Perceptual Audio Coding

Angelos Kamariadis
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
angeloskamariadis@gmail.com

I. PSYCHOACOUSTIC MODEL 1

The purpose of the first part is to create a function that implements Psychoacoustic Model 1. The function will receive the windowed analysis frame as input, and will return the global masking threshold T_g.

A. Signal Preprocessing

Initially, the music signal was read using Python and converted from stereo format to mono. The standard method in digital signal processing for converting an audio signal from stereo format to mono is calculating the average of the two channels. That is, adding the left and right channels, and then dividing the sum by 2.

Subsequently, it was normalized over its entire length by dividing its samples by the absolute maximum value of the signal, so that throughout its duration it has values between $[-1, 1]$.

By dividing by `np.max(np.abs())`, we ensure that the highest peak of the signal will be 1 or -1, and all the remaining samples will be scaled proportionally.

Finally, we make the assumption that for all the following steps, the windowing of the signal will be performed using $N = 512$ samples per analysis frame.

B. Spectral Analysis

Our goal in this step is to obtain a high-resolution estimate of the signal's spectrum, expressed in SPL (Sound Pressure Level) units.

First, we define the Bark scale according to the following equation:

$$b(f) = 13 \arctan(0.00076f) + 3.5 \arctan\left[\left(\frac{f}{7500}\right)^2\right] \quad (1)$$

and subsequently, we calculate the N-point power spectrum $P(k)$ of the signal, with $N = 512$ samples, using the equation:

$$P(k) = PN + 10 \log_{10} \left| \sum_{n=0}^{N-1} w(n) x(n) e^{-j \frac{2\pi k n}{N}} \right|^2 \quad \forall 0 \leq k \leq \frac{N}{2} \quad (2)$$

where PN = 90.302 dB and is the normalization constant and $w(n)$ is the Hanning window which is defined as:

$$w(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right) \quad (3)$$

The results are shown below:

- **STEREO Size:** (725210, 2)
- **MONO Size:** (725210,)
- **Maximum and Minimum Value - Normalization:** 1.00 / -0.98
- **Frame Array Size:** (1416, 512) (1416 frames of 512 samples)
- **Array dimensions:** (1416, 257)
- **Maximum value (dB SPL):** 133.61
- **Minimum value (dB SPL):** -29.70

C. Detection of Tonal and Noise Maskers (Maskers)

Since we have calculated the power spectrum $P(k)$, we will then locate per critical band, local maxima (maskers) which are greater than their neighboring frequencies by at least 7 dB. The neighborhood range for calculating the maskers differs per discrete frequency k, as shown in the following equation:

$$\Delta_k = \begin{cases} 2, & \forall\ 2 \leq k \leq 63,\ (0.17 - 5.5 \text{ kHz}) \\ [2,3]63, & \forall\ 63 \leq k \leq 127,\ (5.5 - 11 \text{ kHz}) \\ [2,6]127, & \forall\ 127 \leq k \leq 250,\ (11 - 20 \text{ kHz}) \end{cases} \quad (4)$$

At high frequencies, the maskers cover wider neighborhoods.

The function

$$S_T(k) = \begin{cases} 0, & \forall\ k \notin [2, 250) \\ P(k) > P(k \pm 1)\ \wedge & \\ P(k) > P(k \pm \Delta_k) + 7\text{dB}, & \forall\ k \in [2, 250) \end{cases} \quad (5)$$

returns boolean values $\{0, 1\}$ that determine whether a tonal masker exists at position k. Each position k where S_T has a logical value of 1 corresponds to a tonal masker that covers the neighboring frequencies.

In order to find and represent the noise maskers, a precalculated array named `P_NM-26` was used.

