DOI: 10.5862/JCSTCS.229.9
UDC 004.627

E.M. Khassina, A.A. Lomov

AUDIO FILES COMPRESSION WITH THE STLS-ESM METHOD*

Е.М. Хасина, А.А. Ломов

СЖАТИЕ АУДИОФАЙЛОВ МЕТОДОМ STLS-ESM

In the paper an audio compression algorithm based on modeling an audio signal by a partial solution of a certain difference equation in the time domain is investigated. The signal is modeled as a sum of exponentially damped sinusoids. Such an approach is thought to be efficient in modeling the transient segments that are present in speech audio signals and audio signals generated by conventional musical instruments. In order to approximate an audio signal frame with a solution of a difference equation, a variational (STLS) problem is solved using the inverse iteration algorithm with an updating inverse matrix. The α-version of the audio codec based on the STLS-ESM scheme was created and tested in comparison with LAME MP3 codec.

AUDIO SIGNALS MODELING; AUDIO CODEC; EXPONENTIAL SINUSOIDAL MODEL; PARAMETRIC IDENTIFICATION; DIFFERENCE EQUATIONS; VARIATIONAL IDENTIFICATION METHOD; STRUCTURED TOTAL LEAST SQUARES.

Рассмотрен алгоритм сжатия аудиосигнала путем его аппроксимации во временной области частным решением некоторого разностного уравнения. Аудиосигнал представлен в виде суммы экспоненциально затухающих/растущих синусоид. Такой подход эффективен для моделирования переходных процессов в речевых сигналах и сигналах традиционных музыкальных инструментов. Для построения аппроксимации решена вариационная задача идентификации с использованием итераций с обновляемой обратной матрицей. Представлена α-версия аудиокодека и приведены результаты тестирования в сравнении с аудиокодеком LAME MP3.

МОДЕЛИРОВАНИЕ АУДИОСИГНАЛОВ; АУДИОКОДЕК; EXPONENTIAL SINUSOIDAL MODEL; ПАРАМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ; РАЗНОСТНЫЕ УРАВНЕНИЯ; МЕТОД ВАРИАЦИОННОЙ ИДЕНТИФИКАЦИИ.

1. INTRODUCTION

Lossless audio codecs, e. g. FLAC, Monkey's Audio APE, TTA can achieve compression ratios of 2-3 times.

Lossy audio codecs such as MPEG-1 codecs (MP1, MP2 and MP3) and OGG Vorbis reduce the sound quality of music files, without, however, degrading it radically. Hence, for the human ear, the distortions made by such codecs are not actually noticeable. A music file compressed to a fifth of its original size still has the sound quality of radio broadcasting.

In their work, lossy codecs tend to decompose an audio signal into harmonics. However, the method often does not correspond to the physical nature of sounds produced by conventional musical instruments, for which the presence of a considerable quantity of transients (high-amplitude, short-duration sound segments followed by an exponential decay) is common. If a piece of a signal contains transients it cannot be considered as a quasi-stationary episode. This is why the MP3 audio codec, for instance, has to use the

* This article is an extended post-conference revision of the paper «Audio Files Compression with the Variational Method of Identification of Modeling Difference Equations» by Eugenia M. Khassina and Andrei A. Lomov published in Pyshkin, E., and Klyuev, V. (Eds). Proceedings of the International Workshop on Applications in Information Technology (IWAIT-2015), The University of Aizu Press, 2015. Available at <http://kspt.ftk.spbstu.ru/media/files/2015/iwait-2015/proceedings/iwait-2015-e-proceedings-release.pdf>

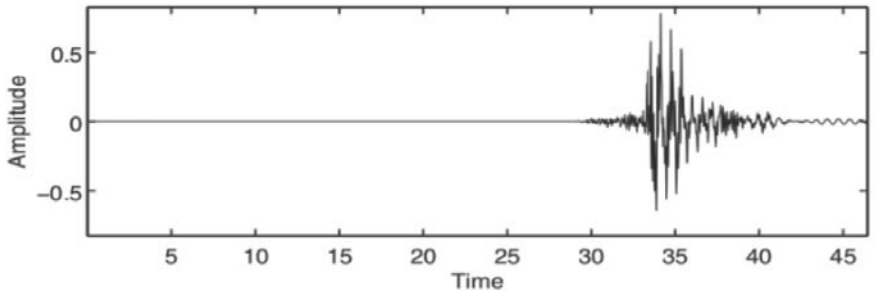

Fig. 1. Transient [1]

Modified Discrete Cosine Transform (MDCT) with windows of varied length while processing a signal. A long window (1024-subband MDCT) is used for quasi-stationary episodes of a signal while a short window (128-subband MDCT) is used for episodes with transient attacks. Providing a better time resolution, the short-block mode of the encoding scheme also helps to avoid what is commonly referred to as a pre-echo artifact: frequently annoying audible quantization noise that occurs before a transient [1].

