Modified BTC Algorithm for Audio Signal Coding

Stefan TOMIĆ, Zoran PERIĆ, Jelena NIKOLIĆ
University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia
stefan@elfak.rs, zoran.peric@elfak.ni.ac.rs, jelena.nikolic@elfak.ni.ac.rs

***Abstract*—This paper describes modification of a well-known image coding algorithm, named Block Truncation Coding (BTC) and its application in audio signal coding. BTC algorithm was originally designed for black and white image coding. Since black and white images and audio signals have different statistical characteristics, the application of this image coding algorithm to audio signal presents a novelty and a challenge. Several implementation modifications are described in this paper, while the original idea of the algorithm is preserved. The main modifications are performed in the area of signal quantization, by designing more adequate quantizers for audio signal processing. The result is a novel audio coding algorithm, whose performance is presented and analyzed in this research. The performance analysis indicates that this novel algorithm can be successfully applied in audio signal coding.**

***Index Terms*—adaptive coding, audio compression, correlation, quantization, signal to noise ratio.**

I. Introduction

Audio signal is one of the most important signals in telecommunications, transmitting a great amount of information daily. Therefore, it is very important that this information has high quality, with the usage of least resources possible. As storing a high quality audio signal demands a lot of memory space, the application of compression algorithms is a necessity in any system working with audio [1], [2]. By applying a compression algorithm, digital representation of audio signal is created, which occupies less memory space than the original signal. Compressed signal quality depends on the bit rate used, which is implicitly specified by signal's application [1]-[3]. Efficient usage of the available bit rate is crucial for achieving higher quality of compressed signal. This gives a great significance to research and development of audio compression algorithms. Although there are a lot of audio coding algorithms, the application of an image coding algorithm to audio is a novelty and a challenge. The novel audio signal coding algorithm, that is the topic of this research, is result from applying some adjustments to the original Block Truncation Coding (BTC) algorithm [4], while preserving the basic idea of the algorithm.

Audio compression algorithms are a need and necessity in signal processing. They represent a need, since the raw audio data requires large amount of resources for storage. On the other hand, they are a necessity, as modern communications enable capturing, using and transmitting audio files on every communication device. Although the bandwidth of new generation telecommunications systems constantly increases, the need for high quality compressed audio signal is present and it is obligating researchers for further studies in this field. Mass usage of digital audio began with the introduction of the compact disc (CD), in 1982 [5]. Compact disk introduced a high quality audio signal, but this came with the usage of high data rates. CD and digital audio tape (DAT) are sampled, typically at 44.1 kHz or 48 kHz, by using pulse code modulation (PCM), with a bit rate of 16 bits per sample [6]. Although the first generation systems were optimized for working with these data rates, problems occurred with the next generation multimedia and wireless systems. As wireless technologies use limited bandwidth, high bit rate signals needed to be compressed, while preserving high signal quality. Audio coding can be lossless or "lossy" [2], [7]. Lossless audio coding implies that the original information can be restored from the compressed signal without any loss of data. This technique is widely used in many coding standards as: Free Lossless Audio Codec (FLAC), MPEG-4 Audio Lossless Coding (MPEG-4 ALS), Windows Media Audio Lossless (WMAL), etc. [8]. When lossy compression is applied, the original data cannot be restored, although this coding technique can still provide high quality audio signal. Some of the most used loosy audio coding standards are: MPEG-2 Audio Layer III (MP3), Advanced Audio Coding (AAC), Dolby Digital, etc.

This paper presents a modified BTC algorithm and its application to audio signal coding. While the aforementioned coding standards are designed with complex and time consuming codecs, the modified BTC algorithm is characterized by low complexity, suitable for implementation in low delay applications. BTC represents an efficient coding algorithm, originally designed for image compression [4]. We show that the BTC algorithm, with certain modifications, can be successfully applied in audio signal coding. To the best of the authors' knowledge, the application of the BTC algorithm in audio signal coding has not been considered so far in the literature. Recent research has showed that the idea of BTC algorithm can be successfully applied in speech signal coding [9]. The high quality of the obtained output speech signal encouraged us to continue the research and apply the idea of the BTC algorithm in audio signal coding. We do not consider straightforward application of the BTC algorithm, but we modify this algorithm to be suitable for the implementation in audio signal coding. The original BTC algorithm utilizes the correlation in the adjacent pixel values. Correlation in the input signal samples can be utilized in speech signal coding, as was demonstrated in [10]. Along with the image compression, BTC algorithm can be implemented in

This work is supported by Serbian Ministry of Education, Science and Technologic Development (Project TR 32035).

Digital Object Identifier 10.4316/AECE.2016.04005

content-based image retrieval from an image database [11]. In this paper, we exploit the correlation which exists between audio signal samples, enabling successful application of the BTC algorithm in audio signal coding. The fact that we have not found in literature that the BTC algorithm is successfully applied in audio signal coding was also a big motivation for this research.

BTC algorithm is based on the input signal decomposition into non overlapping blocks of data, named frames [4]. Frame size is highly important, since it determines the side information that is processed with the audio signal samples. Side information consist of the mean value of the input signal frame and standard deviation of the difference signal frame, which differ from frame to frame. Less side information implies using smaller bit rate for coding, which can be achieved by using larger frame size. However, an arbitrary large frame size cannot be taken, since it can degrade quality of the audio signal as we show in numerical results. Moreover, it is well known that a larger frame size could cause signal latency, when applied in real time signal processing [1].

The original BTC algorithm utilizes three uniform quantizers, using the total bit rate of 2 bits per sample. Mean value of the frame and standard deviation are both quantized by using 8 bits per frame. By using frame size of 4x4, single frame consists of 16 input signal samples, for which the side information is represented by using 16 bits. Each sample is quantized by a two level quantizer, with the bit rate of 1 bit per sample. Additionally, representing the side information requires 1 bit per sample, which outputs to total bit rate of 2 bits per sample. Since using 2 bits per sample for coding implies low output signal quality, we consider coding schemes which utilize higher bit rates.