Finally, the power of each tonal masker was calculated, using the equation:

$$P_{TM}(k) = \begin{cases} 10 \log_{10}\left[10^{0.1P(k-1)} + 10^{0.1P(k)} + 10^{0.1P(k+1)}\right] \text{ dB}, & \text{if } S_T(k) = 1 \\ 0, & \text{if } S_T(k) = 0 \end{cases} \quad (6)$$

The positions of the tonal maskers and their power spectrum are shown below.

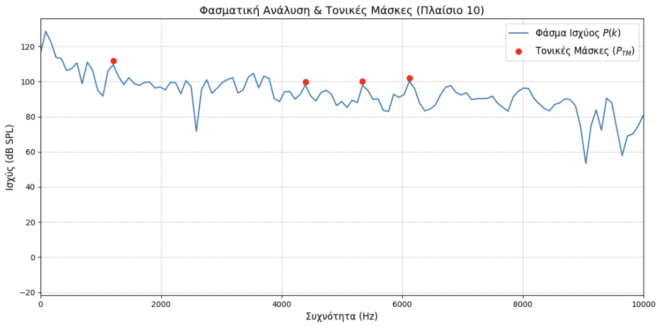

Fig. 1. Spectral Analysis & Tonal Maskers (Frame 10). The blue line represents the Power Spectrum $P(k)$, and the red dots indicate the Tonal Maskers P_{TM}.

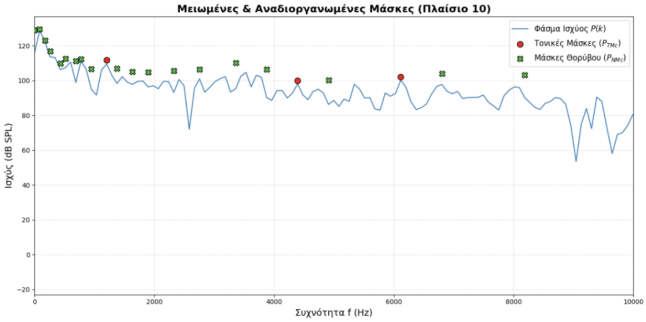

Fig. 2. Reduced Tonal Maskers (Frame 10)

Analyzing the visualization of frame 10, we draw the following conclusions:

Successful application of the 7dB criterion
At low frequencies (0 - 500 Hz) we observe that the spectrum exhibits its maximum energy since it exceeds 120 dB SPL. However, the algorithm did not place any red dot (tonal masker) there because that entire region has high energy. The local maximum does not stand out by 7dB from its neighboring frequencies. That is, it is not an isolated and clear tone, but a broader band of low frequencies. This fact proves that the code does not confuse simple local maxima with actual tonal maskers.

Detection of the actual tones
The algorithm successfully detected 4 tonal maskers in the 0 - 10 kHz spectrum, which are located (approximately) at the frequencies: 1.2 kHz, 4.4 kHz, 5.4 kHz, and 6.1 kHz respectively. These peaks, although they have lower absolute power (around 100-110 dB SPL) than the low frequencies, are very sharp, since they emerge abruptly from their surroundings, satisfying the +7dB rule.

Psychoacoustic interpretation
At the time point of 10 milliseconds (Frame 10), the human brain will focus its attention on these 4 dominant tones. Because these tones are so distinct and strong, they will cause masking in the neighboring, and weaker frequencies.

D. Reduction and reorganization of maskers

Now, extending the previous codes, we reduce the number of maskers, using the arrays pre calculated arrays `P_TMc-26` and `P_NMc-26` which contain the tonal maskers `P_TMc` and the noise maskers `P_NMc` for each analysis window, after the reduction and reorganization.

The reduced tonal maskers and noise maskers are depicted on the following graphical representation.

How many maskers are left and why
In the new graph, we observe that 3 out of the 4 maskers remain, as the masker located (approximately) at 5.4 kHz is gone. This happened because the reorganization algorithm determined that this peak was located very close to another stronger masker and therefore would be masked (overlapped) by it. Since our ear wouldn't hear it anyway, the algorithm deleted it to save unnecessary calculations.