In Fig. 1 a transient can be seen. Even though it is more natural to decompose such signals into a sum of exponentially damped sinusoids, rather than to represent them as a Fourier series, such an approach requires a considerable amount of CPU resources. This is the reason why the creation and the usage of codecs based on the principle have only recently become justified.

In formula (1) below, audio signal $s(n)$ is represented as a superposition of slowly time-varying exponentially weighted sinusoids and quasi-stationary noise $\eta(n)$. The signal model is called the Exponential Sinusoidal Model (ESM). The frequencies ω_i, phases φ_i, amplitudes α_i and damping parameters γ_i can be obtained without switching to the frequency domain.

$$s(n) \approx \sum_{i=1}^{K} a_i(n) e^{-\gamma_i(n)n} \sin(\omega_i(n)n + \phi_i(n)) + \eta(n). \quad (1)$$

In Ref. [2] a Total Least Squares (TLS) problem of order $2K$ is solved to estimate ω_i and γ_i, where K is the number of available exponentially damped sinusoids predefined by a user [3, 4]. On the basis of the TLS-ESM scheme an experimental audio codec was created and tested [2].

TLS searches for the parameters of the damped sinusoids maximizing the signal-to-noise ratio (SNR), that is, the algorithm tries to make the original and the modeled signals as close to each other as possible in the time domain on each set of $2K$ samples. TLS considers the sets independent even if they contain common samples, and if the sets are dependent in reality. The problem is an essential disadvantage of TLS.

It also should be noted that the SNR quality measure is a formal criterion. It effectively allows to achieve a high degree of time resemblance between the original signal and the modeled one on separate sets of samples while ignoring the perceptual quality of the modeled audio that is much more important for an audio codec.

Experiments showed [2] that TLS spends all the modeling sinusoids for low frequencies with high amplitudes. An explanation for this fact is that adding a high-amplitude sinusoid to the modeled signal reduces the difference between the original and the modeled signals more than adding a high-frequency low-amplitude sinusoid does. As a result, there are few available sinusoids left for modeling the high frequencies and this is quite noticeable for the human ear.

Thus, the initial fullband TLS algorithm had to be improved with additional signal processing in the frequency domain. The modified version of the algorithm is called Subband TLS-ESM [2]. The original signal is decomposed into 32 frequency subbands with a fully decimated uniform QMF analysis filter bank (32 channels) used in the MPEG-1 Layer I codec. For each of the subband signals an independent TLS problem is solved with the same number of sinusoids employed for the modeling.

The goal of the work is to research an audio compression algorithm which decomposes a signal into exponentially damped sinusoids in the time domain without switching to the frequency domain. We use an approximation of an audio signal on the whole audio frame using the variational identification method [4–6], close to the Structured Total Least Squares (STLS) [7] and the Global Total Least Squares (GTLS) [8] methods.

In [9] a vocoder based on the ESM-STLS scheme was proposed. The testing of the vocoder was performed in comparison to Code-excited linear prediction (CELP), a standard speech coding algorithm. The results of the testing showed that, providing a similar compression ratio, the new vocoder has a substantially higher signal-to-noise ratio (SNR). However, the speech spectrum is rather simple, which makes speech signals easily compressible. Our experiments showed that Newton's iterative algorithms, to which STLS1 and STLS2 used in [9] belong, have bad convergence when solving the STLS problem for music audio files with wider spectra.

2. THEORETICAL ASPECTS

Coding an audio frame. Our algorithm divides the whole signal of an audio file into frames of N samples each, processing the frames one by one. In section 3 we will consider how N value and other parameters are chosen. Conventionally, N equals 100. A frame of samples is denoted by $s[k], k = \overline{1, N}$.