This research emphasis on quantizer design and its application in the modified BTC algorithm for audio coding. Unlike the original algorithm, the modified BTC algorithm uses three different quantizers, which process different parameters of the input audio signal. The mean value of the frame of input samples is quantized by the compander defined with μ law compression [12]. Quantization of the standard deviation of the difference signal frame is performed by the log-uniform quantizer [13]. The difference between the amplitude of the input signal and the quantized mean of the frame of input samples is quantized by using the forward adaptive optimal compander [14], [15]. Such quantizer choice produced a low complexity system, based on simple scalar quantization techniques, unlike [16], where the audio signal compression was performed with the application of highly complex vector quantization techniques.

The rest of the paper is organized as follows. Section 2 provides a description of the modified BTC algorithm. Quantizer design and application is presented in Section 3. Section 4 is dedicated to the numerical results and detail analysis. Conclusion and final remarks are shown in Section 5, along with the future research directions.

II. MODIFIED BTC ALGORITHM

As we have already noticed, the original BTC algorithm is developed for image compression, so that the successful application in audio signal coding required some modifications, which are described in this paper. In the modified BTC algorithm design, the basic idea of the original BTC algorithm has been preserved, while the majority of changes were performed in the area of input samples quantization process.

As mentioned in Section 1, the original BTC algorithm is based on decomposing the input signal stream into blocks of data, named frames [4]. The parameters of a frame are processed separately, for each of the frames. Utilizing larger frame sizes results in performing less number of computations, since dividing an input signal into larger frames, outputs a smaller total number of frames. This leads to less computations overall, since the frame parameters (mean value and standard deviation) are calculated and quantized smaller number of times. However, as we highlighted in the previous section, an arbitrary large block size cannot be taken, since it can degrade quality of the audio signal. In particular, the parameters estimated for very large frames possibly differ much from those estimated for the smaller frames, which better follow the statistics of the input signal [1]-[3]. Therefore, choosing appropriate frame size is highly important, which we took into consideration in the numerical results analysis.

The modified BTC algorithm, for which the encoder and decoder are shown in Fig. 1, works as follows. The input signal $x = x_i^{(j)}$, $i = 1, 2,\ldots,M$, $j = 1,2,\ldots,L$ is divided into L frames of M samples that are brought to the Buffer 1, which reads one frame at a time. Then, for each frame, the mean value of the input signal samples, $\bar{x}_m^{(j)}$, is estimated and quantized by the compander defined with the μ compression law (Encoder 1 and Decoder 1). The quantized mean of the frame is used for creating the difference signal, by subtracting it from the current sample amplitude $x_i^{(j)}$, for each frame:

$$d_i^{(j)} = x_i^{(j)} - \hat{\bar{x}}_m^{(j)}, i = 1,2,\ldots,M, j = 1,2,\ldots,L, \quad (1)$$

where M denotes the frame size and L denotes the total number of frames. After this step, the thus defined difference signal is processed. The fact that the difference signal has lower amplitude dynamics than the original input signal makes it more suitable for coding from the compression point of view. The difference signal frame is firstly loaded into Buffer 2, after which standard deviation of the frame is estimated, and used for adapting the quantizer of the difference signal. Standard deviation of the difference signal frame is quantized by the log-uniform quantizer, i.e. by the Encoder 2 and Decoder 2 shown in Fig. 1.

As mentioned, the difference signal frame, $d_i^{(j)}$, is quantized with the application of the forward adaptive optimal compander. The forward adaptive quantization can be performed in two phases [17]. Firstly, the fixed representatives, $\hat{d}_i^{(j)}$, are obtained, as the output of the fixed-rate optimal compander. Adaptive representatives, $\hat{d}_i^{a\,(j)}$, are then obtained by multiplying the fixed representatives by the quantized standard deviation of the difference signal frame:

$$\hat{d}_i^{a\,(j)} = \hat{d}_i^{(j)} \times \hat{\sigma}^{(j)}, i = 1,2,\ldots,M, j = 1,2,\ldots,L, \quad (2)$$

where $\hat{\sigma}^{(j)}$ stands for the quantized standard deviation for the current jth difference signal frame. Forward adaptation can be also performed by normalizing the difference signal

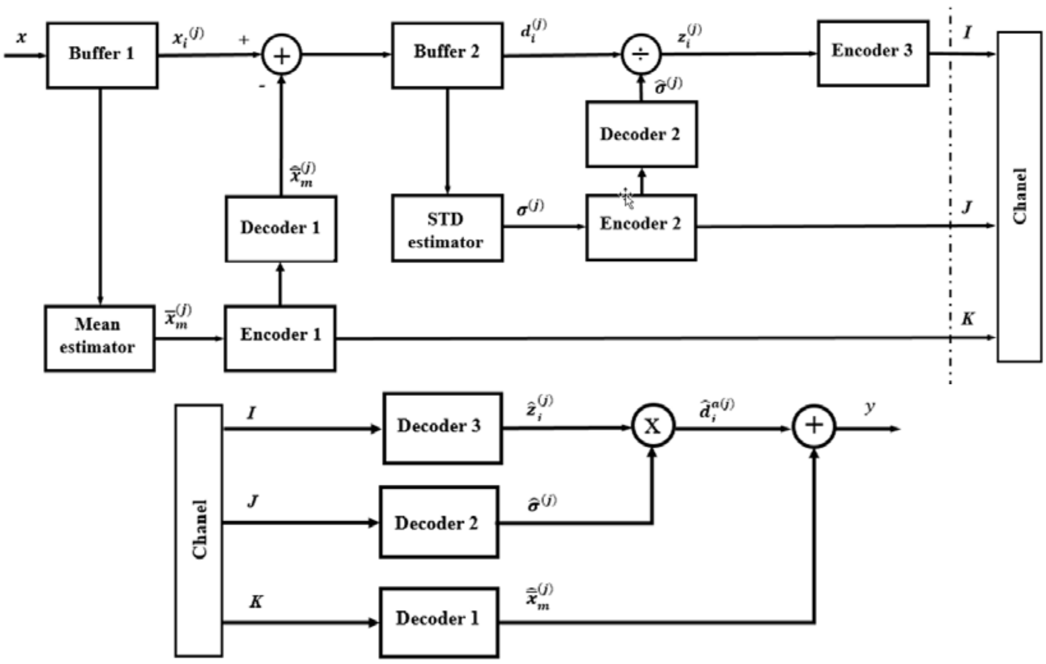


Figure 1. Block diagram of encoder (above) and decoder (bellow) of the modified BTC algorithm

