# Perceptually Transparent Audio Watermarking of Real Audio Signals Based On The CSPE Algorithm

Jian Wang[1], Ron Healy[2], Joe Timoney[3]

Department of Computer Science, National University of Ireland, Maynooth
email: (1) jwang@cs.nuim.ie, (2) rhealy@cs.nuim.ie, (3) jtimoney@cs.nuim.ie

*Abstract* - **This paper outlines a transparent and accurate digital audio watermarking system for real audio signals. Based on work in an earlier paper the current work modifies candidate component selection criteria; introduces a component verification process to guarantee accurate recovery of the watermark; identifies and removes audible 'click' phenomena. Test results on real music signals have shown that this watermarking algorithm is transparent and is highly precise in recovering the embedded watermark.**

***Keywords* – CSPE, AUDIO WATERMARKING, STEGANOGRAPHY**.

## I Introduction

Audio watermarking is a technique whereby data is hidden in a cover or host audio signal as a form of steganography. Depending on whether it requires access to the original signal and/or the actual watermark, the decoding process may be described as 'informed', 'semi-blind' or 'blind'.

Our previous work was a semi-blind watermarking scheme [1] based on the 'Complex Spectral Phase Evolution' (CSPE) algorithm [2] for identifying components within the cover signal. While watermarking of audio can be achieved by adding to the cover signal in which the watermark is to be embedded [3][4], other techniques instead modify the cover signal [5][6][7]. In our scheme, the cover signal is divided into frames and the magnitude of one or two frequency components in each frame is modified depending on the watermark bit. At the receiver, the decoder analyses the audio to determine the relationship between certain components and reconstructs the watermark. One advantage of our scheme is that it does not add anything to the original signal.

To identify components for modification, the DFT was initially used for signal decomposition. However, it was found to be unsuitable because perfect component identification is only possible under strict constraints concerning the length of the DFT. For a multi-component signal, such as audio, the required DFT length required to identify all components accurately would be computationally prohibitive. Thus, computationally efficient high-resolution frequency analysis was required. While others such as the Goertzel algorithm [8], or the Quadratic Interpolated FFT (QIFFT) [9], were examined it was found the CSPE algorithm offered the best result for our application. We demonstrated the CSPE approach with synthesised signals in [1] and it was shown to provide excellent results. However, further work found that applying CSPE to component identification in real audio, as opposed to synthesised signals, was not as effective due to various factors. This paper addresses these shortcomings and proposes improvements to the scheme in [1] for the watermarking of real audio.

Section II provides an explanation of the CSPE algorithm. Section III describes improvements to the component selection algorithm and the removal of audible artefacts that are sometimes created by the component modification process. In Section IV, accuracy of recovering watermarks from real audio signals is measured along with results of tests carried out to evaluate perceptual transparency.

## II CSPE Background

The CSPE algorithm was introduced by Short and Garcia [2] as a method of accurately estimating the frequency of components that exist in a signal. The procedure of the CSPE algorithm is depicted in block diagram form in Figure 1.

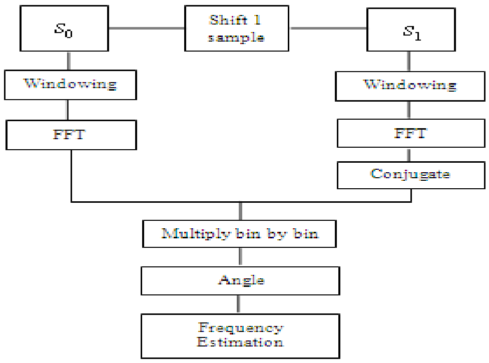

Figure 1: The flow diagram of CSPE

The process can be described as follows: an FFT analysis is performed, first on the signal of interest and again on the same signal but shifted in time by one sample. By multiplying the sample-shifted FFT spectrum with the complex conjugate of the initial FFT spectrum, a frequency dependent function is formed from which the exact values of the frequency components it contains can be detected.

The algorithm produces a graph with a staircase-like appearance where the horizontal parts indicate the exact frequencies of components in the signal.

The width of these parts depends on the main-lobe width of the window function. A wider main-lobe provides better accuracy in frequency identification as it produces a wider horizontal section. Figure 2 shows the output from the CSPE process for a squarewave and a NutallC3 window for illustration purposes.

