14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

STREAMING OVER THE INTERNET WITH A SCALABLE PARAMETRIC AUDIO CODER

J.C. Cuevas-Martinez, P. Vera-Candeas, and N. Ruiz-Reyes

Department of Telecommunication, University of Jaen
C/ Alfonso X el Sabio 28, 23700, Linares, Spain
phone: + (34) 953648554, fax: + (34) 953648508, email: {jccuevas, pvera, nicolas}@.ujaen.es

ABSTRACT

Audio compression has progressively gained higher importance in the Internet thanks to massive amount of multimedia services on it. This new services require coders adapted to that new environment. Therefore, new generation coders use more complex models focused on features which make possible its use for audio streaming over the Internet, mainly low bit rate, scalability and robustness. In our case, a good trade-off between bit rate reduction and audio quality is achieved by using parametric audio coding, and furthermore, this coder has a scalable version, optimized for streaming requirements. This coder avoids differential information between coded audio segments and uses a layered scheme for changing straightforwardly the bit rate. The results reveal our coder as a good candidate for massive distributed audio applications, like music on demand, radio broadcasting or real-time streaming audio. In this article are shown the main features of this coder and their implication on streaming.

1. INTRODUCTION

The continuous increase of multimedia services and content over the Internet has encouraged, in the last years, the search of low bit rate coders with good quality. Moreover, wireless media access begun to be usual so services have to be available everywhere to a wide number of users. From this can be derived two objectives: low bit rate-good quality coding and streaming oriented. For the first one, a parametric audio coder has been developed, and for the last, scalability has been added. Parametric coding of audio signals has become a popular tool for representing audio signals at very low bit rates [1][2][3]. This high level description or model for audio signals can provide a framework to meet the demands of the Internet streaming audio problem and the increase of wireless access through GSM/GPRS and UMTS. The nature of high level signal models allows large compression rates, scalable compression and flexibility, all required for a suitable deployment over the Internet. The coder uses a three components signal model which assumes the signal to be represented as an addition of sines, transients and noise. Then, using psychoacoustic parameters, each component is divided among layers to provide scalability.

In section 2 are shown parametric coder that makes possible the real-time scalability and its streaming applications. Then, section 3 show the scalability features and how can them be used in audio streaming. In section 4 are shown the application of this coder for audio streaming. Finally, section 5 shows the conclusions for this work.

2. SIGNAL MODEL

The signal models used in audio compression assume an underlying structure to the signal in question, in that way, a wide range of audio signals intuitively fit into the three-part model of Sines, Transients and Noise. Transients describe drum hits and the stacks of many instruments, sines describe signal components that have a distinct pitch, and noise often describes the rest of the signal that is neither sinusoidal nor transient. This model consists on three parts that work together and complement each other to form a complete and robust signal model, which makes possible a highly optimized audio compression scheme. Low bit rates are achieved if model parameters are efficiently encoded according to psycho-acoustic criteria. Figure 1 shows the encoder stage of the proposed parametric audio coder.

We have used as a modelling tool for transients and sines the matching pursuits algorithm. This algorithm is an iterative method which extracts at each iteration the atom more correlated with the residual signal. In this way, matching pursuits choose at each iteration the atom that extracts more energy from the current residue. This algorithm can be adapted to modelise transients or sines by making a good choice of the dictionary. This dictionary is generally composed of a family of atoms. For example, sinusoidal modelling can be achieved by matching pursuits and a dictionary composed of complex exponentials.

The matching pursuit algorithm was introduced by Mallat and Zhang in [Mallat93]. So as to explain the basic ideas concerning this algorithm, let's suppose a linear expansion approximating the analyzed signal $\mathbf{x}$ in terms of functions $\mathbf{g}_m$ chosen from a over-complete dictionary $\mathbf{D}=\{\mathbf{g}_m ; m=0,1, \ldots, L\}$.