with $\hat{\sigma}^{(j)}$ after which so obtained normalized samples, $z_i^{(j)}$, can be quantized by a fixed-rate optimal compander (Encoder 3 and Decoder 3), commonly designed for the unit variance. The outputs of the aforementioned encoders are sent through the channel as binary information, denoted by I, J and K. The encoded values are received and decoded for the purpose of reconstructing the input signal. After the side information is decoded, a denormalization procedure, which results in the output of the forward adaptive quantizer, $\hat{d}_i^{a\,(j)}$, is provided. The quantized output signal samples $y = y_i^{(j)}$, $i = 1, 2,\ldots, M$, $j = 1, 2, \ldots, L$ are reconstructed by adding the decoded adaptive representatives from Eq. (2) to the decoded quantized mean value of the current frame:

$$y_i^{(j)} = \hat{d}_i^{a\,(j)} + \hat{\bar{x}}_m^{(j)},\ i = 1,2,\ldots,M,\ j = 1,2,\ldots,L\,, \tag{3}$$

This completes the procedure of the input samples quantization, after which we are able to measure the performance of the system we proposed. Objective performance was analyzed by observing the signal to quantization noise ratio (SQNR), in the cases of using different bit rates.

Total bit rate of the system depends on the frame size and the amount of bits per sample used for representing the side information. Increasing bit rate for representing the mean value of the frame increases the total bit rate, which is calculated as:

$$R = r_3 + \frac{r_1 + r_2}{M}\,, \tag{4}$$

where r_3 denotes the bit rate used for fixed rate coding the difference signal, r_1 denotes the bit rate used for coding the mean value of the input signal samples frame, r_2 denotes the bit rate used for coding the standard deviation of the difference signal frame, while M denotes the frame size.

III. Quantizer Design

Quantization is an important part of digitalization process, so that overall performance of the system depends on the quantizer design [18]. When designing a quantizer, computational complexity and robustness has to be considered. Higher complexity can increase the amount of time required for calculations, which can limit the practical application of the algorithm. On the other hand, robustness implies that quantizer can provide almost constant quality of the quantized signal for input audio signals with different characteristics. This feature can be highly significant in processing audio signals with high amplitude dynamics [1], [2], [18].

Audio signal can be modelled well with a memoryless Laplacian source, with the mean value equal to zero. Probability density function (PDF) of a Laplacian source of variance σ^2 with the zero mean value is defined by [2], [19]:

$$p(x) = \frac{1}{\sqrt{2\sigma^2}} \exp\left\{ -\frac{|x|\sqrt{2}}{\sigma} \right\}, \tag{5}$$

The original BTC algorithm implements uniform quantization with low bit rate, which cannot provide high quality audio signal output (see section IV). Since two level quantization is not suitable for application in audio signal coding, some modifications are done for applying BTC algorithm to audio signal. Fig. 1 shows block diagram of the proposed modified BTC algorithm. The modified BTC algorithm utilizes three different quantizers. The mean value of the input signal frame is quantized by applying quasilogarithmic quantizer, which represents a logarithmic quantizer defined with the μ compression law. The designed quasilogarithmic quantizer, performs a signal compression by applying a compressor function defined as in [12], [18]:

$$c_\mu(x) = \frac{x_{\max}}{\ln(1+\mu)} \ln\left(1 + \mu \frac{|x|}{x_{\max}}\right) \operatorname{sgn}(x), \quad |x| \le x_{\max}, \quad (6)$$

where μ stands for the compression factor [18], while $x_{\max}$ denotes the support limit of the quantizer. In this paper, we implement the optimal support limit, determined for the quasilogarithmic quantizer with 64 quantization levels designed for the Laplacian source of unit variance. Logarithmic quantizers are robust and especially suitable for application in the case of high variance range of input signals, providing output of approximately constant SQNR [18]. We design the quantizer with the compression factor equal to 255, with 64 levels for representing the mean, which implies using 6 additional bits per frame for coding the mean value of the frame.

Standard deviation of the difference signal frame is quantized by the log-uniform quantizer with N_g quantization levels: [13]:

$$20\log\left(\hat{\sigma}^{(j)} = \hat{\sigma}_k^{(j)}\right) = 20\log_{10}(\sigma_{\min}) + \frac{2k-1}{2}\Delta^{\mathrm{lu}}, \quad (7)$$

where k=1,…,Ng, while the quant width is defined by:

$$\Delta^{\mathrm{lu}} = \frac{20\log_{10}\left(\dfrac{\sigma_{\max}}{\sigma_{\min}}\right)}{N_g}. \quad (8)$$

The log-uniform quantizer performs an uniform distribution of the standard deviation range (B) in dB [20log($\sigma_{\min}$), 20log($\sigma_{\max}$)] into N_g amplitude quants with the width equal to Δ^{lu} [13]:

$$\hat{\sigma}^{(j)} = \hat{\sigma}_k^{(j)} \,\Big|\, 20\log_{10}\left(\sigma^{(j)}\right) \in \left[V_k^{\mathrm{l}}[\mathrm{dB}], V_k^{\mathrm{h}}[\mathrm{dB}]\right], \quad (9)$$

$$V_k[\mathrm{dB}] = 20\log_{10}\left(\hat{\sigma}_k^{(j)}\right) = \left(V_k^{\mathrm{l}}[\mathrm{dB}] + V_k^{\mathrm{h}}[\mathrm{dB}]\right)/2, \quad (10)$$

where $k \in \{1, 2, ..., N_{\mathrm{g}}\}$, while the amplitude quant to which the current standard deviation belongs is determined by:

$$V_k^{\mathrm{l}}[\mathrm{dB}] = 20\log_{10}\left(\sigma_k^{\mathrm{l}}\right) = V_1^{\mathrm{l}}[\mathrm{dB}] + (k-1)|B|/N_{\mathrm{g}}, \quad (11)$$

$$V_k^{\mathrm{h}}[\mathrm{dB}] = 20\log_{10}\left(\sigma_k^{\mathrm{h}}\right) = V_1^{\mathrm{l}}[\mathrm{dB}] + k\,|B|/N_{\mathrm{g}}, \quad (12)$$

$$B = \bigcup_{k=1}^{N_{\mathrm{g}}} \left[V_k^{\mathrm{l}}[\mathrm{dB}], V_k^{\mathrm{h}}[\mathrm{dB}]\right], \quad k = 1, 2, ..., N_{\mathrm{g}}. \quad (13)$$

Since the designed log-uniform quantizer, as in [13] uses N_g = 16 quantization levels, we need 4 additional bits per frame for coding the standard deviation of the signal difference frame.

Third quantizer implemented in the modified BTC algorithm is forward adaptive optimal compander and it is used for quantization of the difference between the amplitude of the input signal and the quantized mean of the input samples frame. As we have already described, the quantization process was performed in two phases. Firstly, the fixed-rate optimal compander was utilized and the fixed representatives were obtained. Afterwards, we implemented the adaptation, according to the difference signal frame variance, for the purpose of obtaining the adaptive representation levels. With compander model, the signal is firstly compressed, thereafter uniformly quantized and lastly expanded (see Fig. 2), so that it holds $Q(x) = c^{-1}(Q_u(c(x))) = y$.

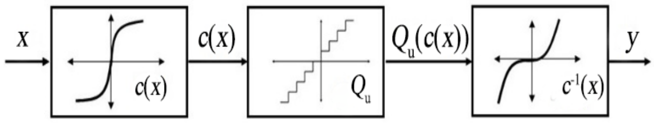


Figure 2. Block diagram of compander model

The optimal compander is designed for the optimal compressor function, which provides maximal SQNR for the variance assumed in designing the compander in question. The optimal compressor function can be defined as [15], [18]:

$$c(x) = t_{\max} \operatorname{sgn}(x) \frac{\int_0^{|x|} p^{1/3}(t)\,dt}{\int_0^{t_{\max}} p^{1/3}(t)\,dt}, \quad |x| \le t_{\max}, \quad (14)$$

where $t_{\max}$ denotes the maximal amplitude of the input signal x, for which the compander is designed. This compressor function performs mapping of [-$t_{\max}$, $t_{\max}$] into [$t_{\max}$, $t_{\max}$].

In this paper we implement a modified optimal compressor function, defined as:

$$c_{\mathrm{mod}}(x) = \begin{cases} -1, & x < -t_{\max} \\ -1 + 2\dfrac{\int_{-t_{\max}}^{x} p^{1/3}(t)\,dt}{\int_{-t_{\max}}^{t_{\max}} p^{1/3}(t)\,dt}, & x \in [-t_{\max}, t_{\max}], \\ 1, & x > t_{\max} \end{cases} \quad (15)$$

which provides coping of (-∞, ∞) into [-1, 1]. This modified compressor function inserts smaller distortion into output signal, in comparison to the one defined with (14), providing better objective quality and greater SQNR values [14]. In our analysis and design, we assume that PDFs of the source signal and the difference signal are Laplacian ones given by Eq. (5), so that Eq. (15) transforms into:

$$c_{\mathrm{mod}}(x) = \begin{cases} -1, & x < -t_{\max} \\ -1 + 2\dfrac{\exp\left\{\dfrac{\sqrt{2}}{3}t_{\max}\right\} - \exp\left\{-\dfrac{\sqrt{2}}{3}x\right\}}{\exp\left\{\dfrac{\sqrt{2}}{3}t_{\max}\right\} - \exp\left\{-\dfrac{\sqrt{2}}{3}t_{\max}\right\}}, & x \in [-t_{\max}, t_{\max}], \\ 1, & x > t_{\max} \end{cases} \quad (16)$$

As shown in Fig. 2, after compression, uniform quantization is performed so that the output of the compressor ranging [-1, 1] should be uniformly quantized. Accordingly, the decision thresholds and the representation levels of the uniform quantizer, denoted by Q_u in the block diagram of the compander shown in Fig. 2, can be specified by using the following equations:

$$t_{u,m} = -1 + \frac{2m}{N}, m = 0, 1, ..., N, \quad (17)$$

$$y_{u,m} = -1 + \frac{2m-1}{N}, m = 1, 2, ..., N, \quad (18)$$

where N denotes the number of representation levels of the optimal compander.

By using (17) and (18) the decision thresholds t_m and the representation levels y_m, of our optimal compander are then determined as the solutions of the following equations:

$$t_m = c^{-1}\left(t_{u,m}\right), m = 0,1,....,N\ , \tag{19}$$

$$y_m = c^{-1}\left(y_{u,m}\right), m = 1,2,....,N\ , \tag{20}$$

where obviously, due to the symmetry about zero in the quantizer characteristics, it holds that $t_m = t_{N}$-m, $m = 1, 2, \ldots, N/2 -1$, $y_m = y_{N-m+1}$, $m = 1, 2, \ldots, N/2$. This symmetry is an intuitionally expected result when the input has a PDF that is symmetrical about zero. The Laplacian PDF, we consider here, is indeed symmetrical about zero. Using (16)-(20) by direct computation yields:

$$t_m = \frac{3}{\sqrt{2}}\ln\left(\frac{N}{2N-2m-(2m-N)\exp\left\{-\frac{\sqrt{2}}{3}t_{\max}\right\}}\right), \tag{21}$$

$$y_m = \frac{3}{\sqrt{2}}\ln\left(\frac{N}{2N-2m+1+(2m-1-N)\exp\left\{-\frac{\sqrt{2}}{3}t_{\max}\right\}}\right), \tag{22}$$

where m is defined as $\frac{N}{2} \le m \le N$ while the support limit of the quantizer, $t_{\max}$, is defined as in [14]:

$$t_{\max} = \frac{3}{\sqrt{2}}\ln(N+1) \tag{23}$$

At this point, we determined the fixed representation levels and decision thresholds, which we can adapt to the standard deviation of the current difference signal frame and obtain the adaptive representation levels and decision thresholds as:

$$y_m{}^a = \hat{\sigma}^{(j)} y_m\ , \tag{24}$$

$$t_m{}^a = \hat{\sigma}^{(j)} t_m\ , \tag{25}$$

The final step in obtaining the quantized representative of the input signal is adding the quantized mean value of the input signal samples frame to the adaptive representation levels defined by (24).