Behavior of the noise maskers
We observe that the green x's are concentrated in regions of the spectrum that do not have sharp peaks. They essentially correspond to background noise or the broadband energy within each critical band. Finally, we observe that at very low frequencies (0 - 1000 Hz), where the signal has immense energy but no clear tones, we have numerous noise maskers, while at higher frequencies, they become significantly sparser.

E. Calculation of the two different masking thresholds (Individual Masking Thresholds)

After reducing the number of maskers, we calculate the two different masking thresholds, which represent the masking percentage at point i originating from the tonal or noise masker at point j. The thresholds are calculated as follows:

$$T_{TM}(i,j) = P_{TM}(j) - 0.275b(j) + SF(i,j) - 6.025 \text{ (dB SPL)} \quad (7)$$

$$T_{NM}(i,j) = P_{NM}(j) - 0.175b(j) + SF(i,j) - 2.025 \text{ (dB SPL)} \quad (8)$$

where $P_{TM}(j)$ and $P_{NM}(j)$ are the power of the maskers at point j, and $b(j)$ are the frequencies on the Bark scale.

The function $SF(i,j)$ calculates the extent of the masking from point j where the masker is located, to point i which is being masked, and for the case of tonal maskers, it is given in the assignment description.

Below, the effect of the spreading function is depicted, as well as the individual masking threshold.

Analyzing the above visualization, we draw the following conclusions:

Asymmetry of the umbrella
Observing the orange dashed line, we see that symmetry is absent around the red dot. At lower frequencies, the curve descends steeply. This means that our tone at 1.2 kHz has a hard time hiding sounds that are located at lower frequencies

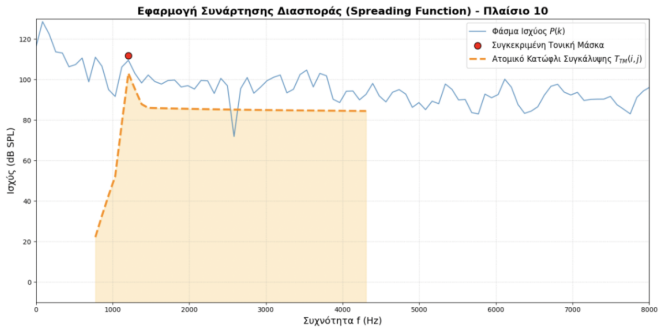

Fig. 3. Application of the Spreading Function (Frame 10). The blue line shows the Power Spectrum $P(k)$, the red dot indicates a Specific Tonal Masker, and the dashed orange line represents the Individual Masking Threshold $T_{TM}(i, j)$.

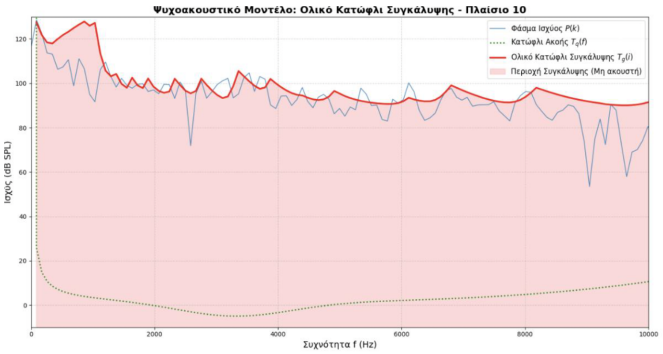

Fig. 4. Psychoacoustic Model: Global Masking Threshold (Frame 10). The blue line represents the Power Spectrum $P(k)$, the green dotted line is the Absolute Threshold of Hearing $T_q(f)$, and the thick red line is the Global Masking Threshold $T_g(i)$. The shaded red area represents the masked (inaudible) region.

than it. At higher frequencies, the curve fades very smoothly and spreads over a huge range, reaching almost up to (approximately) 4.3 kHz. Therefore, a strong bass or mid sound easily covers the more treble elements of the audio.