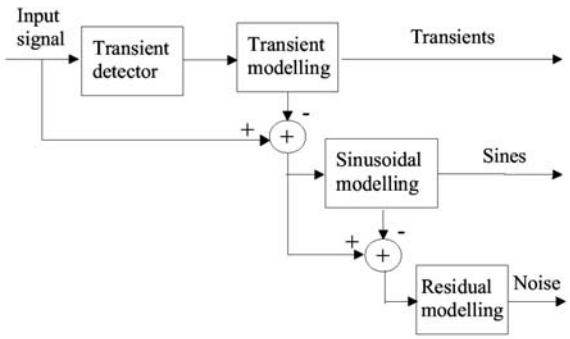

Figure 1 - Block diagram of the encoder stage

At first iteration of matching pursuits, the atom $\mathbf{g}_m$ which gives the largest inner product with the analyzed signal $\mathbf{x}$ is chosen. The contribution of this vector is then subtracted from the signal and the process is repeated on the residue. At the *i*-th iteration, the residue is:

$$\mathbf{r}^i = \begin{cases} \mathbf{x} & i = 0 \\ \mathbf{r}^{i-1} - \boldsymbol{\alpha}_{m(i)}\mathbf{g}_{m(i)} & i > 0 \end{cases} \tag{2}$$

where $\boldsymbol{\alpha}_{m(i)}$ is the weight associated to the optimum atom $\mathbf{g}_{m(i)}$ at the *i*-th iteration, and *m(i)* the dictionary index of the optimum atom chosen at the *i*-th iteration. Both $\boldsymbol{\alpha}_{m(i)}$ weight and the optimum atom $\boldsymbol{g}_{m(i)}$ are computed by minimising the residual energy at each iteration

$$\min_{\mathbf{g}_m \in \mathbf{D}} \left\| \mathbf{r}^i \right\|^2, \quad \mathbf{r}^i = \mathbf{r}^{i-1} - \boldsymbol{\alpha}_m^i \mathbf{g}_m^i \tag{3}$$

where $\boldsymbol{\alpha}_m^i$ are the weights associated to each element $\mathbf{g}_m$ of dictionary $\mathbf{D}$. From this definition we can see that the residual energy is minimised by the chosen atom $\boldsymbol{g}_{m(i)}$. The computation of $\boldsymbol{\alpha}_m^i$ weights is obtained substituting $\mathbf{r}^i$ by its definition and minimising the result. The solution can be written as

$$\max_{\mathbf{g}_m \in \mathbf{D}} \left\| \boldsymbol{\alpha}_m^i \right\|^2, \quad \boldsymbol{\alpha}_m^i = \frac{\left\langle \mathbf{r}^{i-1}, \mathbf{g}_m \right\rangle}{\left\| \mathbf{g}_m \right\|^2} \tag{4}$$

To enable representation of a wide range of signal features, a large dictionary of time-frequency atoms is used in matching pursuit. The computation of correlations $\left\langle \mathbf{r}^{i-1}, \mathbf{g}_m \right\rangle$ for all $\mathbf{g}_i \in \boldsymbol{D}$ at each iteration is highly computational consuming. As derived in [Mallat93], this computation effort can be substantially reduced using an updating formula based on equation (2). The correlations at the *m*-th iteration are given by:

$$\left\langle \mathbf{r}^i, \mathbf{g}_m \right\rangle = \left\langle \mathbf{r}^{i-1}, \mathbf{g}_m \right\rangle - \boldsymbol{\alpha}_{i(m)} \left\langle \mathbf{g}_{m(i)}, \mathbf{g}_m \right\rangle \tag{5}$$

where the only new computation required for the correlation updating procedure refers to the cross-correlation term $\left\langle \mathbf{g}_{m(i)}, \mathbf{g}_m \right\rangle$, which can be pre-calculated and stored, once over-complete set $\mathbf{D}$ has been determined. For first iteration, the computation of correlations between the signal and all atoms, $\left\langle \mathbf{x}, \mathbf{g}_m \right\rangle$, is also needed.

2.1 Transient modelling

We propose using matching pursuits with a dictionary of orthogonal wavelet functions for transient modelling. The over- complete dictionary $\mathbf{D}$ is made up with those functions which give rise to the *J*-depth full Wavelet-Packet (WP) decomposition, being $M_{WP}=J \cdot N$ the WP dictionary size, and *N* the frame length. The inner products of the signal with the wavelet-based atoms in set $\mathbf{D}$ lead to all the wavelet coefficients that can be considered in the *J*-depth full WP tree. These coefficients can be identified using three indexes, $\{s,p,d\}$, which indicate the sub-band at a given decomposition depth, the decomposition depth and the delay, respectively. The wavelet coefficients at the *i*-th iteration of matching pursuit and the wavelet-based atoms can be expressed as follows:

$$\begin{aligned} \boldsymbol{\alpha}_{\{s,p,d\}}^i &= \left\langle \mathbf{r}^{i-1}, \mathbf{g}_{\{s,p,d\}} \right\rangle \\ g_{\{s,p,d\}}[n] &= g_{\{s,p\}}[n - 2^p d] \end{aligned} \tag{6}$$