In general, the shortage of the quantization process lies in the fact that we make a small error each time we map the input sample to a quantizer representation level [18]. These errors are summed up in an objective quality measure named signal distortion and it can be calculated as [18]:

$$D = \frac{1}{S}\sum_{n=1}^{S}(x_n - y_n)^2\ , \tag{26}$$

where $S = M \times L$ is the number of the input signal samples, while x_n and y_n represent the original and quantized input signal samples values, respectively. Thus defined distortion determines the main objective signal quality measure used in this paper, named signal to quantization noise ratio (SQNR). SQNR also depends on the variance of the input signal

$$\sigma^2 = \frac{1}{S}\sum_{n=1}^{S}(x_n)^2 \tag{27}$$

and it is calculated by applying the following equation [18]:

$$\mathrm{SQNR}\ [\mathrm{dB}] = 10\log_{10}\left(\frac{\sigma^2}{D}\right)\ . \tag{28}$$

IV. Experimental Results and Analysis

This section presents the results of the experiment we have conducted, where the modified BTC algorithm has been applied to the audio signal coding. The goal of this experiment is measuring the objective quality of the output audio signal, when the proposed modified BTC algorithm is applied. Moreover, the SQNR values calculated for the proposed modified BTC algorithm were compared to the performance of similar complexity audio signal coding schemes. Both the input and the output of the system are audio signals with the sampling frequency of 44.1 kHz. Input signal was originally encoded with 16 bits per sample, while the bit rate of the output signal varied from 7 to 9 bits per sample. The difference between these two bit rates represents the compression which is achieved by applying the modified BTC algorithm. Quality is measured through SQNR, determined by Eqs. (26)-(28). In this paper, we assume different frame sizes, which resulted in different SQNR and total bit rates used, as described in Section 2. The performance of the modified BTC algorithm is presented in the following Tables, along with the performance of the comparable systems for audio signal compression.

Table I presents the results of applying two variations of the modified BTC algorithm to the audio signals. The first variation, denoted as BTC3U, denotes the modified BTC algorithm with the application of three uniform quantizers. This is an initial system which applies multilevel uniform quantization, more similar to the original BTC algorithm, which utilizes three uniform quantizers [4]. The difference is in the implementation of an adaptive uniform quantizer for the difference signal quantization, with 256 representation levels, denoted by N_u, which implies using 8 bits per sample. Adaptive quantizer is utilized since the straightforward uniform quantization outputs significantly smaller SQNR values when applied to audio signals. Mean value of the input signal samples frame is quantized by uniform quantizer using 6 additional bits per frame. The standard deviation of the difference signal frame is quantized by uniform quantizer with 4 additional bits per sample. The support limit of all uniform quantizers designed ($x_{\max u}$) is defined as [20]:

$$x_{\max u} = \sqrt{2}\ln(N_u) - \sqrt{2}\ln\left(\frac{2}{3}\ln(N_u)\right). \tag{29}$$

TABLE I. SQNR FOR DIFFERENT FRAME SIZES AND BITRATES OBTAINED FOR CLASSICAL MUSIC SIGNAL AND ROCK MUSIC SIGNAL BY APPLYING DIFFERENT VARIATIONS OF THE BTC ALGORITHM

Frame size [samples]	R [bit/sample]	SQNR [dB] class. music		SQNR [dB] rock music	
		BTC3U	BTC1U	BTC3U	BTC1U
40	8.25	28.8709	34.307	34.886	38.5205
80	8.25	28.8573	34.145	34.875	37.8494
120	8.125	28.8355	33.9998	34.865	37.5655
160	8.083	28.8303	33.9024	34.855	37.2777
200	8.062	28.8173	33.8242	34.834	37.045
240	8.05	28.8221	33.7667	34.839	36.8659
280	8.042	28.8162	33.783	34.819	36.6875
320	8.036	28.8274	33.7814	34.789	36.5425

The same bit allocation is utilized in all implemented modifications of the BTC algorithm, so that, for the given fixed bit rates of the three abovementioned quantizers, the total bit rate depends on the frame size, which is the only variable left in Eq. (4).

BTC1U denotes the modified BTC algorithm where an adaptive uniform quantizer is used for processing the difference signal samples, while different quantizers are used for side information. The mean value of the frame of input samples is quantized by the μ law quantizer, while the standard deviation of the signal difference is quantized by the log-uniform quantizer. In other words, BTC1U presents the scheme the most similar to the one we propose, which is described in detail in Section 3. From Table I one can observe that, for the same total bit rate, BTC1U in comparison to BTC3U provides the gain in SQNR from around 5 dB to 5.4 dB for the considered classical music signal and from around 1.7 dB to 3.6 dB for the rock music signal, depending on the frame size. This gain is a direct result of applying more robust quantizers in quantization of the side information parameters. The input signals used in our experiment are rock music signal and classical music signal sampled at 44.1 kHz, coded by using 16 bits per sample. Table II presents important parameters of the input signals, as the signal variance, total samples number, and the amplitude dynamics in dB, depending on the frame size. Amplitude dynamics for every frame is calculated by:

$$Ad^{(j)} = 20\log_{10}\left(2\frac{x_{n\max}^{+(j)}}{x_{n\min}^{+(j)}}\right), \tag{30}$$

where $x_{n\max}^{+(j)}$ denotes the maximal positive input sample amplitude value of the jth frame, while $x_{n\min}^{+(j)}$ stands for the minimal positive input sample amplitude value of the jth frame. The value of amplitude dynamics for the complete signal sequence, shown in Table II, was calculated as a mean value of the amplitude dynamics calculated for all L frames of the input signal

$$Ad = \frac{1}{L}\sum_{j=1}^{L} Ad^{(j)} . \tag{31}$$

One can observe that the classical music signal is characterized by low variance and amplitude dynamics.