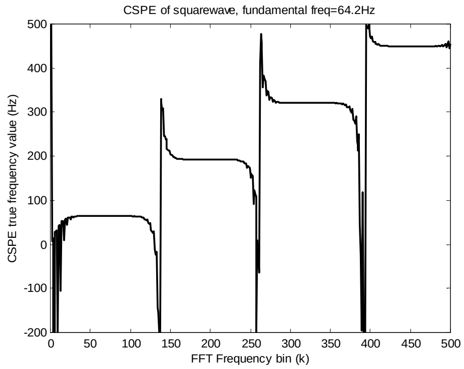

Figure 2: Output of the CSPE process for a squarewave of 64.2 Hz and a NutallC3 window.

We can see from Figure 2 that there are clearly identifiable horizontal step-like sections which represent actual components present in the signal.

During the development of the original scheme described in [1] we concluded that the CSPE algorithm was extremely accurate in frequency estimation for signals containing components that were constant, at least over a relatively long period and when the components were not too close to each other spectrally. CSPE was used to accurately identify the inherent components present in a cover signal, which were then modified to represent the watermark. Recovery of the watermark was almost perfect, achieving 99.99% for 500 signals.

However, when the process was applied to real music signals, we noticed a marked deterioration in results. For real music signals, CSPE is not as accurate in identifying the inherent frequency components, often identifying ‘ghost’ components that were not actually present in the signal. This appears to be as a result of comparatively large numbers of frequency components in real signals which are rapidly changing and often very close to each other in the spectral range.

## III ALGORITHM

We use two indices to evaluate the performance of the scheme: *Precision* and *CodecPrecision*. *CodecPrecision* is an evaluation of the accuracy of watermark recovery for a single instance of the watermark. It is calculated as follows:

$$CodecPrecision_k = \frac{L - \sum_{i=1}^{L} \left| DCode(i) - ECode(i) \right|}{L} \qquad (1)$$

where *ECode* denotes the bit embedded during the embedding process and *DCode* denotes the output bit identified during decode. The watermark bit sequence length is represented by ‘*L*’

*Precision* is derived as a result of an improvement in the scheme which repeatedly embeds the watermark throughout the cover signal. Repeated decoding of the watermark, followed by use of a statistical mode operation to select the most commonly identified bit in each index, acted as an error-correction mechanism. *Precision* is calculated as follows:

$$Precision = \frac{L - \sum_{i=1}^{L} \left| DMCode(i) - ECode(i) \right|}{L} \qquad (2)$$

where *DMCode* denotes the most commonly identified bit based on repetition and statistical mode.

We found that *Precision* and *CodecPrecision* results were much worse with real audio than we achieved in [1] using synthesized signals. This prompted further investigation to find the reasons for this.

By analysing the watermarked audio signals we found that, on occasion, components selected by the CSPE algorithm as candidates for modification at the encode stage were not subsequently identified during decoding. Since only the magnitudes of selected components were modified, it seemed likely that this modification was the reason for the inability of CSPE to identify the component after watermarking. However, the CSPE method does not rely on a component’s magnitude to identify it in a signal so it is likely that some other indirect result of modification affects CSPE’s identification of components in the watermarked signal.

As mentioned in Section II, the CSPE algorithm does not produce very accurate frequency resolution when components are close to each other spectrally. We therefore hypothesised that components identified by CSPE may become impossible to identify as a result of their magnitude being reduced and the subsequent masking of the reduced component by a spectrally close component. This concept is illustrated in Figure 3.

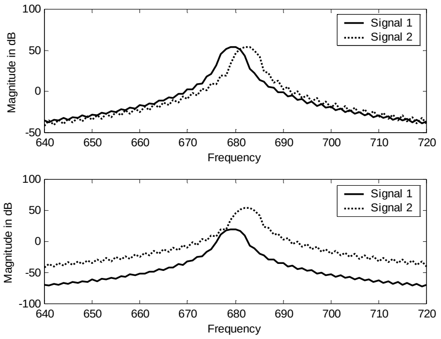

Figure 3: Illustration of masking phenomena

In the first sub-plot of Figure 3, we see two components of equal magnitude which are spectrally close together. When the magnitude of Signal 1 is iteratively reduced, its spectrum eventually falls below the spectrum of Signal 2, as shown in the second sub-plot. Signal 1 will then be unidentifiable by CSPE. This masking effect caused problems with decoding as the masked components could not be identified. This meant that comparison to determine if they represented a 1 or 0 bit was impossible.