We will treat the vector $s \doteq (s[1];\cdots;\ s[N])$ as a perturbed observation of a solution process $z \doteq (z[1];\cdots;\ z[N])$ of a certain homogeneous linear difference equation with real coefficients. We will solve the inverse problem of identifying the unknown coefficients of the equation. Let us take as an example a difference equation of order $p = 3$:

$$\begin{gathered} z[k+3] + \alpha_2 z[k+2] + \\ + \alpha_1 z[k+1] + \alpha_0 z[k] = 0, \quad k = \overline{1, N-3}. \end{gathered} \tag{2}$$

Let us denote the characteristic roots of the system (2) by ξ_i. The characteristic polynomial of the system is real, hence all the roots that are complex should occur in complex-conjugate pairs. For the present example, let us suppose that the single real root of the characteristic polynomial is ξ_2. Then we have the general solution of (2) as follows:

$$z[k] = C_0 \xi_1^k + C_1 \overline{\xi_1}^k + C_2 \xi_2^k . \tag{3}$$

We are interested only in real solutions of the difference equation, as audio samples of observation s are real. Therefore, we can switch to the real form of the general solution (3), assuming C_0 and C_1 to be complex conjugates, and convert it to a sum of exponentials and exponentially damped sinusoids:

$$\begin{gathered} z[k] = C_1 \xi_1^k + \overline{C_1 \xi_1}^k + C_2 \xi_2^k = \\ = C_1 e^{(\gamma_1 + i\omega_1)k} + \overline{C_1} e^{(\gamma_1 - i\omega_1)k} + C_2 e^{\gamma_2 k} = \\ = A_1 \rho_1^k \cos(k\omega_1) + A_2 \rho_1^k \sin(k\omega_1) + \\ + A_3 \rho_2^k, \ \forall i \ A_i \in \Re. \end{gathered} \tag{4}$$

We introduce the following notation for the vector of the coefficients of the difference equation:

$$\gamma \doteq (\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad 1)^\top .$$

Let us define the objective function for the identification of vector γ:

$$\begin{gathered} J(\gamma) = \|s - z(\gamma)\|^2 , \\ z(\gamma) \doteq \arg\min_{z:\ (2)} \|s - z\|^2 . \end{gathered} \tag{5}$$

This variational problem was first formulated and solved by A.O. Egorshin [4, 5]. For its numerical solution we apply the iterative algorithm with an updating inverse matrix proposed by A.O. Egorshin and, independently, by M.R. Osborne [10]. We use the least-squares estimate γ_{LS} [11, 12] as the initial γ for the iterative algorithm w. Now we define special matrices in order to derive the formula for γ_{LS} and to describe the iterative algorithm solving problem (5).

First, let us transform the difference equation (2) to a matrix form:

$$\underbrace{\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 1 & & 0 \\ 0 & \alpha_0 & \alpha_1 & \alpha_2 & 1 & \\ \vdots & & \ddots & \ddots & \ddots & \ddots \\ 0 & & & \alpha_0 & \alpha_1 & \alpha_2 & 1 \end{pmatrix}}_{G} \underbrace{\begin{pmatrix} z[1] \\ z[2] \\ \vdots \\ z[N] \end{pmatrix}}_{z} = 0. \tag{6}$$

We search for vector γ by adjusting the residual $e_\gamma \doteq G_\gamma s$ to approach zero. Now the objective function for the least-squares problem is as follows:

$$J_\gamma = \arg\min_\gamma e_\gamma^{\top} e_\gamma \ , \quad e_\gamma = G_\gamma s. \tag{7}$$

Then we use the identity $G_\gamma s \equiv V(s)\gamma$, where V is a Hankel matrix:

$$\underbrace{\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 1 & & 0 \\ 0 & \alpha_0 & \alpha_1 & \alpha_2 & 1 & \\ \vdots & & \ddots & \ddots & \ddots & \ddots \\ 0 & & & \alpha_0 & \alpha_1 & \alpha_2 \quad 1 \end{pmatrix}}_{G_\gamma} \underbrace{\begin{pmatrix} s[1] \\ s[2] \\ \vdots \\ s[N] \end{pmatrix}}_{s} \equiv \overbrace{\left(\begin{array}{ccc|c} s[1] & s[2] & s[3] & s[4] \\ s[2] & s[3] & s[4] & s[5] \\ \vdots & \vdots & \vdots & \vdots \\ s[N-3] & s[N-2] & s[N-1] & s[N] \end{array}\right)}^{V} \underbrace{\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ 1 \end{pmatrix}}_{\gamma}. \tag{8}$$