TABLE II. PARAMETERS OF THE INPUT AUDIO SIGNALS

Input signal	Signal variance	Number of samples	Frame size [samples]		
			160	220	320
			Amplitude dynamics [dB]		
Rock music	0.0361	2741355	93.2753	94.0288	93.9094
Classical music	0.0089	832531	56.1892	55.5290	53.8474

Table III shows the results of applying the proposed modified BTC algorithm to an audio signal and its comparison to forward adaptive optimal compander (FAOC) [13], and logarithmic PCM system (LOGPCM) [12] for very close bit rate values. The SQNR values are paired with the total bit rates used, for achieving a fair comparison. The bit rate of the FAOC is calculated also by using Eq. (4), where r_1=0. Bit rate of the LOGPCM is constant, since it does not depend on the frame size and the technique does not use side information in input signal coding. From Table III, we can observe that the proposed modified BTC algorithm provides a significant gain in SQNR, in comparison to the LOGPCM for every frame size, so as when compared to FAOC, for using smaller frame sizes. The proposed modified BTC algorithm provides the gain from around 8.1 dB to 5.9 dB in comparison to the LOGPCM. When compared to FAOC, the proposed modified BTC algorithm provides gain from about 1.9 dB to 0.5 dB for smaller frame sizes ($M \leq 120$), while the gain is descending, when we use larger frames. This interesting observation can be explained by the characteristics of the input signal used, since it has overall mean value very close to zero. When we use larger frame sizes, the frame mean of this particular signal is actually closer to zero, than to its quantized value with the application of the 64 levels log-uniform quantization. If we focus on the smaller frame sizes up to 120 samples, we can observe that the gain is varying from around 8.1 dB to 6.6 dB when compared to LOGPCM, while it amounts from around 1.9 dB to 0.5 dB, when compared to the FAOC. We can therefore conclude that the modified BTC algorithm is suitable for coding of a classical music signal, especially when smaller frame sizes are utilized.

TABLE III. SQNR FOR DIFFERENT FRAME SIZES AND BITRATES OBTAINED FOR CLASSICAL MUSIC SIGNAL BY APPLYING DIFFERENT CODING SYSTEMS

Frame size [samples]	$SQNR_{BTC}$ [dB], R_{BTC} [bit/sample]	$SQNR_{FAOC}$[dB], R_{FAOC} [bit/sample]	$SQNR_{LOGPCM}$ [dB], R_{LOGPCM} [bit/sample]
40	43.7468, 8.25	41.8704, 8.1	35.5672, 8
80	42.8037, 8.125	41.8118, 8.05	35.5672, 8
120	42.223, 8.0833	41.721, 8.0333	35.5672, 8
160	41.926, 8.0625	41.6449, 8.025	35.5672, 8
200	41.6867, 8.05	41.5733, 8.02	35.5672, 8
240	41.553, 8.0417	41.5297, 8.0167	35.5672, 8
280	41.524, 8.036	41.5258, 8.0143	35.5672, 8
320	41.504, 8.031	41.5329, 8.0125	35.5672, 8

Table IV presents the performance of applying the proposed modified BTC algorithm when a rock music signal with higher amplitude dynamics and variance. Additionally, Table IV shows SQNR values, obtained for the same input signal, by applying the FAOC and LOGPCM, paired with the bit rates used. One can observe that the proposed modified BTC algorithm provides a significant gain in SQNR in comparison to the LOGPCM, and it amounts from around 8.6 dB when we use a frame size of 40 samples, to about 6.1 dB in the case of 320 samples per frame. When compared to FAOC, the modified BTC algorithm provides a maximum gain of 3.9 dB, for the fame size of 40 samples, while the gain equals to about 0.5 dB, when the frame size is equal to 320 samples.

TABLE IV. SQNR FOR DIFFERENT FRAME SIZES AND BITRATES OBTAINED FOR ROCK MUSIC SIGNAL BY APPLYING DIFFERENT CODING SYSTEMS

Frame size [samples]	$SQNR_{BTC}$ [dB], R_{BTC} [bit/sample]	$SQNR_{FAOC}$ [dB], R_{FAOC} [bit/sample]	$SQNR_{LOGPCM}$ [dB] R_{LOGPCM} [bit/sample]
40	45.4518, 8.25	41.4788, 8.1	36.7840,8
80	44.486, 8.125	41.8003, 8.05	36.7840,8
120	44.098, 8.0833	42.2055, 8.0333	36.7840,8
160	43.7618, 8.0625	42.3998, 8.025	36.7840,8

200	43.5035, 8.05	42.5243, 8.02	36.7840,8
240	43.3065, 8.0417	42.5327, 8.0167	36.7840,8
280	43.1178, 8.0357	42.5285, 8.0143	36.7840,8
320	42.9574, 8.0313	42.4897, 8.0125	36.7840,8

The main difference between the proposed modified BTC algorithm and FAOC is in using the information about the mean value of the frame. In FAOC we do not calculate or quantize the mean, which leave us with less information about the input signal. By calculating the mean and creating the difference signal (1), we narrow the input signal amplitude dynamics, which makes it more suitable for processing. This has a significant impact, especially for usage of smaller frame sizes, where the mean of the frame more correctly describes the particular samples, than in the case of larger frames

