient ordering at each turbo iteration resembles the interleaver in turbo coding schemes [12], whose purpose is to randomize the channel errors between the two constituent convolutional codes. Random interleaving usually yield better performance. In our context, we have more degrees of freedom because we do not have constituent codes and we could have a different ordering for every iteration.

Running the trellis quantization or the forward-backward algorithms multiple times would improve the performance. However, the critical step that contributes to the superior performance of turbo quantization is the propagation of quantization estimates through turbo iterations such that the reconstruction error is monotonically decreasing. In other words, the quantization path at turbo iteration l is passed as *extrinsic* information to the trellis expansion for turbo iteration $l+1$, which has a different ordering of the frame coefficients. This extrinsic information is exploited by the constituent trellis quantization procedure or the forward-backward algorithm such that the reconstruction error is monotonically decreasing (or non-increasing). This resembles the extrinsic information that carries the a priori probabilities of information bits in the turbo coding context [12].

To enable the deployment of the extrinsic information across turbo iterations, we introduce slight modification to the trellis expansion. Recall that, in the original trellis quantization procedure in section 2.3, the recursive metric $J_\tau^{(k)}$ represents the reconstruction error due to the quantization path that leads to state k at step τ *assuming* that the quantization error of future samples is zero. Let $\{\varepsilon_t^{(k)}\}_{t \le \tau}$ be the quantization path to state k. Define

$$\boldsymbol{\xi}_\tau^{(k)} \triangleq [\varepsilon_1^{(k)}\ \ldots\ \varepsilon_\tau^{(k)}\ 0 \ldots 0]' \tag{17}$$

which has $N-\tau$ zeros. Then it is straightforward from (4) to show that $J_\tau^{(k)}$ in (8) could be expressed as

$$J_\tau^{(k)} = \langle L\, \boldsymbol{\xi}_\tau^{(k)}, \boldsymbol{\xi}_\tau^{(k)} \rangle \tag{18}$$

which resembles (13) for the forward-backward algorithm. Assume that, the best quantization path after the turbo iteration l is

$$\boldsymbol{\eta}^{(l)} = [\eta_1^{(l)}\ \ \ldots\ \eta_N^{(l)}]' \tag{19}$$

where $\eta_i \triangleq e_i - \alpha_i$ (which is defined similar to ε_i) is introduced for notational convenience. Let $\boldsymbol{\Pi}_N^{(l)}$ denote the coefficient ordering for iteration l as in (15). The extrinsic information about quantization is propagated from one turbo iteration to the following by modifying $\boldsymbol{\xi}_\tau^{(k)}$ in (17) at turbo iteraion l as

$$\tilde{\boldsymbol{\xi}}_\tau^{(k)}(l) \triangleq [\varepsilon_{\pi_1^{(l)}}^{(k)}\ \ldots\ \varepsilon_{\pi_\tau^{(l)}}^{(k)}\ \eta_{\pi_{\tau+1}^{(l)}}^{(l-1)} \ldots \eta_{\pi_N^{(l)}}^{(l-1)}]' \tag{20}$$

i.e., for future samples we use the optimal quantization error from the earlier turbo iteration rather than zero. The corresponding recursive metric becomes

$$J_\tau^{(k)}(l) = \langle L\, \tilde{\boldsymbol{\xi}}_\tau^{(k)}(l),\ \tilde{\boldsymbol{\xi}}_\tau^{(k)}(l) \rangle \tag{21}$$

and we have similar relations for the backward part of the forward-backward algorithm in the turbo iteration l:

$$\begin{aligned} \tilde{\boldsymbol{\zeta}}_\tau^{(k)}(l) &\triangleq [\eta_{\pi_1^{(l)}}^{(l-1)}\ \ldots\ \eta_{\pi_{\tau-1}^{(l)}}^{(l-1)}\ \varepsilon_{\pi_\tau^{(l)}}^{(k)}\ \ldots\ \varepsilon_{\pi_N^{(l)}}^{(k)}]' && (22)\\ R_\tau^{(k)}(l) &= \langle L\, \tilde{\boldsymbol{\zeta}}_\tau^{(k)}(l),\ \tilde{\boldsymbol{\zeta}}_\tau^{(k)}(l) \rangle && (23) \end{aligned}$$

It is straightforward to show that the overall reconstruction error of the forward-backward algorithm is non-increasing with turbo iterations since the solution of the earlier turbo iteration can always be found by setting $\varepsilon_{\pi_\tau^{(l)}}^{(k)} = \eta_{\pi_\tau^{(l)}}^{(l-1)}$ for states on the optimal path of iteration $l-1$. This choice is a subset of all possible choices at all steps in the forward-backward algorithm. Hence, the reconstruction error is non-increasing.