V_1 V_2

We denote the vector of the identified coefficients by θ and rewrite the expression for the residual e_γ:

$$\theta = (\alpha_0 \quad \alpha_1 \quad \alpha_2)^{\top}, \ \gamma = \begin{pmatrix} \theta \\ 1 \end{pmatrix}, \qquad e_\gamma = V(s) \cdot \gamma = V_1(s) \cdot \theta + V_2(s). \tag{9}$$

Solving the linear least-squares problem we get the values of the coefficients θ:

$$\theta_{\mathrm{LS}} = -(V_1^{\top} V_1)^{-1} V_1^{\top} V_2. \tag{10}$$

Now, the iterations with an updating inverse matrix which solve the variational identification problem (5) are:

1. The initial value: $\gamma = \gamma(0) = \gamma_{\mathrm{LS}}$.
2. For $k \geq 0$

$$\begin{cases} \tau = (V(s)^{\top} (G_{\gamma(k)} G_{\gamma(k)}^{\top})^{-1} V(s))^{-1} \cdot \gamma(k), \\ \gamma(k+1) = \dfrac{1}{(0...01)\tau} \tau. \end{cases} \tag{11}$$

The last equation means the division of the whole auxiliary vector θ by its last element in order to make the last element of vector $\gamma(k+1)$ equal to unity. The main difference of the Egorshin–Osborne iterations from the computational TLS algorithm consists in the presence of the inverse matrix $(G_{\gamma(k)} G_{\gamma(k)}^{\top})^{-1}$ which is updated at each iteration.

Using the calculated estimate for γ, we find the modeling process $z(\gamma)$ nearest to the observation s as the linear projection [4, 5]:

$$z(\gamma) = \underbrace{(I - G_\gamma^{\top} (G_\gamma G_\gamma^{\top})^{-1} G_\gamma)}_{I - \Pi} \cdot s, \tag{12}$$

where I is an identity matrix.

Matrix $I - \Pi$ is a projection matrix to the subspace of solutions of the equation $G_\gamma z = 0$. The obtained coefficient vector γ and the corresponding process $z(\gamma)$ that fit the observation s are used further, as we will show below.

Note also that in this section the order p of the difference equation is considered known. A way of choosing p and the problems encountered when using the iterations (11) at the implementation stage will be conveyed in the next section.

Decoding an audio frame. From (4) we can see that, in order to restore process z, for each complex-conjugate pair of roots of the characteristic polynomial of the difference equation we need to know the real argument and the real modulus of the polar form of that root of the pair that lies above the real axis and we also need to know the set of real coefficients Ai. Thus, we should keep $2p$ float numbers to restore one audio frame. Besides, we should also keep one byte (or two/three for the last frame of an audio file) of service information for each audio frame such as the order of a difference equation and the frame size type. We will not describe in detail the structure of the frame service information byte in this paper.

The only thing left for us to understand is how to get coefficients A_i, $i = \overline{1, p}$. Let us transform the expression (4) to a matrix form:

$$\begin{pmatrix} z[1] \\ \vdots \\ z[N] \end{pmatrix} = \underbrace{\begin{pmatrix} \rho_1 \cos(\omega_1) & \rho_1 \sin(\omega_1) & \rho_2 \\ \rho_1^2 \cos(2\omega_1) & \rho_1^2 \sin(2\omega_1) & \rho_2^2 \\ \vdots & \vdots & \vdots \\ \rho_1^N \cos(N\omega_1) & \rho_1^N \sin(N\omega_1) & \rho_2^N \end{pmatrix}}_{H} \times \tag{13}$$

$$\times \underbrace{\begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix}}_{d} = Hd. \qquad (13)$$