Fig. 3 and Fig. 4 are useful for the analysis of utilizing the mean value of the frame and its quantized representative in the modified BTC algorithm. For the purpose of this analysis, we used a section of the rock music signal used for performance analysis of the modified BTC algorithm, shown in Table IV. This section consists of 10240 samples, producing 64 frames at frame size of 160 samples. Fig. 3.a) shows the waveform of the amplitudes of the considered section of the rock music signal, while Fig. 3.b) shows SQNR obtained by applying FAOC. On the other hand, Fig. 4.a) shows the waveform of the amplitudes of the difference signal, while Fig. 4.b) shows SQNR obtained when the modified BTC algorithm is applied. In both figures, SQNR values are calculated and presented for individual frames of the considered input signal sequence. In the calculation, Eqs. (26)-(28) are applied where instead of S we use M, since S denotes the number of the input signal samples, while M denotes the frame length. One can observe that the difference signal has smaller amplitude dynamics, i.e. narrower range of amplitude values, which makes it more suitable for quantization. At this point, the performance of the proposed modified BTC algorithm relies on the mean quantization process. As already described, the mean of the frame of input signal samples is quantized by robust μ law quasilogarithmic quantizer, using 6 additional bits per frame for its representation. By comparing the Fig. 3.b) and Fig. 4.b) we can clearly see the gain in SQNR, which is result of utilizing the mean value of the frame of input signal samples in the modified BTC algorithm. One can also notice the difference in SQNR from frame to frame, which is result of the input signal nature, which is characterized by high amplitude dynamics (see Table II). This analysis confirms that quantization of the mean value of the frame improves performance of this algorithm. Additionally, it is shown that using more robust quantizers provide significant gain in SQNR.

Performance of the applied compression algorithm can also be presented through SQNR characteristics. Fig. 5 shows SQNR characteristics of the proposed modified BTC algorithm, along with the systems compared. One can easily observe the gain in SQNR in favor of the proposed modified BTC algorithm, for different bit rates. These characteristics are obtained by using Eqs. (26)-(28) for the rock music signal, with the frame size of 160 samples and for the bit rate equal to 6 bit/sample, 7 bit/sample, 8 bit/sample and 9 bit/sample. We can clearly see the gain in SQNR, when compared both to LOGPCM and to FAOC. In addition, we can observe the following. By comparing the SQNR of the LOGPCM in the case of using 8 bits per sample, with the SQNR of the proposed modified BTC algorithm when using LOGPCM in the case of using 8 bits per sample, with the SQNR of the proposed modified BTC algorithm when using 7 bits per sample, we can see that proposed modified BTC system outputs higher SQNR. This means that we achieve compression of more than 1 bit per sample in comparison to LOGPCM, while preserving the signal quality. This makes the proposed modified BTC algorithm suitable for application in audio signal coding and compression.

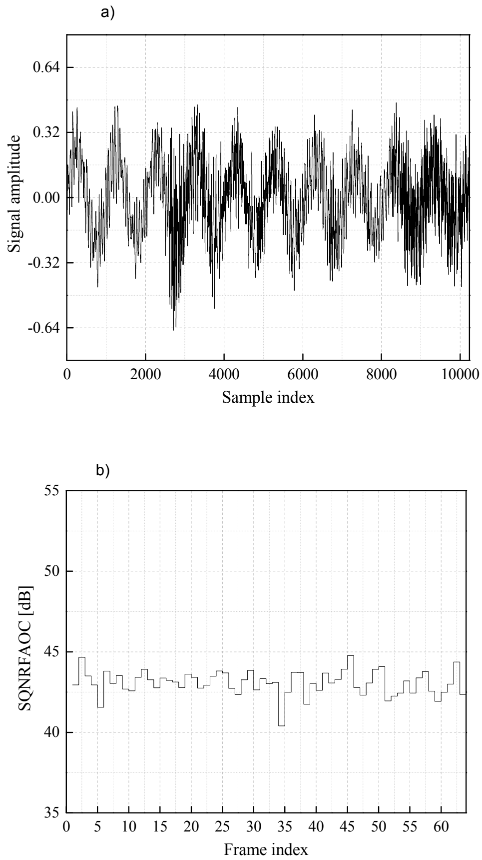
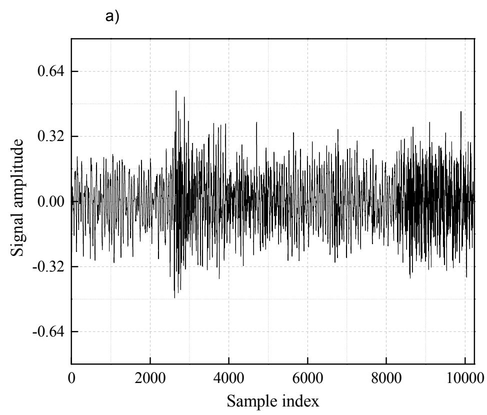


Figure 3. a) Waveform of the amplitudes of a section of the rock music signal, b) SQNR values per frames of FAOC for 64 individual frames of size $M = 160$ samples



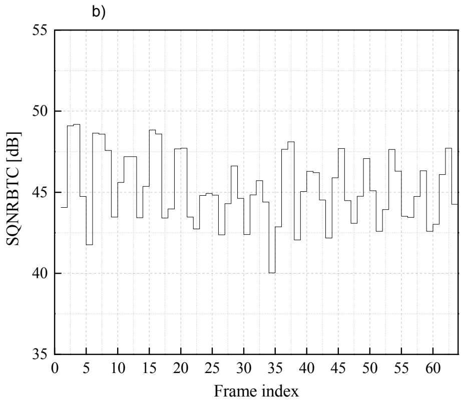


Figure 4. a) Waveform of the amplitudes of a difference signal b) SQNR values per frames of the modified BTC algorithm for 64 individual frames of size M = 160 samples