### *(a) Using bin location as alternative criteria*

One problem with the method in [1], in which we reduced the magnitude of a candidate component to represent a bit, was that some components would be reduced to almost zero to satisfy the embedding criteria. These components might subsequently be impossible to detect in decoding. In this paper, by using frequency bin value as the criteria for modification, we can address this issue.

When the CSPE-identified component in the decode phase is different from the component identified in encode phase, its corresponding bin value also changes. We use this fact to our advantage by altering our selection criteria to ensure that bin values *(lbin* and *rbin* respectively*)* must now meet the following restriction in the embedding process:

$$|lbin\%2 - rbin\%2| = 0 \text{ where } code\ (i) = 0$$

and

$$|lbin\%2 - rbin\%2| = 1 \text{ where } code\ (i) = 1 \qquad (3)$$

where *code(i)* denotes the watermark bit to be embedded and *%* denotes the modulus operation.

By way of illustration, if the watermark bit is 0, it requires that the bin location of both components should be either both odd or both even (|*lbin%2 – rbin%2*| would return 0). If the embed bit is 1, then one bin location should be odd and one even (and |*lbin%2 – rbin%2*| would return 1).

### *(b) Component verification process*

As an additional pre-watermarking step to improve decoding accuracy, we analyse the proposed modifications of components at locations *lbin* and *rbin* to see if the modified components will still meet the condition set out in Eq.(3) after modification. If not, we repeat the modification of the components' magnitudes until the condition has been satisfied. This guarantees that these components will be identifiable by the CSPE process in decoding. By performing this step, we add a processing overhead to the embedding phase but since embedding is a one-off process and not time-critical it is a satisfactory compromise for improved accuracy. A further step was then added which served to reduce the number of iterations required to select components that satisfy Eq.(3), improving embedding efficiency and achieving 100% recovery.

With the modified algorithm, we performed the full encode-embed-decode cycle on 25 music files randomly selected from multiple genres. Watermark recovery (*Precision)* was 100% for each file. More encouragingly, *CodecPrecision* was almost 100%. *CodecPrecision* represents the accuracy of watermark recovery without the benefit of the statistical mode operation.

### *(c) Transparency and audible artefacts*

In a steganographic audio watermarking system, audible artefacts are unacceptable as they allow listeners to deduce that there might be a watermark present. One negative characteristic of our watermarking scheme at this point was the introduction of unexpected audible artefacts in the shape of random 'pops' or 'clicks'. We analysed various frames where these artefacts were present and noticed two distinct types of artefact with two distinct causes. We define the resultant artefacts as 'Type I' or 'Type II' clicks.

By way of illustration, the analyses of frames 82 and 313 from a watermarked signal are used here to demonstrate the phenomena. Figure 4 shows the spectral difference between the original, intermediate and watermarked signal of frame 82.

The solid line in Figure 4 represents the original signal, the dash dot line represents the signal after it has been modified once (denoted as 'intermediate signal'), while the dashed line represents the final signal, in which the selected bins satisfy the condition in Eq.(3).

In order to represent the watermark, the algorithm only reduces the magnitude of - at most - one component in any frame. Any difference between the modified signal and the original signal would therefore center on those bins in which the component has been changed. Recall from Section III(b) that the component to be modified may need more than one iteration to identify and watermark the appropriate component that satisfies Eq.(3). Two iterations were required in the example in Figure 4.

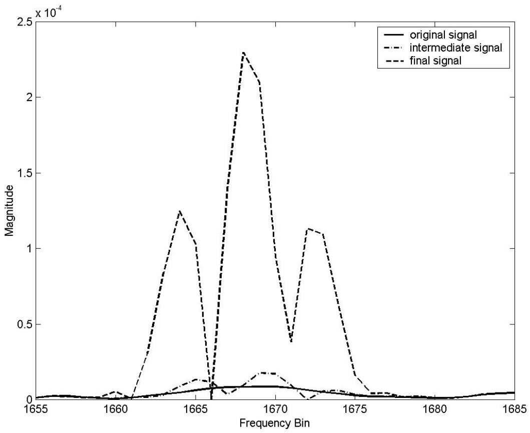

Figure 4: Spectrum of frame 82 for original signal, intermediate signal and final signal.