Using the iterations (11) we have found the model G and the process z, corresponding to the original observation s, such that $G\,z = 0$. Knowing G (the coefficients of the difference equation), we can find the characteristic roots of the equation and, thus, the matrix H. Note that the next expression is true:

$$G\,z = GHd = 0,\ d \neq 0 \Rightarrow G \perp H.$$

The needed vector d containing coefficients A_i can be found with the least squares method:

$$d = (\bar{H}^\top \bar{H})^{-1} \bar{H}^\top \bar{z}, \qquad (14)$$

where $\bar{H}$ is a submatrix composed of $\geq p$ rows of matrix H and $\bar{z}$ is a subvector composed of the corresponding elements of vector z. Also the following equations take place:

$$d = (H^\top H)^{-1} H^\top z =$$
$$= (H^\top H)^{-1} H^\top H (H^\top H)^{-1} H^\top s =$$
$$= (H^\top H)^{-1} H^\top s.$$

3. CODEC IMPLEMENTATION

We have implemented an audio codec (consisting of the coder and decoder modules), based on the theory described above, in the Scilab cross-platform environment (http://www.scilab.org) supplying a high-level programming language resembling MATLAB, the language interpreter and an ample set of mathematical functions. The codec implementation and testing were performed in the Debian GNU/Linux operating system. As input the codec accepts a mono WAV audio file with a sample rate of 44.1 KHz and a bit depth of 16 bits. As output the codec produces a compressed file, whose special structure we had to define with a new extension .cod. In Fig. 2 you can see the general scheme of the codec operation.

An original audio signal to be compressed is divided into frames of N samples with a shift of $(N - M)$ samples. That is to say, each pair of consequent frames overlap by M samples to be glued smoothly after their decompression. After the decoding procedure two neighboring frames are summed on the gluing area preliminarily weighted. The weighting coefficients vary from 0 to 1. The weighting functions we use are $\sin^2(ck)$ and $1 - \sin^2(ck)$ where the time index k runs from 0 to $M - 1$ and $c(M - 1) = \pi/2$. In Fig. 3 you can see the gluing scheme.

The values of N and M can be predefined by a user. It was discovered that for frames longer than 150 samples the coding procedure often fails because the iterations (11) do not converge, and if we take M value < 5, an audible noise appears on joints of neighboring frames. We chose the average values: $N = 100$ and $M = 10$.

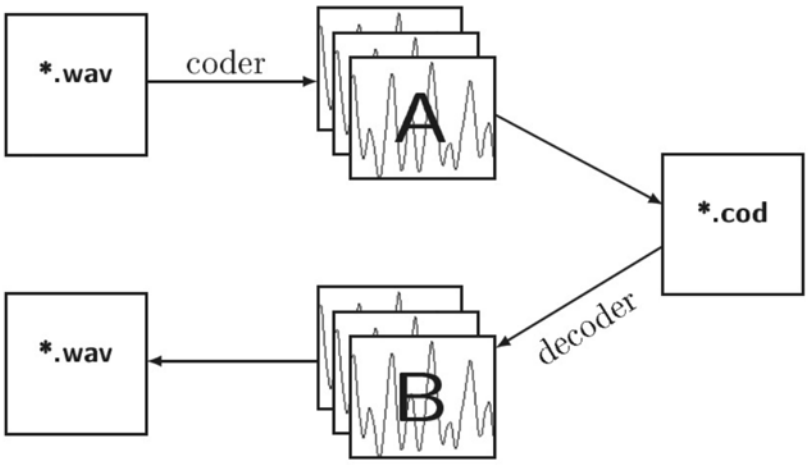

Fig. 2. General scheme of the codec operation:
A. Original frames. B. Decompressed frames

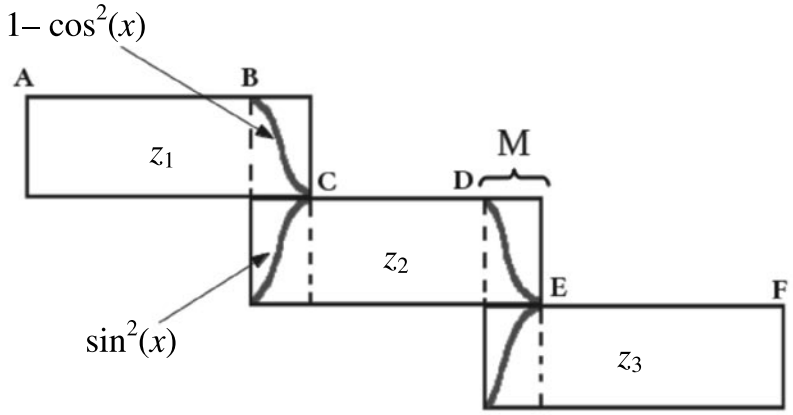

Fig. 3. Gluing of frames after decompression

As an example, let us have an initial audio signal s consisting of 280 samples. We will model it with three frames that consist of $N = 100$ samples each and overlap by $M = 10$ samples.