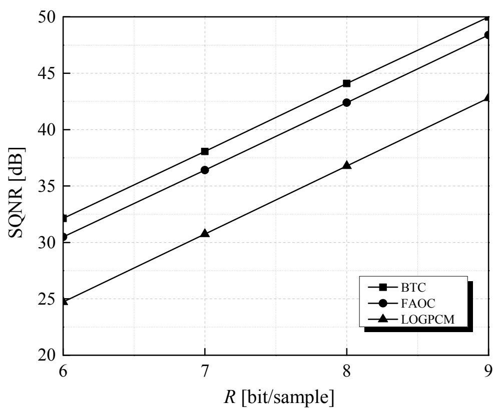


Figure 5. SQNR characteristics of the proposed modified BTC algorithm compared to FAOC and LOGPCM

V. Conclusion

This paper has presented the modified BTC algorithm and its possible application in audio signal coding. It has been shown that the idea of the original BTC algorithm can be successfully applied in audio signal coding. This is possible with some modifications that are described in this paper. Quality of the output has been determined by observing the SQNR values, while using different bit rates, frame sizes and input signals. Moreover, SQNR has been obtained for logarithmic PCM, forward adaptive optimal compander and BTC algorithm with the implementation of multilevel uniform quantization, for comparison purposes. Numerical results have shown that smaller frame sizes provide higher SQNR due to smaller error in estimating the input signal parameters. Also, the comparative analysis of the results has shown that the proposed algorithm provides a significant gain in SQNR and achieves notable signal compression. The proposed modified BTC algorithm provides the gain in SQNR up to 8.6 dB when compared to LOGPCM, while it ranges up to 3.7 dB, when compared to FAOC, for smaller frame sizes. Additionally, we provided the SQNR characteristics for the proposed algorithm, along with the systems compared, which shows obvious and constant gain in SQNR for bit rate values of 6, 7, 8 and 9 bits per sample. The modified BTC algorithm, proposed in this paper is characterized by using simple quantization methods with low computational complexity. This leaves the possibility of upgrading the system, by applying more complex quantization and adaptation techniques, which is left to future research. Also, with small modifications, it is possible to apply this algorithm in the coding of speech or electrocardiography (ECG) signal. Finally, we can conclude that the proposed modified BTC algorithm can be successfully applied to audio signal coding.

Acknowledgment

This work is supported by Serbian Ministry of Education, Science and Technologic Development (Project TR 32035).

References

[1] W. C. Chu, Speech Coding Algorithms: Foundation and Evolution of Standardized Coders. John Wiley & Sons, New Jersey, Chapters 5-6, pp. 143-183, 2003.

[2] L. Hanzo, C. Somerville, J. Woodard, Voice and Audio Compression for Wireless Communications. London, John Wiley & Sons, Chapter 1, pp. 1-28, 2007.

[3] A. Kondoz, Digital Speech: Coding for Low Bit Rate Communication Systems. John Wiley & Sons, Ch. 2–3, pp. 5–55, 2004.

[4] E. J. Delp, O. R. Mithcell, “Image compression using block bruncation coding”, IEEE Trans. Commun., vol. 27 no. 9, pp. 1335-1342, 1979. doi: 10.1109/tcom.1979.1094560

[5] Compact Disc Digital Audio System, (IEC/ANSI) CEI-IEC-908, 1987.

[6] P. Ted, A. Spanias, “Perceptual coding of digital audio” Proceedings of the IEEE, vol. 88 no. 4, pp. 451-515, 2000. doi: 10.1109/5.842996

[7] K. Sayood, Introduction to data compression, Newnes, Chapter 1, pp. 3-38, 2012.

[8] M. Bosi, R. Goldberg. Introduction to digital audio coding and standards. Vol. 721. Springer Science & Business Media, Chapter 11-15, pp. 264-425, 2012.

[9] Z. Perić, S. Tomić, M. Tančić, “High quality speech signal coding with the application of BTC algorithm”, International Conference on Advanced Technologies, Systems and Services in Telecommunications - TELSIKS, Nis, Serbia, October 2015, pp. 23-26

[10] V. Despotovic, Z. Peric, L. Velimirovic, V. Delic, "DPCM with forward gain-adaptive quantizer and simple switched predictor for high quality speech signals," Advances in Electrical and Computer Engineering, vol.10, no.4, pp.95-98, 2010. doi:10.4316/AECE.2010.04015

[11] Q. Guoping, "Color image indexing using BTC", IEEE Transactions on Image Processing, vol. 12, no. 1 pp. 93-101, 2003.

[12] ITU-T, Recommendation G.711, “Pulse Code Modulation (PCM) of Voice Frequencies”, International Telecommunication Union, 1972.

[13] J. Nikolic, Z. Perić, “Lloyd-Max’s algorithm implementation in speech coding algorithm based on forward adaptive technique”, Informatica, vol. 19, no. 2, pp. 255–270, 2008.

[14] Z. Perić, M. Petković, M. Dinčić, “Simple compression algorithm for memoryless Laplacian source based on the optimal companding technique”, Informatica, vol. 20, no. 1, pp. 99-114, 2009.

[15] N. Judell, L. Scharf, “A simple derivation of Lloyd's classical result for the optimum scalar quantizer”, IEEE Trans. Inform. Theory, vol. 32, no. 2, pp. 326-328, 1986. doi: 10.1109/tit.1986.1057150

[16] S. Jiang, R. Yin, P. Liu, “Finite-state entropy-constrained vector quantiser for audio modified discrete cosine transform coefficients uniform quantisation“, IET Signal Processing, vol. 9 , no. 1, pp. 30-36, 2015. doi: 10.1049/iet-spr.2014.0089

[17] R. Farias, J. Brossier, “Adaptive quantizers for estimation”, Signal Processing, vol. 93, no. 11, pp. 3076-3087, 2013. doi: 10.1016/j.sigpro.2013.04.012

[18] N. S. Jayant, P. Noll, Digital coding of waveforms. New Jersey, Prentice Hall, Chapter 5, pp. 221-251, 1984.

[19] A. Spanias, T. Painter, V. Atti, Audio signal processing and coding, John Wiley & Sons, 2006.

[20] D. Hui, D. Neuhoff, “Asymptotic analysis of optimal fixed-rate uniform scalar quantization”, IEEE Trans. Inform. Theory, vol. 47, no. 3, pp. 957-977, 2001. doi: 10.1109/18.915652