Audio Files Compression with the Variational Method of Identification of Modeling Difference Equations*

Eugenia M. Khassina
Novosibirsk National Research State University
Department of Information Technologies
Novosibirsk, Russia
jenya-100@yandex.ru

Andrei A. Lomov†
Sobolev Institute of Mathematics of the Siberian
Branch of the Russian Academy of Sciences
Novosibirsk National Research State University
Novosibirsk, Russia
lomov@math.nsc.ru

ABSTRACT

In this paper we consider the variational audio compression algorithm based on signal modeling with solutions of linear difference equations from a certain parametric family in the time domain using the STLS cost function. The identification of a modeling difference equation parameters for each frame of an audio file signal allows one to perform compression of the file representing the signal frames in the adaptive Laplace basis of exponentially damped sinusoids. Such an approach better reflects the physics of audio signals generated by real musical instruments than the traditional Fourier representation of the signals with harmonics that is used in some popular audio codecs such as OGG Vorbis and MP3.

Categories and Subject Descriptors

G.1.2 [**Numerical Analysis**]: Approximation—*structured total least squares approximation*; G.1.6 [**Numerical Analysis**]: Optimization—*least squares methods*; H.5.5 [**Information interfaces and presentation**]: Sound and Music Computing—*modeling*

Keywords

audio signals modeling, audio codec, exponential sinusoidal model, parametric identification, difference equations, variational identification method, structured total least squares

1. INTRODUCTION

Lossy audio codecs such as MPEG-1 codecs (MP1, MP2 and MP3) and OGG Vorbis 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 can not be considered as a quasi-stationary episode. That is why MP3 audio codec, for instance, has to use the Modified Discrete Cosine Transform (MDCT) with windows of varied length while processing a signal. Using a short-window mode of the encoding scheme allows to avoid what is commonly referred to as a pre-echo artifact [1].

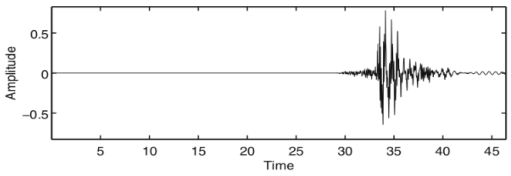

Figure 1: Transient [1].

In Figure 1 a transient can be seen. 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 a_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 [2] a Total Least Squares (TLS) problem of order $2K$ is solved to estimate ω_i and γ_i, where K is a number of available exponentially damped sinusoids predefined by a user. On the basis of the TLS-ESM scheme an experimental audio codec was created and tested [2]. An essential disadvantage of the method, however, is that TLS tries to make the original and the modeled signals as close to each other as possible in the time domain not on the whole frame of the signal but on separate sets of 2K samples along the frame, considering

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

†Research supervisor of the work.

IWAIT '15, Oct. 8 – 10, 2015, Aizu-Wakamatsu, Japan.

the sets independent.

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 [5, 6, 7], close to the Structured Total Least Squares (STLS) [8] and the Global Total Least Squares (GTLS) [9] methods.

In [3] a vocoder based on 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, speech spectrum is rather simple that makes speech signals easily compressible. Our experiments showed that Newton's iterative algorithms, to which STLS1 and STLS2 used in [3] belong, have bad convergence when solving the STLS problem for music audio files with wider spectra.

2. THEORETICAL ASPECTS

2.1 Coding of 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. By $s[k]$, $k = \overline{1,N}$ denote a frame of samples.

We will treat the vector $s \doteq (s[1]; \ldots; s[N])$ as a perturbed observation of a solution process $z \doteq (z[1]; \ldots; z[N])$ 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 $p = 3$:

$$\begin{aligned} z[k+3] + \alpha_2 z[k+2] + \alpha_1 z[k+1] + \alpha_0 z[k] &= 0, \qquad (2) \\ k &= \overline{1, N-3}. \end{aligned}$$

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, suppose that the single real root of the characteristic polynomial is ξ_2. We are interested only in real solutions of the difference equation, as audio samples of observation s are real. Therefore, we can transform the general solution of (2) to the real form as follows:

$$\begin{aligned} z[k] = C_1\xi_1^k + \bar{C_1}\bar{\xi_1}^k + C_2\xi_2^k = \qquad (3) \\ = A_1\rho_1^k \cos(k\omega_1) + A_2\rho_1^k \sin(k\omega_1) + A_3\rho_2^k\,,\ \forall i\ A_i \in \Re. \end{aligned}$$

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

$$\gamma \doteq \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 1 \end{pmatrix}^\top.$$

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

$$J(\gamma) = \|s - z(\gamma)\|^2, \quad z(\gamma) \doteq \arg\min_{z:\,(2)} \|s - z\|^2. \qquad (4)$$

This variational problem was first formulated and solved by A. O. Egorshin [5, 6]. 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]. As the initial γ for the iterative algorithm we use the least-squares estimate γ_{LS} [11].