According to (5), the only necessary correlations to implement the matching pursuits are $\left\langle \mathbf{x}, \mathbf{g}_{\{s,p,d\}} \right\rangle$ and $\left\langle \mathbf{g}_{\{s_1,p_1,d_1\}}, \mathbf{g}_{\{s_2,p_2,d_2\}} \right\rangle$. The first ones are obtained from the WP transform of $\mathbf{x}$, while correlations between atoms are precalculated and memory stored. These cross-correlations are formulated in [VeraIEEE] when wavelet-based dictionaries built from orthonormal wavelets are used, which results in:

$$\left\langle g_{\{s_1,p_1,d_1\}}[n], g_{\{s_2,p_2,d_2\}}[n] \right\rangle = \begin{cases} \boldsymbol{\delta}[d_2 - d_1] & s_1 = s_2, p_1 = p_2 \\ 0 & s_2 \neq \left\lfloor \dfrac{s_1}{2^{p_1 - p_2}} \right\rfloor \\ g_{\{s,p,d_1\}}[d_2] & s_2 = \left\lfloor \dfrac{s_1}{2^{p_1 - p_2}} \right\rfloor \end{cases} \tag{7}$$

where $p=p_1-p_2$ and $s=((s_1))_{2^p}$. Therefore, according to (7), the iterative procedure to update correlations requires impulsive responses of the synthesis WP tree branches to be stored [VeraIEEE].

2.2 Sinusoidal modelling

For sinusoidal modelling, we propose the use of matching pursuits with a dictionary of windowed complex exponential functions, instead of a set of windowed sinusoidal functions, in order to reduce the computational complexity. Using windowed complex exponential sets, only the frequency of every exponential function must be determined, which involves a significant reduction of the dictionary size [3]. Phase is computed by the correlations due to its complex nature. The functions that belong to the considered set can be expressed as follows:

$$g_k[n] = S_w \cdot w[n] \cdot e^{j\frac{2\pi k}{2L}n}, \qquad k = 0,1,\ldots,L \tag{8}$$

The constant S_w is selected in order to obtain unit-norm functions, $w[n]$ is the N-length analysis window, $L+1$ the number of frequencies within the dictionary and k is the index of each discrete frequency. Amplitude, frequency and phase are the three parameters that define each extracted tone by the sinusoidal model.

The implemented matching pursuits algorithm for sinusoidal modelling is psychoacoustic-adaptive as in [6]. According to this approach, the tone extracted at each iteration is the perceptually the most important one. This perceptual measure is basically a modification of the energy weights, α_k^i, defined by matching pursuits taking perceptual information into account in the following way:

$$\left\|\alpha_k^i\right\|_{PMP} = \int_0^B \frac{\left\|\alpha_k^i\right\|^2 \left|G_k(b)\right|^2}{T(b)} db \qquad (9)$$

where Perceptual Matching Pursuits (PMP) algorithm is defined by integrating into the Bark scale the division between the energy of each atom, $|G_k(b)|^2$, and the estimated threshold, $T(b)$. Also, this algorithm can be halted following a perceptual stopping criterion [6].

The energy parameters in time and frequency of the final residual are obtained by linear predictive coding (LPC) to model this signal as a noise. For the frequency case, the main drawback of LPC model is that the underlying spectral resolution is not matched to that of the human auditory system. Frequency warping in combination with LPC, termed warped LPC [WLPC], does allow transformation (or warping) of the frequency axis according to psycho-acoustic principles, and we have therefore applied it to noise modelling.

Summarising, this audio coder extracts from the input audio signal a set of different parameters to be sent to the decoder. These parameters represent the information provided by the three-part model (Sines + Transient + Noise). They are quantified using psycho-acoustical information to ensure that decoded signals are perceptually identical to the original ones. Before transient modelling, transient detection is required. Our transient detector is based on sudden energy change detection. Besides, an adaptive tiling of the time axis is required to achieve a right performance of the proposed audio coder. We have used the algorithm proposed in [5]. The audio signals compressed with this coder are mono and sampled at 44.100 Hz with 16 bits per sample. The coder results for good perceptual quality give bit rates about 10-12 Kbps, for test signals and commercial music too.