Let us denote the first frame we are to approximate by $s_1 = (s[1]; \cdots ; s[100])$. We find $2p$ float numbers needed to restore s_1. We denote the restored version of the frame by z_1 and we will call its endings A and C as it is shown in Fig. 3.

Now we take the following frame $s_2 = (s[91]; \cdots ; s[100]; \cdots ; s[190])$ and model it. The result of the frame modeling is z_2 whose endings are B and E.

The two modeled frames overlap in the area [B, C]. We multiply the overlapping part $(z_1[91]; \cdots ; z_1[100])$ of z_1 by a decreasing square sinusoid

$$\left(\sin^2\left(\frac{\pi}{2}\right); \sin^2\left(\frac{\pi}{2}\cdot\frac{M-2}{M-1}\right); \cdots ; \sin^2\left(\frac{\pi}{2}\cdot\frac{1}{M-1}\right); \sin^2(0)\right)$$

component-wise. The dimensions of the multiplied vectors are both equal to M. At the same time we multiply the overlapping part $(z_2[1]; \cdots ; z_2[10])$ of z_2 by the vector of an increasing square sinusoid

$$\left(\sin^2(0); \sin^2\left(\frac{\pi}{2}\cdot\frac{1}{M-1}\right); \cdots ; \sin^2\left(\frac{\pi}{2}\cdot\frac{M-2}{M-1}\right); \sin^2\left(\frac{\pi}{2}\right)\right).$$

The dimensions of the vectors multiplied are equal again.

The result of gluing z_1 and z_2 is a consequence of the samples of [A, B] part of z_1, the samples of a component-wise sum of the overlapping parts [B, C] of z_1 and z_2 and the samples of [C, E] part of z_2. The resulting vector [A, E] consists of 190 samples.

Now we model the last frame s_3. The result of the modeling is frame z_3 with the endings D and F. We glue the resulting vector [A, E] with z_3 in the same way. Now the overlapping area is [D, E].

The described approach was implemented in our codec and all the tests showed that the gluing is performed smoothly. However, we should note that the overlaps decrease the resulting compression ratio because the quantity of samples that we can code using n frames declines from $n \cdot N$ to $n \cdot (N - M) + M$ when the frames overlap.

When coding a frame we increment a model order p in a cycle from $p_{\min} = 2$ to $p_{\max} = 13$. For each p the identification of coefficients of the equation (2) is performed and the model process (12) is found. After this, the relative modeling error is counted as follows:

$$\frac{\|z - s\|}{\|s\|} \leqslant 5\ \%\ , \qquad (15)$$

where the value 5 % is the relative error threshold. If the relative error counted does not exceed the threshold then we break the p cycle and write the found $2p$ float numbers and a service information byte to the output

Testing results

Files type	Files number	Our codec		LAME MP3 128 kbps		LAME MP3 256 kbps	
		compression ratio	relative error	compression ratio	relative error	compression ratio	relative error
Piano	20	3.334	0.028	5.15	0.052	2.575	0.001
Electric guitar	20	1.451	0.028	5.15	0.056	2.575	0.003

.cod file. Obviously, the least suitable model order is preferable to make the compressed file as small as possible. If the coder fails for any order p with the chosen relative precision 5 % then it divides the frame in half and tries to model each of the two smaller frames again. If the modeling process for a smaller frame is not successful anyway, then the frame is written to .cod file directly without compression. The relative error threshold is predefined by a user and, in general, can be set to an arbitrarily small number but it would lead to a low compression ratio.

4. TESTING

The testing was performed over 20 piano audio files and 20 electric guitar audio files of 44 100 samples each (one second duration) for our audio codec and also for LAME MP3 [version 3.99.5] [x86] codec with constant bit rates (CBR) 128 Kbps and 256 Kbps in order for us to be able to assess the effectiveness of our codec comparing to it. You can see the results of the testing in the Table.

We compressed each original WAV audio file with a coder to a .cod or MP3 file and then decoded the compressed file to a WAV file again. After that, we counted the relative error between the audio signal of the original WAV file and the signal of the decompressed WAV file as shown in (15). The relative error values presented in the Table are average for both sets of 20 audio files. The relative error threshold in our codec is 5 %, thus, the average relative error values relating to our codec are less than 0.05.