To describe the minimization iterative algorithm, first, let us notice that the difference equation (2) can be transformed to the matrix form:

$$\underbrace{\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 1 & & 0 \\ 0 & \alpha_0 & \alpha_1 & \alpha_2 & 1 & \\ \vdots & & \ddots & \ddots & \ddots & \ddots \\ 0 & & & \alpha_0 & \alpha_1 & \alpha_2 & 1 \end{pmatrix}}_{G} \underbrace{\begin{pmatrix} z[1] \\ z[2] \\ \vdots \\ z[N] \end{pmatrix}}_{z} = 0. \qquad (5)$$

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

$$\underbrace{\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 1 & & 0 \\ 0 & \alpha_0 & \alpha_1 & \alpha_2 & 1 & \\ \vdots & & \ddots & \ddots & \ddots & \ddots \\ 0 & & & \alpha_0 & \alpha_1 & \alpha_2 & 1 \end{pmatrix}}_{G_\gamma} \underbrace{\begin{pmatrix} s[1] \\ s[2] \\ \vdots \\ s[N] \end{pmatrix}}_{s} \equiv \qquad (6)$$

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

Now, the iterations with an updating inverse matrix which solve the variational identification task (4) are:

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

$$\begin{cases} \tau = \left(V(s)^\top \left(G_{\gamma(k)} G_{\gamma(k)}^\top\right)^{-1} V(s)\right)^{-1} \cdot\ \gamma(k)\,, \\ \gamma(k+1) = \frac{1}{(0\ldots01)\tau}\,\tau\,. \end{cases} \qquad (7)$$

The last row means the division of the whole auxiliary vector τ 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 $\left(G_{\gamma(k)} G_{\gamma(k)}^\top\right)^{-1}$ which is updated on each iteration.

Using the calculated estimate for γ, we find the modeling process $z(\gamma)$ nearest to the observation s as the linear projection [5, 6]:

$$z(\gamma) = \left(I - G_\gamma^\top (G_\gamma G_\gamma^\top)^{-1} G_\gamma\right) \cdot s\,, \qquad (8)$$

where I is an identity matrix.

The obtained coefficient vector γ and the corresponding process $z(\gamma)$ that fit the observation s are used in the further course as we will show in subsection 2.2.

Note also that in the section the order p of the difference equation is considered known. A way of choosing p and problems encountered when using the iterations (7) at the implementation stage will be conveyed in the next section.

2.2 Decoding of an audio frame

From (3) we can see that, in order to restore process z, for each complex-conjugate pair of the 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

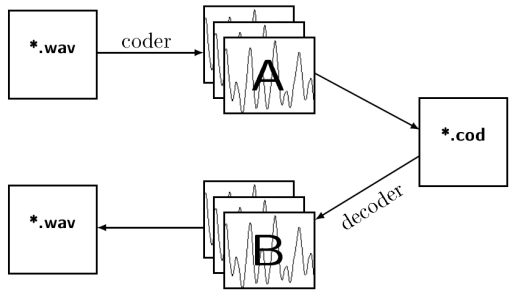

Figure 2: General scheme of work of the codec. A. Original frames. B. Decompressed frames.

that root of the pair that lies above the real axis and we also need to know the set of real coefficients A_i. 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 an order of a difference equation and the frame size type. At the paper we will not describe in detail the structure of the frame service information byte.

The only thing left for us to understand is how to get coefficients A_i, $i = \overline{1,p}$. Let us transform the expression (3) to the matrix form:

$$\begin{pmatrix} z[1] \\ \vdots \\ z[N] \end{pmatrix} = \underbrace{\begin{pmatrix} \rho_1 \cos(\omega_1) & \rho_1 \sin(\omega_1) & \rho_2 \\ \rho_1^2 \cos(2\omega_1) & \rho_1^2 \sin(2\omega_1) & \rho_2^2 \\ \vdots & \vdots & \vdots \\ \rho_1^N \cos(N\omega_1) & \rho_1^N \sin(N\omega_1) & \rho_2^N \end{pmatrix}}_{H} \cdot \underbrace{\begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix}}_{d} = Hd. \tag{9}$$

Using the iterations (7) we have found the model G and the process z, corresponding to the original observation s, such that $Gz = 0$. 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:

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

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

$$d = (\bar{H}^\top \bar{H})^{-1} \bar{H}^\top z, \tag{10}$$

where $\bar{H}$ is a submatrix composed of $\geqslant p$ rows of matrix H.