3. LAYERED SCALABLE QUALITY FEATURES

This parametric coder develops an efficient, accurate and flexible multi-part model for wide-band speech and audio coding. Besides, coded information is generated with a streaming oriented structure which divides the original audio signal among several coded segments of variable length. This structure is based in the division of a segment in several layers, each one with different audio information. Quality can be increase just getting more layers from coded audio, but it is not necessary to use all of them to recognise the audio signal. The layer splitting process is explained below.

3.1. Layer splitting

Once, all the parameters have been extracted from the audio segment, they are classified depending on their significance into a 5 layers scheme: sines are organized in terms of their perceptual importance and transients are layered in function of the energy of each wavelet-packet atom. However, noise model is more difficult to be layered. We have followed this principle: no energy is removed. So, tones at each layer that are not included due to bit rate restrictions are modelled by noise modelling [Myburg]. As a consequence, to implement this principle, noise coded parameters correspond to the residual of the first (lowest bit rate) layer. The synthesized noise for the other layers is estimated removing the energy of sines and transients for these layers from the energy of layer 1 noise. Consequently, we can state for layer 1 that this layer usually has a bit rate around 7-8 Kbps and, obviously not a good quality, but the audio signal can be recognised perfectly and the main artefacts are noisy background. The resulting bit rates are depicted in figure 2 and subjective results based on MUSHRA methodology are presented in figure 3. MPEG proposed signals for coding tests have been used.

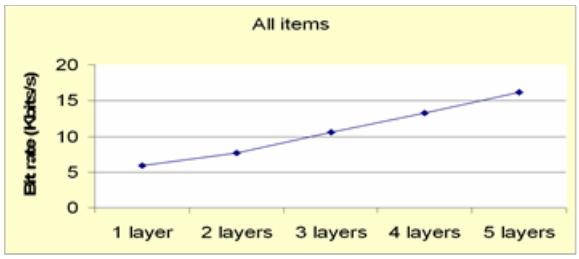

Figure 2 – Resulting bit rates

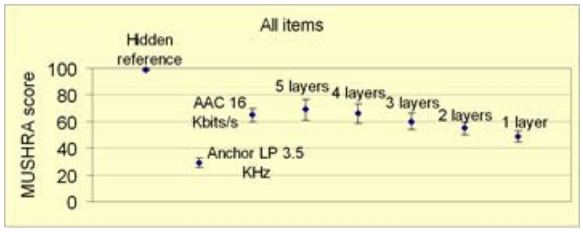

Figure 3 – Subjective results

4. STREAMING APPLICATIONS OF THE CODER

From now on, some keys are considered in order to achieve streaming application. The main one: Low bit rate with high quality audio has been achieved with techniques like three part model and matching pursuits described above. These features allow bit rates from 32 Kbps to 8 Kbps with very good or good perceptual quality, making possible even raw streaming over low speed Internet connections. But, this feature, that is usually the unique in most coders used in audio streaming, has been complemented with real time scalable

quality for streaming over a digital network. This has been possible due to a layered model for coded audio information, which groups the three part model parameters, transients, sinusoids and noise with similar psychoacoustic values, in the same layer.

The structure of coded audio information is tree-based. Therefore, the audio file or real-time stream of audio is divided among fragments of variable number of samples, so this coder has a variable instantaneous bit rate. Each segment is also divided in several independent layers. Every layer carries full absolute audio information about transients and sinusoids of its quality level. These layers are organized from bottom to top in complexity, so lower layers have the audio components which allow recognizing the original signal, and higher layers convey refinements to increase perceptual quality.

Because there is no inter-segment or inter-layer information, it is possible to decode any segment, using just a few layers, even it lower layers have been missed. Nevertheless, the resulting audio signal will probably lose important psychoacoustic elements which had been carried in lower layers, and the quality of the decoded audio will be very poor.

This system is complemented with a real time scalable decoder, which can recover the audio signal with any number of layers for a segment of coded audio without any inter-segment, or inter-layer information. The layered scheme allows the real-time scalability without any re-coding. Moreover, the audio segments received with low quality can be refined with subsequent segment, which can be sent when network status makes it possible or user requires a better quality.