The ear's blind spot

The entire area with the pale orange color under the curve is practically useless information for this millisecond. Any peak of the blue line that happens to be lower than the orange line, a human is not capable of hearing.

The 8 Bark limit

At lower frequencies, the orange line is cut off abruptly. This is due to the limitation of the neighborhood Δ_b defined in the assignment.

F. Calculation of the Global Masking Threshold

The individual masking thresholds which were calculated previously, can be combined to create the global threshold, at each discrete frequency separately. The global threshold $T_g(i)$ for each analysis frame is calculated additively by the equation:

$$T_g(i) = 10\log_{10}\left[10^{0.1T_q(i)} + \sum_{l=0}^{255} 10^{0.1T_{TM}(i,l)} + \sum_{m=0}^{255} 10^{0.1T_{NM}(i,m)}\right] \text{ dB SPL} \tag{9}$$

where $T_q(i)$ is the Absolute Threshold of Hearing (ATH) at each discrete frequency i, and $T_{TM}(i, l)$ and $T_{NM}(i, m)$ are the individual masking thresholds of the tones and noise respectively, as calculated above. To calculate the individual masking thresholds $T_{TM}(i, l)$ and $T_{NM}(i, m)$ for each critical band, which corresponds to the different maskers, the individual thresholds are summed in each analysis frame separately.

The global masking threshold for frame 10 is depicted below.

From the above visualization, we draw the following conclusions:

The weakness of the human ear

We observe that a very large part of the blue line, i.e., the actual audio signal, is submerged within the red shaded area. All this energy, although it exists in the `.wav` file, is masked by neighboring, stronger tones. However, since humans cannot hear it, it means that the encoder can completely remove this information and we will notice absolutely no difference in sound quality.

Threshold of hearing

The green dashed line starts very high at low frequencies, so we need a lot of dB to hear a deep bass, but then it decreases sharply. Human evolution has made our ears hypersensitive in this region. That's why the threshold there falls even below 0 dB SPL.

Signal to Mask Ratio

If we look closely, only some sharp peaks of the blue line manage to pierce the red ceiling, that is, the global threshold, and come out into the clear air. We hear these frequencies crystal clear.

After all the above steps, we arrived at the final expression of the Psychoacoustic Model 1 function T_g, which is:

$$T_g(i) = 10\log_{10}\left[10^{0.1T_q(i)} + \sum_{l=0}^{255} 10^{0.1T_{TM}(i,l)} + \sum_{m=0}^{255} 10^{0.1T_{NM}(i,m)}\right] \text{ dB SPL} \tag{10}$$

II. TIME-FREQUENCY ANALYSIS WITH A BANDPASS FILTER BANK

In this part, we filter the signal windows $x(n)$ with the analysis filter bank $h_k(n)$ and the synthesis filter bank $g_k(n)$. The goal is to create a function that implements the process of the following figure:

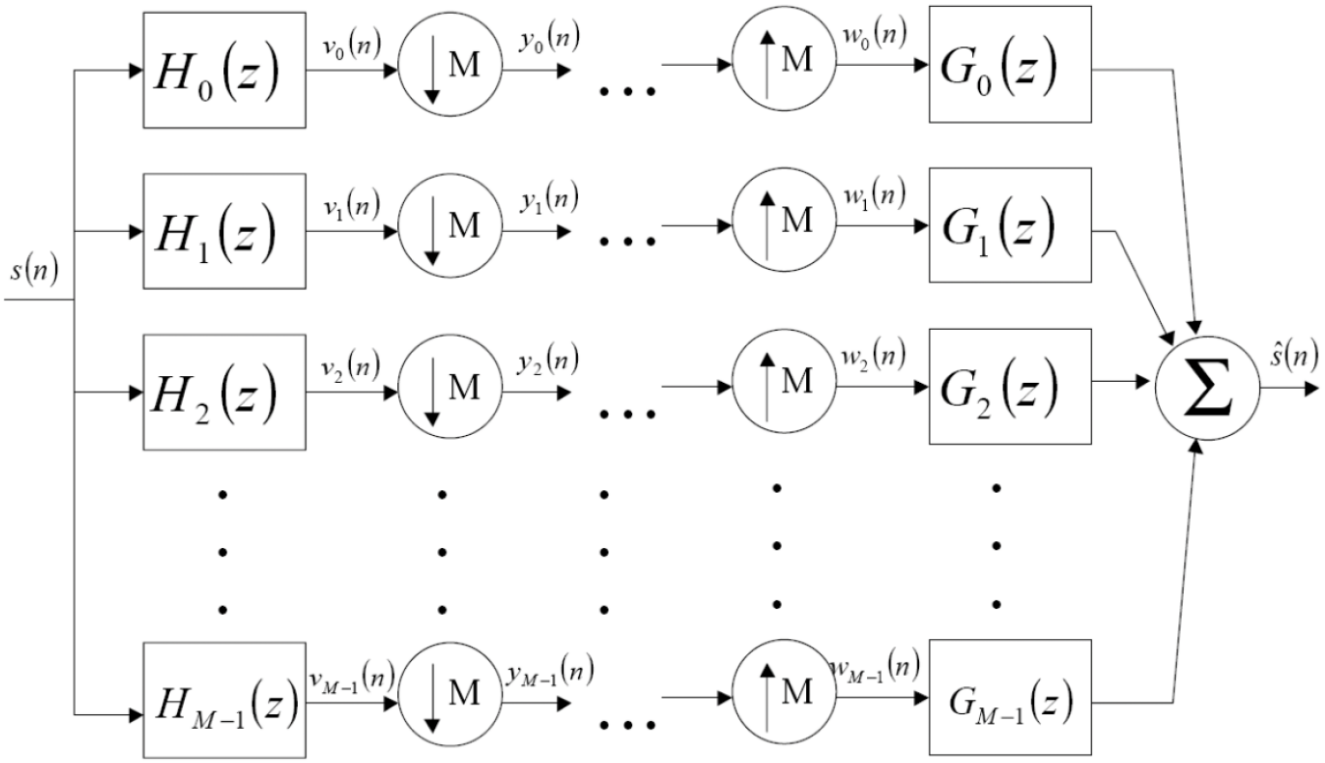

Fig. 5. Filter bank analysis and synthesis process.

which receives as input each analysis frame $x(n)$, the filter bank, and the global masking threshold that was calculated in Part 1. The output of the function is

the reconstructed signal $\hat{x}(n)$ and the number of bits used for its creation.

A. Bandpass Filter Bank (Filterbank)

For the analysis of the signal into its critical components per time frame, bandpass filter banks are used, as they are presented in the above figure. The filters were designed based on the Modified Discrete Cosine Transform (MDCT), which is fully invertible and introduces no errors in the signal coding.

In the encoding and decoding stage of the compression system implemented in the exercise, $M = 32$ analysis filters are used, whose impulse responses are given by the equation:

$$h_k(n) = \sin\left[\left(n + \frac{1}{2}\right)\frac{\pi}{2M}\right] \times \sqrt{\frac{2}{M}}\cos\left[\frac{(2n + M + 1)(2k + 1)\pi}{4M}\right] \tag{11}$$

and $M = 32$ synthesis filters whose impulse responses are given by the equation:

$$g_k(n) = h_k(2M - 1 - n) \tag{12}$$

B. Analysis with Filter Bank

In this step, we perform a convolution of the signal $x(n)$ with the analysis filters $h_k(n)$ and then, downsampling of the filtered signal. The output sequence is given by the equation:

$$u_k(n) = h_k(n) * x(n) = \sum_{m=0}^{L-1} x(n - m)h_k(m), \quad \forall k = 0, \ldots, M - 1 \tag{13}$$