Using a bit rate of 128 Kbps typically results in a sound quality equivalent to what we would hear on the radio. As you can see the relative error for our codec is less than the one for LAME MP3 codec with CBR 128 Kbps. However, the relative error is only an objective sound quality measurement. When we were assessing the subjective perceptual quality of the sound produced by our codec by listening to it in headphones, the sound happened to be distinctly worse than the sound of the audio files produced by LAME MP3 codec with CBR 128 Kbps.

The compression ratio values related to our codec are average for both sets of 20 audio files in the Table. The compression ratio reached by LAME MP3 was identical for all the audio files (5.15 times for 128 Kbps and 2.575 times for 256 Kbps) as we used it in the constant bit rate mode.

Considering the work of our codec, one can also notice that the average compression ratio reached by the codec for “simple” piano files is two times bigger than the ratio for “complicated” electric guitar files. The reason for this is that the iterations (11) converge worse for the latter. Therefore, more frames of an electric guitar file are written fully to an output compressed file .cod increasing its size.

We consider the codec as an interesting application of parametric identification methods in the time domain. The key point of its work is the variational (STLS) objective function (5) that is minimized in our modeling algorithm. The iterations (11) minimize the function over difference equation coefficients effectively for simple piano music files and the algorithms STLS1 and STLS2 used in [9] also solve the STLS problem well for speech signals. However, the STLS approach does not work properly for more complicated music files. We are going to

handle the problem by dividing an audio signal into frequency subbands and coding each of them independently. Additionally, we are planning to search for more efficient ways of minimizing the variational objective function; one option is to do it over the roots of the characteristic polynomial of the difference equation.

The work has been supported by the Russian Foundation for Basic Research (project No. 13-01-00329).

REFERENCES

1. **You Y.** *Audio Coding Theory and Applications*, New York: Springer, 2010.

2. **Hermus K., Verhelst W., Lemmerling P., Wambacq P., van Huffel S.** Perceptual Audio Modeling with Exponentially Damped Sinusoids. *Signal Processing*, 2005, Vol. 85, No. 1, Pp. 163–176.

3. **de Groen P.P.N.** An Introduction to Total Least Squares. *Nieuw Archief voor Wiskunde*, 1996, Vol. 14, No. 4, Pp. 237–253.

4. **Lomov A.A.** Variational identification methods for linear dynamic systems and the local extrema problem. *Upravlenie Bol'shimi Sistemami*, 2012, Vol. 39, Pp. 53–94. Available: http://ubs.mtas.ru/upload/library/UBS3903.pdf (Accessed: 01.11.2015). (rus)

5. **Egorshin A.O.** *Computational closed algorithms of identification of linear objects. Optimal and Self-Adjusting Systems*, Novosibirsk, 1971, Pp. 40–53.

6. **Egorshin A.O.** Least square method and fast algorithms in variational problems of identification and filtration (VI method). *Avtometriya*, 1988, Vol. 1, Pp. 30–42. (rus)

7. **Moor B.D.** Structured total least squares and L2 approximation problems. *Linear Algebra and its Applications*, 1993, Vol. 188–189, Pp. 163–207.

8. **Roorda B., Heij C.** Global total least squares modelling of multivariable time series. *The IEEE Transactions on Automatic Control*, 1995, Vol. AC-40, Pp. 50–63.

9. **Lemmerling P., Mastronardi N., van Huffel S.** Efficient implementation of a structured total least squares based speech compression method. *Linear Algebra and its Applications*, 2003, Vol. 366, Pp. 295–315.

10. **Osborne M.R., Anderssen R.S.** *A class of nonlinear regression problems. Data Representation*, Saint Lucia: University of Queensland Press, 1970, Pp. 94–101.

11. **Khassina E.M.** Audio Files Compression with the Variational Method of Identification of Modeling Difference Equations. *Proceedings of the International Workshop on Applications in Information Technology*, The University of Aizu in cooperation with St. Petersburg State University and Peter the Great St. Petersburg Polytechnic University, Aizu-Wakamatsu, Japan, Oct. 2015, Pp. 1–4. Available: http://kspt.ftk.spbstu.ru/media/files/2015/iwait-2015/proceedings/iwait-2015-e-proceedings-release.pdf (Accessed: 01.11.2015).