4.1. Embedded audio coding

Derived from this layered structure, is the capability of the coder to embed audio information from previous segments into new ones. Just with segment sequence number and layer identifier it is possible to refine the quality of a segment received with less quality, for example, due to short network congestion status, and play it without any retransmission of already sent data, and for music download it allows to listen quickly to a sample of a song, and if this song is chosen, just send layer of received segments to achieve the required quality. This capability saves a lot of bandwidth, avoiding spurious retransmissions.

4.2. Multi-channel casting

Another ability of this coder is that it can simply deliver in the same packet different instants of a real time stream. This is very useful for multicast sessions, where receivers can choose the main stream to listen to the most recently coded audio, or choose between a group of low quality streams which have the same audio information coded earlier, divided among intervals of, for example, one or two minutes. Then, when the listener chooses a delayed stream the embedded capability allows refining that low bit rate audio. Furthermore, when the instant of listening reaches the point when listener connected, the information received earlier will be used from that moment, so the user can stop receiving delayed streams. This capability is mainly oriented for real time broadcasting of cultural or sport events over the Internet. In Figure is shown an example of multicast channeling. One user begins to listen to an online concert, but he or she is some seconds late and decides to chooses the one minute late stream. The other streams are received too, but thirty second after the beginning the 2 minutes later stream is replaced by another layer for 1 minute stream, because the user desires more quality. Meanwhile, the main stream is stored. When the connection reaches the minute of audio played, the current played audio change from network to disk, because this information is in the cache of the connection. Then, if the user was storing the played audio could require layers to refine the one minute later stream.

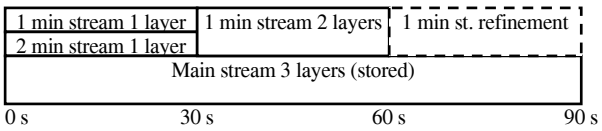

Figure 4 - Packet stream that uses multicast channeling

4.3. Forward multi-time segment streaming.

Other of the features of this layered schema is the error prevention, due to transmission errors in segments. When a segment is discarded because bit errors that can produce a gap in the play-out, several techniques can be used to avoid it:

- Retransmission: This technique involves that the receiver has to say to the sender what segment or segments has lost or have error, and then, the sender has to re-transmit the required information. This solution generated undesired traffic over the network but is the easier to implement, and for not real-time applications could be use without notice it, but in time constrained applications it can derive in a play-out glitch or a delayed response.
- Forward Error Correction Techniques: These methods are commonly used in reliable multicast protocols [12] over best-effort networks. They require very complex algorithms depending of types of information to protect [13], so information added to protect information could affect seriously to protocol and error overhead, which have to be avoided due to low bit rate schema used in this coder.
- Forward multi-time segment streaming: This is the solution that we propose. The main stream does not suffer any increase in bandwidth; just some layers are delayed from others, giving online backup information for the case of segments lost of errors. At the beginning of the stream, a buffering time is required to get enough information from the sender (t_{b1}), about two to five seconds of the main stream (each packet can carry more than one segment). Then, when initial buffering with layer 1 or even 2 are accomplished, the sender begins to send the following packets (t_{m1}) with all the required layers (up to 5). Each packet carries on time layer 1 or 2 and the rest, delayed packets. In that way, if a packet is delayed the

buffered packets are used, decreasing quality only for the period of the delay. If a packet is lost (t_l) or has errors, we can use the lower quality layers received before to avoid any play-out glitch (t_{b2}). Then, the client requires the refinement for that loss and when correct ones arrive again, the stream recovers its quality (t_{m2}), so feedback information is minimum and does not make the play-out to stop. Moreover, the initial buffering time cam be used for handshaking purposes to establish other communication parameters.