3. CODEC IMPLEMENTATION

We have realized an audio codec (consisted of two modules: a coder and a decoder) based on the theory described above in Scilab environment (`http://www.scilab.org`), similar to MATLAB. The codec realization and testing were performed on the Debian GNU/Linux operating system. As input the codec accepts a mono WAV audio file with sample rate 44.1 KHz and bit depth 16 bits. As output the codec produces a compressed file, whose structure was defined by us, with a new extension .cod. In Figure 2 you can see the general scheme of work of the codec.

An original audio signal to be compressed is divided into frames of N samples with a shift of $(N - M)$ samples. That is, each pair of consequent frames overlap by M samples to be glued smoothly after their decompression. After the

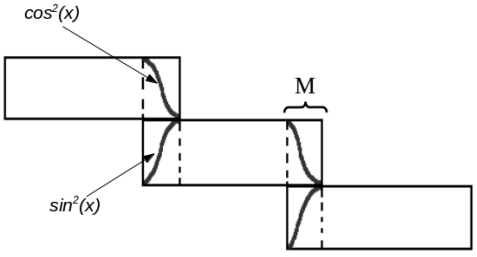

Figure 3: Gluing of frames after decompression.

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 time index k runs from 0 to $M - 1$ and $c(M - 1) = \pi/2$. In Figure 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 (7) 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 $M = 10$.

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 (8) is found. After this, the relative modeling error is counted as follows:

$$\frac{\|z - s\|}{\|s\|} \leqslant 5\,\%, \tag{11}$$

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 .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 44100 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 Table 1.

We compressed each original WAV audio file with a coder to .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 (11). The relative error values presented in the Table 1 are average for

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

Table 1: Testing results.

the 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 usually 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 relating to our codec are average for the both sets of 20 audio files in the Table 1. 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 of it is that the iterations (7) 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.

5. CONCLUSIONS

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 (4) that is minimized in our modeling algorithm. The iterations (7) minimize the function over a difference equation coefficients effectively for simple piano music files and the algorithms STLS1 and STLS2 used in [3] 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. Besides, we search for more efficient ways of minimization of the variational objective function. For instance, we try to do it over the roots of the characteristic polynomial of the difference equation.

References

[1] Yuli You. *Audio Coding Theory and Applications*. New York: Springer, 2010.

[2] Kris Hermus et al. "Perceptual Audio Modeling with Exponentially Damped Sinusoids". In: *Signal Processing* 85.1 (2005), pp. 163–176.

[3] Philippe Lemmerling, Nicola Mastronardi, and Sabine Van Huffel. "Efficient implementation of a structured total least squares based speech compression method". In: *Linear Algebra and its Applications* 366 (2003), pp. 295–315.

[4] Pieter P. N. de Groen. "An Introduction to Total Least Squares". In: *Nieuw Archief voor Wiskunde* 14.4 (1996), pp. 237–253.

[5] Andrei A. Lomov. "Variational identification methods for linear dynamic systems and the local extrema problem". In: *Upravlenie Bol'shimi Sistemami* 39 (2012). `http://ubs.mtas.ru/upload/library/UBS3903.pdf` [Last accessed 26 June 2015], pp. 53–94.

[6] Alexey O. Egorshin. "Computational closed algorithms of identification of linear objects". In: *Optimal and Self-Adjusting Systems*. Novosibirsk: ed. of Institute of Automation and Electrometry of the USSR Academy of Sciences, 1971, pp. 40–53.

[7] Alexey O. Egorshin. "Least square method and fast algorithms in variational problems of identification and filtration (VI method)". In: *Avtometriya* 1 (1988), pp. 30–42.

[8] Bart De Moor. "Structured total least squares and L_2 approximation problems". In: *Linear Algebra and its Applications* 188-189 (1993), pp. 163–207.

[9] Berend Roorda and Christiaan Heij. "Global total least squares modelling of multivariable time series". In: *the IEEE Transactions on Automatic Control* AC-40 (1995), pp. 50–63.

[10] Michael Robert Osborne and Robert Scott Anderssen. "A class of nonlinear regression problems". In: *Data Representation*. Saint Lucia: University of Queensland Press, 1970, pp. 94–101.

[11] Eugenia M. Khassina. "File compression by use of the method of variation identification of modeling difference equations". In: *Proceedings of the X International Conference "System Identification and Control Problems" SICPRO '15*. V.A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences. Moscow, Russia, Jan. 2015, pp. 648–658.

[12] Eugenia M. Khassina. "Modifications of Vorbis algorithms for audio file compression". In: *Materials of LII International Scientific Student's Conference "The Student and scientific and Technical Progress"*. Section "Mathematical Modeling". Novosibirsk State University. Novosibirsk, Russia, 2014, p. 151.