12. **Khassina E.M.** File compression by use of the method of variation identification of modeling difference equations. *Proceedings of the X International Conference System Identification and Control Problems SICPRO'15,* Moscow, 2015, Pp. 648–658. Available: http://www.sicpro.org/sicpro15/proc/procdngs/648.pdf (Accessed: 01.11.2015).

СПИСОК ЛИТЕРАТУРЫ

1. **You Y.** Audio Coding Theory and Applications. New York: Springer, 2010.

2. **Hermus K., Verhelst W., Lemmerling P., Wambacq P., van Huffel S.** Perceptual Audio Modeling with Exponentially Damped Sinusoids // Signal Processing. 2005. Vol. 85. No. 1. Pp. 163–176.

3. **de Groen P.P.N.** An Introduction to Total Least Squares // Nieuw Archief voor Wiskunde. 1996. Vol. 14. No. 4. Pp. 237–253.

4. **Ломов А.А.** Вариационные методы идентификации линейных динамических систем и проблема локальных экстремумов // Управление большими системами. 2012. № 39. С. 53–94 [электронный ресурс] / URL: http://ubs.mtas.ru/upload/library/UBS3903.pdf (дата обращения: 01.11.2015).

5. **Егоршин А.О.** Computational closed algorithms of identification of linear objects. Optimal and Self-Adjusting Systems. Новосибирск, 1971. С. 40–53.

6. **Егоршин А.О.** Метод наименьших квадратов и быстрые алгоритмы в вариационных задачах идентификации и фильтрации (метод ВИ)// Автометрия. 1988. № 1. С. 30–42.

7. **Moor B.D.** Structured total least squares and L2 approximation problems // Linear Algebra and its Applications. 1993. Vol. 188–189. Pp. 163–207.

8. **Roorda B., Heij C.** Global total least squares

modelling of multivariable time series // The IEEE Transactions on Automatic Control. 1995. Vol. AC-40. Pp. 50–63.

9. **Lemmerling P., Mastronardi N., van Huffel S.** Efficient implementation of a structured total least squares based speech compression method // Linear Algebra and its Applications. 2003. Vol. 366. Pp. 295–315.

10. **Osborne M.R., Anderssen R.S.** A class of nonlinear regression problems. Data Representation. Saint Lucia: University of Queensland Press, 1970. Pp. 94–101.

11. **Khassina E.M.** Audio Files Compression with the Variational Method of Identification of Modeling Difference Equations // Proceedings of the International Workshop on Applications in Information Technology (IWAIT-2015). The University of Aizu in cooperation with St. Petersburg State University and Peter the Great St. Petersburg Polytechnic University. Aizu-Wakamatsu, Japan, 2015. Pp. 1–4 [электронный ресурс] / URL: http://kspt.ftk.spbstu.ru/media/files/2015/iwait-2015/proceedings/iwait-2015-e-proceedings-release.pdf (дата обращения: 01.11.2015).

12. **Khassina E.M.** Сжатие аудиофайлов методом вариационной идентификации моделирующих разностных уравнений // Матер. X междунар. конф. Идентификация систем и задачи управления. SICPRO-15. М., 2015. С. 648–658 [электронный ресурс] / URL: http://www.sicpro.org/sicpro15/proc/procdngs/648.pdf (дата обращения: 01.11.2015).

KHASSINA Eugenia M. *Novosibirsk State University.*
630090, Pirogov Str. 2a, Novosibirsk, Russia.
E-mail: jenya-100@yandex.ru

ХАСИНА Евгения Михайловна – *магистрант кафедры систем информатики Новосибирского государственного университета.*
630090, Россия, г. Новосибирск, ул. Пирогова, д. 2а.
E-mail: jenya-100@yandex.ru

LOMOV Andrei A. *Sobolev Institute of Mathematics.*
630090, Acad. Koptyug Str. 4, Novosibirsk, Russia.
E-mail: lomov@math.nsc.ru

ЛОМОВ Андрей Александрович – *старший научный сотрудник Института математики имени С.Л. Соболева СО РАН, доктор физико-математических наук.*
630090, Россия, г. Новосибирск, ул. Коптюга, д. 4.
E-mail: lomov@math.nsc.ru