As a result, Figure 5 shows graphically the play-out of the audio stream and approximately the band-with used for the stream. This figure shows how error are treated and avoided without play-out stop if $t_l < t_{b1}$. And this can be achieved thanks to the special coder described in this paper and used for audio compression

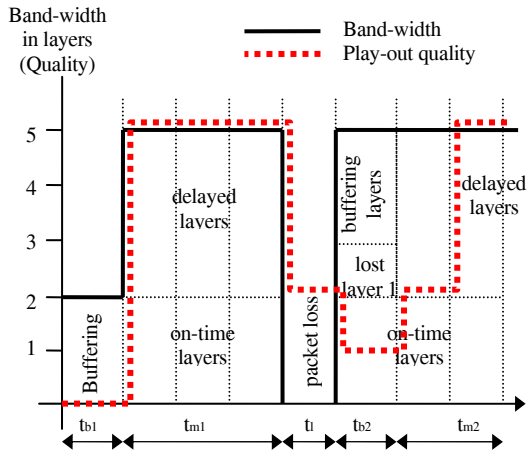

Figure 5- Stream using multi-time segment forwarding

5. CONCLUSIONS

In this document has been shown a scalable parametric coder which makes possible low bit rate codification with a good perceptual quality due to the three part model used. Besides, this coder is specially adapted to audio streaming over the Internet because it can be used for embedded audio, multi-channel streaming or even, instead FEC (or with them) to solve packet lost recovery using only a small buffering time and a dual channel audio stream.

6. REFERENCES

[1] H. Purnhagen, B. Edler, and C. Ferekidis, “Object-Based Analysis / Synthesis Audio Coder for Very Low Bit Rates”, AES 104th Convention, (Amsterdam, The Netherlands), Preprint 4747, May 1998.

[2] S. Levine and J. Smith, “A Sines+Transients+Noise Audio Representation for Data Compression and Time/Pitch Scale Modifications”, AES 105th Convention (San Francisco, CA, USA), preprint 4781, September 1998.

[3] A.C. Den Brinker, E.G.P. Schuiijers and A.W.J. Oomen, "Parametric coding for high quality audio", 112th AES Convention, Preprint 5554, May 2002.

[4] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, J.C. Cuevas-Martínez and J.L. Blanco-Claraco, “A Sinusoidal Modeling Approach Based on Perceptual Matching Pursuits for Parametric Audio Coding”. 118th AES Convention. Barcelona May 2005.

[5] N. Ruiz, M. Rosa, F. Lopez and P. Vera, ”New algorithm for achieving an adaptive tiling of the time axis for audio coding purposes”, Electronic Letters, vol. 80, April 2002, pp. 434-435.

[6] ISO/IEC JTC/SC29 WG11 MPEG. “Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, part 3: Audio.” Standard international IS 11172-3.

[7] ISO/IEC, JTC1/SC29/WG11 MPEG. “Information technology—Coding of moving pictures and associated audio—Audio (non backward compatible coding, NBC)”. ISO/IEC, JTC1/SC29/WG11 MPEG, Committee Draft 13 818-7 1996 (“MPEG-2 NBC/AAC”).

[8] ISO/IEC, JTC1/SC29/WG11 N4380. Audio Subgroup. “Workplan for MPEG-4 Audio Extension I core experiments” ISO/IEC, JTC1/SC29/WG11 MPEG, 2001 (“MPEG-4”).

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[10] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[11] ITU-T Recommendation H.323 “Packet-based multimedia communications systems”. Version 5. July 2003.

[12] C. Neumann, V. Roca and R. Walsh. Large Scale Content Distribution Protocols. ACM SIGCOMM Computer Communication Review. Volume 35, number 5. October 2005.

[13] V. Roca and C. Neumann. Design, Evaluation and Comparison of Four Large Block FEC Codecs, LDPC, LDGM, LDGM Staircase and LDGM Triangle, plus a Reed-Solomon SmallBlock FEC Codec. Research report 5225, INRIA. June 2004.

[14] [Mallat] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries”, IEEE Trans. on Signal Processing, vol. 41, no. 12, pp. 3397-3415, December 1993.

[15] [VeraIEEE] Vera, P. and Ruiz, N. and Rosa, M. and Martinez, D. and Lopez, F., "Transient Modeling by Matching Pursuits with a Wavelet Dictionary for Parametric Audio Coding", IEEE Signal Processing Letters, vol. 11, nº 3, pp. 349-352, Marzo, 2004

[16] [WLPC] Harma, A. and Karjalainen, M. and Savioja, M. and Valimaki, V., "Linear prediction on a warped frequency scale", J. Acoust. Eng. Soc., vol. 48, pp. 1011-1031, 2000.

[17] [Myburg] Myburg, F.P., "Design of a scalable parametric audio coder", University de Eindhoven", 2004