From the above discussion, the turbo quantization procedure could be summarized in the following steps:

1. Run the forward-backward algorithm once on the frame coefficients to compute $\boldsymbol{\eta}^{(1)}$, set the turbo iteration index to $l=2$.
2. Pick a *random* start coefficient and order the rest of the frame coefficients as in (16) to compute $\boldsymbol{\Pi}_N^{(l)}$.

3. Run the forward-backward algorithm with the new ordering $\mathbf{\Pi}_N^{(l)}$ and the extrinsic information $\boldsymbol{\eta}^{(l-1)}$ as in (21) and (23) to update $\boldsymbol{\eta}^{(l)}$ as in (14).

4. If the maximum number of turbo iterations is reached output $\boldsymbol{\eta}^{(l)}$ as the final quantization path, otherwise increment l and go to step 2.

Denote the final metric of the forward-backward algorithm at iteration l by $\Psi^{(l)}$. The turbo iterations guarantee that $\Psi^{(l+1)} \leq \Psi^{(l)}$. Since $\{\Psi^{(l)}\}$ is a monotonic *bounded* sequence, it is guaranteed to converge to the optimal solution by the monotone convergence theorem [13]. Further, unlike the sigma-delta frame quantization, e.g., [5, 6], the proposed turbo quantization does not have stability issues as the quantization error is always bounded. Note that, the atomic trellis quantization procedure could be modified to allow more general configurations, e.g., using more than two quantization levels or including the quantization of more than one coefficient in the state, without changing the structure of the turbo quantization algorithm.

The complexity of coefficient ordering and the overhead of the metric computation with the extrinsic information are negligible. Therefore, the complexity of the turbo quantization algorithm is roughly the complexity of the forward-backward algorithm multiplied by the number of turbo iterations.

5. SIMULATION RESULTS

We tested the proposed quantization algorithms using geometrically uniform tight frames that are generated as described in [3]. The test vectors in R^K are zero-mean Gaussian iid sequences with unity variance. The coefficients are normalized to the maximum value prior to quantization. The frame functions $\{\phi_j\}$ in the synthesis process are not quantized. In the following, r denotes the redundancy factor (which equals $\log_2(N/K)$). In Fig. 1, we compare all the procedures introduced in the paper, with 8 bits of quantization (we get similar results with different resolutions). We also included the rounding results, which behave as $O(1/r)$. In this figure, we use a trellis diagram with four states at each step by combining two coefficients together. As noticed from the figure, the forward-backward algorithm performs significantly better than the trellis quantization in [7]. The multiple forward-backward algorithm runs multiple forward-backward iterations with different ordering and picks the best one (but without passing extrinsic information). The improvement of the turbo quantization against multiple forward-backward algorithms is due to the extrinsic information. In both cases we run 20 iterations. The performance of the turbo quantization algorithm is on average better than $O(1/r^5)$, and it is better than $O(1/r^6)$ at high r; a significant improvement over earlier reported results, e.g., [3]-[6].

In Fig. 2, we show a typical example for the improvement of the reconstruction error with turbo iteration. In Fig 2a, we show the reconstruction error with different redundancies versus the number of turbo iterations. Note that, there is incremental change when doubling the turbo iterations from 10 to 20 iterations. This is manifested in Fig 2b where we show the average cumulative reconstruction error improvement versus the number of turbo iterations when $r = 7$. It resembles the EXIT charts in turbo coding context [14]. Most of the error improvement takes place in the first 10 iterations.

Note that, the performance of proposed procedures could be further improved by increasing the complexity of the underlying quantization trellis by allowing more quantization levels or grouping more coefficients in a single trellis step.

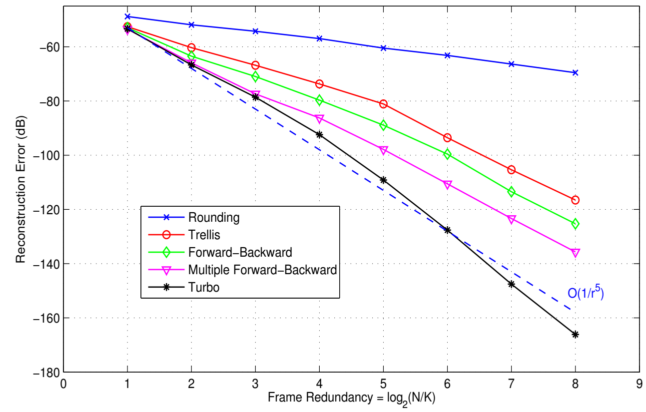

Fig. 1. Performance of frame quantization algorithms with $K = 4$

.