As can be seen from Figure 4, the selected component's magnitude in the final signal is apparently much larger than its value in the original signal and noticeably larger than neighboring components' magnitudes. Given that the algorithm only *reduces* amplitudes, this should not happen. By investigating this phenomenon, we found it occurs because CSPE sometimes identifies components which do not actually exist in the original. We term these 'ghost components'. When a 'ghost component' is identified by CSPE as a candidate for modification, our algorithm has the reverse effect, adding a new component into the signal. The added component causes the phenomenon we call a 'Type I' click. The phenomenon which causes the 'Type II' click is subtly different from the 'Type I' phenomenon. The spectrum of frame 313, shown in Figure 5, demonstrates the 'Type II' click.

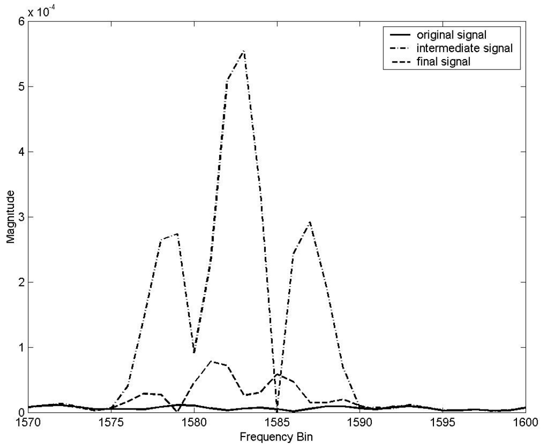

Figure 5: Spectrum of frame 313 for original signal, intermediate signal and final signal.

The cause of the 'Type II' phenomenon can be described as follows. As with the 'Type I' phenomenon, a 'ghost component' is identified by CSPE and reducing its magnitude results in adding this component into the signal. However, in this scenario, the component's bin location still does not satisfy the condition set out in Eq.(3) so its amplitude is further reduced. This should make its amplitude very low in the final signal. However, from the spectrum in Figure 5, we can see that the component's magnitude is not as low as expected. Conversely, it cannot be identified by CSPE and another component which meets the condition in Eq.(3) is then selected for modification. The artefact left behind by the modified 'ghost component' is audible. We call this phenomenon a 'Type II' click.

By repeated analysis of files which display audible artefacts, we found that all can be categorised into these two types. 'Type II' clicks occur very rarely, with average probability of 0.2%~0.4% (e.g. when embedding 1000 bits only 2~4 'Type II' clicks occur). 'Type I' clicks are more common, with probability sometimes above 1% (e.g. as many as 7 'Type I' clicks have been identified when embedding 672 bits). The number of 'Type I' and 'Type II' clicks depends on the components present in any frame and on the necessity or otherwise of modification of a component depending on the watermark bit. It is therefore impossible to predict where, or in what type of audio, either type of click may occur.

### *(d) Solving the audible artefact issue*

Perceptual transparency plays a more important role than accuracy of watermark detection in evaluating the performance of an audio watermarking scheme. Issues of audio quality, on the one hand, and transparency of the watermarking process on the other, must be addressed very carefully.

Since 'Type II' clicks occur only very occasionally, the recommended solution to this artefact is to return the component to its original state. This can result in a 0.1% reduction of the *Precision* but this is a small price to pay for increased perceptual transparency. For 'Type I' clicks, which occur more often, we have to find a more proactive solution to minimise what would be a much greater impact on the *Precision* value. As shown in Figure 5, with regard to 'Type I' clicks, the modified component has noticeably higher amplitude. In this case, we specifically remove this component by reducing its magnitude again.

In order to remove either type of artefact, we must add a step within the algorithm that first identifies whether they are 'Type I' or 'Type II'. We therefore designed the following rules based on listening tests and signal analysis:

1. If the modified component has a magnitude greater than 10 times the original component's magnitude, this is identified as 'Type I' click.

2. If the spectrum of the watermarked signal has bins with magnitudes that are different from the corresponding bins in the original spectrum, we pick out peaks of those differing bins from each spectrum. If a peak exists in the spectrum of the watermarked signal with a magnitude greater than 3 times the magnitude of the original corresponding peak, and if this peak is *also* greater than the magnitude of the neighbouring ((180*$F_s$)/$N$) Hz bandwidth of the original spectrum (where $F_s$ is sampling frequency and $N$ is window length), then this can be identified as a ‘Type II’ click.