Using Python, we implemented the subband analysis stage of the encoder, where the original broadband signal is separated into 32 narrowband signals through a bandpass filter bank (as depicted above) of cosine modulation (MDCT). Then, after designing the filters using a sinusoidal window to minimize spectral leakage, we applied linear convolution of the signal with them and proceeded to critical downsampling (decimation by a factor = 32), keeping the total volume of data constant without loss of information. Any aliasing errors from the non-ideal, overlapping bands are inherently handled through the TDAC (Time-Domain Aliasing Cancellation) property of the MDCT, which, as stated in the assignment, guarantees perfect reconstruction, ultimately resulting in a complete spectral decorrelation of the signal. This now allows us to apply independent, targeted quantization in each subband based on the global threshold of the psychoacoustic model. 32

analysis filters were created, of length 64 samples each. Output dimensions $Y_k(n)$: (32, 22665).

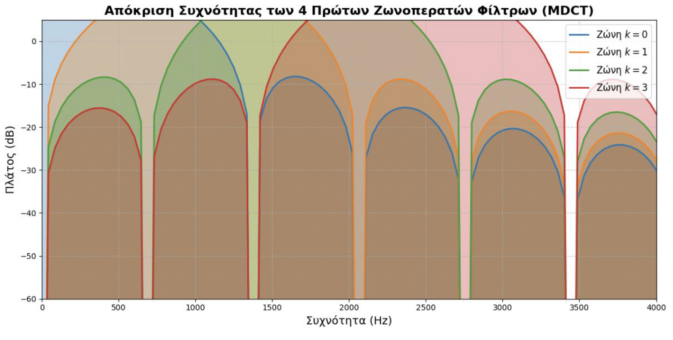

Fig. 6. Frequency Response of the First 4 Bandpass Filters (MDCT). The different colors represent the frequency bands $k = 0, 1, 2$, and 3.

From the above graph, we draw the following conclusions:

Separation into parts

Each color represents a different filter. Here we observe the first 4 out of 32 in total. As it appears, each filter has a peak which is located at a different frequency. The blue band takes over the very low frequencies. The orange one the immediately following ones. The green one the next ones, and so on. Essentially, the sound enters this system and is distributed into these successive channels.

Aliasing

If the filters were ideal, the graph would look like rectangular shapes where one would start where the other ends. However, as we see, the curves spread out and overlap with each other.

MDCT (Time-Domain Aliasing Cancellation)

As mentioned in the assignment, the MDCT transform is fully invertible. Even though the graph shows that our filters leak into each other, the MDCT algorithm, when it is time to reunite the sound (MP3), helps us to mathematically cancel out these errors (TDAC). Therefore, in practice, we do not care at all that the filters are not perfectly rectangular.

C. Quantization

In this step, we implemented the actual compression, that is, the data reduction of the audio signal, by connecting the filters with the Psychoacoustic Model of Part 1.

Every digital encoder has a quantizer, which maps the values of the samples of a discrete sequence to numbered quantization levels.

We implemented an adaptive uniform quantizer of 2^{B_k} levels, where B_k is the number of encoding bits per sample of the sequence $y_k(n)$ in the current analysis frame $x(n)$ of the signal, which is calculated by the formula:

$$B_k = \text{int}\left(\log_2\left(\frac{R}{\min(T_g(i))}\right) - 1\right) \tag{14}$$

where $T_g(i)$ is the global masking threshold of the psychoacoustic model for the analysis frame and $R = 2B$ is the number of amplitude levels of the original signal $s(n)$, encoded at B bits per sample. $T_g(i)$ is defined in the domain of the analysis filter which is given in the assignment description.