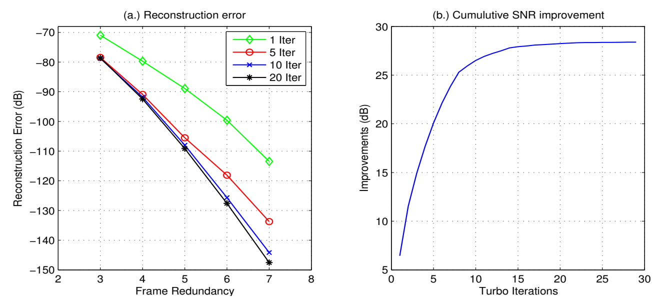

Fig. 2. Performance Improvements with Turbo Iterations

.

6. DISCUSSION

We presented a new concept for quantization, which we called *turbo quantization*, that uses a simple core quantization procedure and monotonically improves the quantization with turbo iterations. For the core quantization procedure, we introduced the forward-backward algorithm that generalizes trellis quantization by allowing expansion in the forward and reverse directions, and investigating the quantization paths at all steps. In addition to the core quantization procedure, the turbo quantization has two components: coefficient reordering to achieve error randomization across turbo iterations, and exchanging extrinsic quantization information across iterations to monotonically improve the optimization metric. The convergence and stability of the proposed procedure are always satisfied. The complexity of the proposed algorithm is linear with the frame size and the number of turbo iterations. The algorithm is scalable by varying the number of turbo iterations or the complexity of the core quantization algorithm. The turbo quantization algorithm can be regarded as a feedback procedure at the whole frame level rather than a sample-by-sample feedback in noise-shaping procedures. It was shown to provide significantly better quantization performance than earlier reported results.

Although the discussion focused on finite-dimensional Hilbert spaces, the results could be generalized to infinite-dimensional spaces using sliding quantization windows that are commonly used in the context of error control coding for long information blocks. Further, the core quantization procedure could be generalized to other integer programming techniques, e.g., branch-and-bound, that could exploit the extrinsic information for improved performance.

7. REFERENCES

[1] R. J. Duffin and A. C. Schaeffer, “A Class of non-harmonic Fourier series”, Trans. American Math. Soc., vol. 72, no. 2, pp. 341-366, Mar. 1952.

[2] I. Daubechies, *Ten Lectures on Wavelets*, SIAM-CBMS, 1992.

[3] V. Goyal, M. Vetterli, N. Thao, “Quantized Overcomplete Expansions in R^N, Analysis, Synthesis, and Algorithms”, IEEE Trans. on Information Theory, vol. 44,no. 1, pp. 16–31, Jan. 1998.

[4] B. Lazano, A. Ortega, “Efficient Quantization of Overcomplete Expansion in R^N”, IEEE Trans. on Information Theory, vol. 49, no. 1, pp. 129–150, Jan. 2003.

[5] P. Boufounos, A. Oppenheim, “Quantization noise shaping on arbitrary frame expansion”, IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, ICASSP, pp. 205-208, 2005.

[6] F. Abdelkefi, “Performance of sigma-delta quantizations in finite frames”. IEEE Transactions on Information Theory, vol. 56, no. 8, pp. 4157–4165, Aug. 2010.

[7] M. Mansour, “Trellis quantization of frames”, IEEE Intl. Conf. on Acoust., Speech, and Sig. Proc., ICASSP, pp. 4002-4005, 2010.

[8] R. Schreier, C. Gabor, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE press, 2004.

[9] E. Lawler and D. Wood, “Branch-And-Bound Methods: A Survey” , Operations Research, vol. 14, no. 4, pp. 699-719, Jul. 1966.

[10] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for minimizing symbol error rate”, IEEE Transactions on Information Theory, vol. 20, no. 2, pp.284-287, Mar. 1974.

[11] L. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition”. Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[12] C. Berrou, A. Glavieux, P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes”, IEEE Intl. Conf. on Comm. (ICC), vol.2, pp. 1064-1070, 1993.

[13] R. Bartle, *The Elements of Real Analysis*, John Wiley & Sons, second edition, 1976.

[14] S. Brink, “Convergence of iterative decoding”, Electron. Letters, vol. 35, pp. 806-808, May 1999.