A block diagram of the improved algorithm, which automatically detects, categorises and removes ‘Type I’ and ‘Type II’ artefacts, is shown in Figure 6.

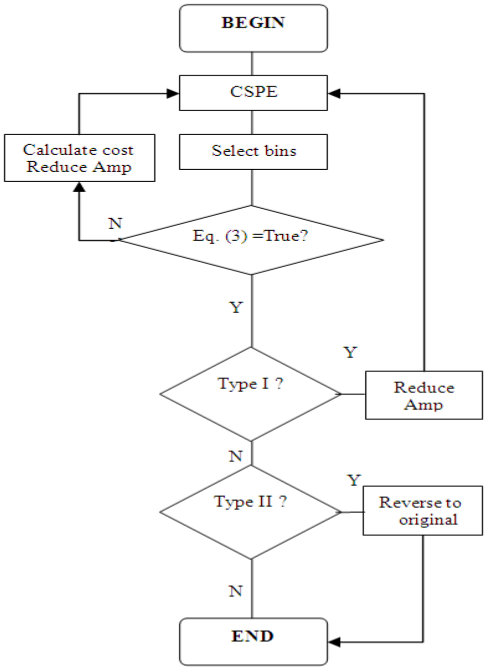

Figure 6: Flow chart of improved algorithm

### *(e) Watermark decoding procedure*

The watermarking scheme has been deliberately designed with a view to real-time decoding of watermarks. The decoding phase is very simple and computationally efficient, achieving real-time decoding of watermarked files. The flow chart of the decoding process is shown in Figure 7.

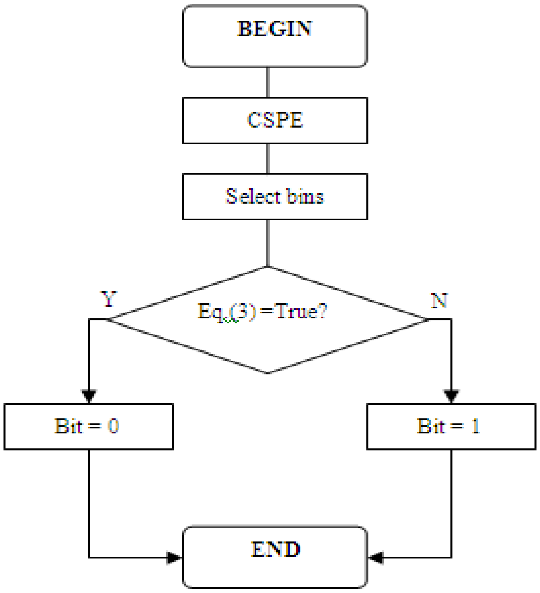

Figure 7: Watermark decoding process for each frame

## IV RESULTS

We randomly selected 25 candidate music files from different genres from a collection of 350 and performed the full encode-embed-decode cycle on each file. These files were all in 48000 Hz, 16 bit WAV format. In each case we determined the accuracy of the recovery of the watermark bit sequence in a single iteration as well as in repeated iterations. Recall from Section III that *CodecPrecision* and *Precision* are used to evaluate the performance of the algorithm.

The *CodecPrecision* value, for successfully decoding one single instance of the full watermark bit sequence from 25 files, was between 99% - 100%. The few errors are a direct result of the addition of the steps described in Section III(d) to remove audible artefacts. Such a low rate of error is a satisfactory compromise when the alternative is to have audible artefacts in the watermarked signal. The *Precision* value, calculated after repeated embed/decode and with the benefit of the statistical mode operation used as a means of error correction, was 100%. This means that every watermark was successfully decoded.

In order to evaluate the perceptual transparency of the scheme, we used ‘PQEvalAudio’ [10], which is an implementation of the ‘Perceptual Evaluation of Audio Quality’ (PEAQ) [11], to calculate the ‘Objective Difference Grade’ (ODG), between original and watermarked files. The ODG scale varies from 0 to -4, where 0 indicates that two compared files are perceptually identical and a score of -4 indicates that the differences between them are perceived to be ‘very annoying’. We compared 25 watermarked files against their unwatermarked counterparts. The average ODG score was 0.0467 meaning that there was almost no perceptual difference between them. The standard deviation of ODG score was 0.0782. The distribution of ODG scores is shown in Figure 8. Note that all of the ODG scores are consistently close to 0, meaning that the effect of watermarking is almost imperceptible.