The quantization step Δ, is adjusted based on B_k and the respective range of values $[X_{\min}, X_{\max}]$ in the current analysis frame and is given by the equation:

$$\Delta = \frac{x_{\max} - x_{\min}}{2^{B_k}} \tag{15}$$

Finally, we experimented with a non-adaptive quantizer, compressing the signal using a constant number of $B_k = 8$ bits in each analysis frame, and a constant quantization step Δ for $[X_{\min}, X_{\max}] = [-1, 1]$.

For frame 10, the results are shown below:

- **Bit Allocation (Adaptive Quantizer):** [2 4 5 6 6 5 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 4 2 0 0 0 0]
- **Total frame bits (Adaptive):** 167 out of the original 512 (32x16)
- **Total frame bits (Constant):** 256 out of the original 512

From the above code, we confirm the value of compression. It proves that we can dramatically reduce the volume of an audio file without degrading its perceived quality. Specifically, we see the algorithm allocating 0 bits to very high frequencies, completely deleting information that is lost anyway below the threshold of human hearing, while in the audible bands it allocates the bits giving exactly as much precision as needed to hide the quantization noise from the dominant maskers. Thus, the adaptive quantizer achieves superior compression requiring only 167 bits. That is, 1/3 of the original size of 512 bits, surpassing by far in efficiency and acoustic logic the constant quantizer of 256 bits, which wastes space on frequencies that we practically do not hear.

D. Synthesis

In the final stage, the quantized sequences $\hat{y}_k(n)$ are sent to the decoder where they are interleaved with M zeros and upsampled, according to the relation:

$$w_k(n) = \begin{cases} \hat{y}_k\left(\frac{n}{M}\right), & \forall n = 0, M, 2M, 3M, \dots \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Subsequently, the sequences $w_k(n)$ are convolved with the synthesis filters:

$$g_k(n) = h_k(n)(2M - 1 - n) \tag{17}$$

The output $\hat{x}(n)$ of the filter bank is equal to the shifted input $x(n - n_0)$ if the quantization error is zero, which does not happen in most cases.

The final reconstruction of the music signal $\hat{s}(n)$, was done by applying the OverLap-Add technique without overlap between the successive analysis frames.

Thus, we achieved the implementation of the process of the filter bank analysis and synthesis.

Finally, we completed the compression cycle by implementing the decoder, where the mathematical integrity of our system was proven in practice. Through the upsampling of the quantized subbands and their convolution with the synthesis filters, we confirmed the critical TDAC (Time-Domain Aliasing Cancellation) property of the MDCT transform. This means that although the analysis had created aliasing between the channels, the synthesis filtering managed to mutually cancel out these errors algebraically. The final, comprehensive conclusion is that by summing the 32 reconstructed bands, we produce a new audio signal which, although it now lacks 2/3 of its original data due to the aggressive adaptive quantization, fools the human ear and sounds practically identical to the uncompressed original, thus sealing the absolute success of our psychoacoustic model.

III. CONCLUSIONS

Overall, taking into account the function of Psychoacoustic Model 1, we managed to build a fully functional perceptual audio coder, simulating the core of MP3 technology, combining mathematical signal processing with psychoacoustic theory. We completely removed the frequencies that a human cannot perceive, thus allocating the available space only to the critical audio information, that is, the ones that a human hears. Finally, we achieved a good data compression, while during the final synthesis stage we proved that we can reconstruct the song maintaining its perceived acoustic quality practically unaltered!

As proof of this, we present below a screenshot from our computer, which clearly shows the effective compression of the `music_dsp2026` file with the two different quantization methods (Adaptive Quantization and Fixed 8 Bit Quantization). We observe that the size of the file was reduced from 2.9 MB to 1.5 MB.

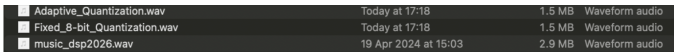

Fig. 7. Comparison of file sizes between the uncompressed original audio (`music_dsp2026.wav`) at 2.9 MB and the compressed outputs (`Adaptive_Quantization.wav` and `Fixed_8-bit_Quantization.wav`) at 1.5 MB.