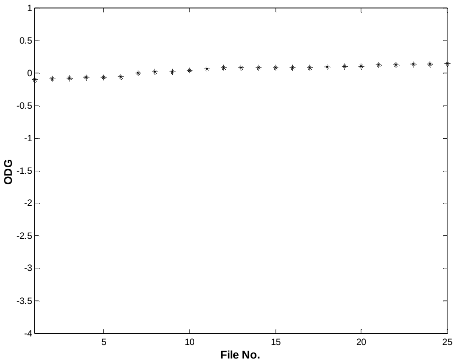

Figure 8: The distribution of ODG scores for 25 sample watermarked files

As an aside, more than half of our results from the PEAQ test were above 0. Note that scores above 0 are outside the defined range of the ODG scoring system. These anomalous results were therefore a cause of interest. We decided to perform the PEAQ test on 5 pairs of *identical* unwatermarked files and these uniformly produced results above 0. This suggests the PEAQ test evaluated some of our watermarked files as being perceptually identical to their original counterparts. This, of course, is a positive outcome.

## V CONCLUSION

We have presented an improved semi-blind audio watermarking scheme, capable of real-time decoding, that is near-perfect for use with real signals. The scheme is shown to be perceptually transparent. Watermark recovery was 100% for 25 randomly-selected sample music files. The scheme is designed to be adaptable to many applications in many domains as the watermark data has been made part of the cover signal rather than an addition to it. Therefore it is likely to be robust in most domains and open to further domain-specific or application-specific development to protect against attacks common to a given domain.

## VI REFERENCES

[1] J. Wang, R. Healy, and J. Timoney, 'Digital Audio Watermarking by Magnitude Modification of Frequency Components Using the CSPE Algorithm'. Proceedings of the China-Ireland Information and Communications Technologies Conference CIICT 2009, National University of Ireland Maynooth , 18th-21st August 2009.

[2] K. M. Short and R. A. Garcia, 'Signal Analysis using the Complex Spectral Phase Evolution (CSPE) Method', *Audio Engineering Society 120th Convention,* May 2006, Paris, France.

[3] Fujimoto, R., Iwaki, M., and Kiryu, T. 2006. 'A Method of High Bit-Rate Data Hiding in Music Using Spline Interpolation'. *Proceedings of the International Conference on intelligent information Hiding and Multimedia Signal Processing*. Pasadena, 18-20 December 2006, pp. 11-14.

[4] Fallahpour, M., and Megias, D., 'High capacity audio watermarking using FFT amplitude interpolation', *IEICE Electronics Express*, Volume 6, No. 14, pp. 1057 - 1063, June 2009.

[5] Yang, H., Sun, X. and Sun, G., 'A semi-fragile watermarking algorithm using adaptive least significant bit substitution'. *Information Technology Journal*, 2009.

[6] K. Gopalan, et al, 'Covert Speech Communication Via Cover Speech By Tone Insertion', *Proceedings of the 2003 IEEE Aerospace Conference*, Big Sky, MT, March 2003.

[7] Takahashi, A., Nishimura, R., Suzuki, Y., 'Multiple Watermarks for Stereo Audio Signals Using Phase-Modulation Techniques', *IEEE Transactions on Signal Processing*, Volume 53, No. 2, February 2005, pp. 806-815

[8] Goertzel, G., 'An Algorithm for the evaluation of finite trigonometric series', *American Math Month,* Volume 65, 1958 pp34-35

[9] J. Rauhala, H.-M. Lehtonen and V. Välimäki, 'Fast automatic inharmonicity estimation algorithm', *Journal of the Acoustical Society of America,* Volume 121, no. 5, pp. EL184-EL189, 2007.

[10] PQEvalAudio software for MATLAB from: http://www-mmsp.ece.mcgill.ca/Documents/Software/Matlab/PEAQ/PQevalAudio-v1r0.tar.gz

[11] D. Câmpeanu, A. Câmpeanu, 'PEAQ – An Objective Method To Asses The Perceptual Quality of Audio Compressed Files', *Proceedings of International Symposium on System Theory,* SINTES 12, October 2005, România, pp. 487-492