Perceptual Audio Coding – MUSIC 422
Homework # 5

Handed out on: February 9, 2024
Due on: February 16, 2024 - 2:30 pm

The homework must be submitted using Canvas's assignment submission system (in the Assignments tab). Select the associated link in that tab and proceed with your submission (for additional information regarding file submissions for assignments on Canvas, read the page here)

You must **turn in 2 items**:

- **A write-up (with all the instructed answers and figures)** in PDF format. Scanned and typed submission are both acceptable.
- **All your code and data files in a single compressed** file (zip, tar or tar.gz) containing all the files requested in the homework instructions. The compressed folder should contain:
  - `bitalloc.py`

For your files, use the following **naming convention**: `hwX_suid.type` where X is the homework number and suid is your Stanford ID. For example, the write-up for homework 5 will be `hw5_mab.pdf` and the code files `hw5_mab.zip/hw5_mab.tar.gz/hw5_mab.tar`.

The complete homework policy is outlined on the course website.

Problem 1: Bit Allocation (40 pts.)

In this exercise, first you will compare bit allocation methods applied to the *test signal* studied in the exercise for Assignment #4 (where you determined the masked threshold and SMRs for this signal). The goal is to gain an appreciation of how different bit allocations sound. Next you will design and fine-tune an automated bit allocation routine based on the SMR values by testing it with different sound excerpts.

Reminder: As in homework 1, we use the convention 1kb = 1000b for data rates.

1.a) **(7 pts.)**

We want to encode the N/2 spectral lines coming from an $N = 1024$ MDCT applied to an input file sampled at 48 kHz. To do so, we use block floating point quantization, where each sub-block correspond to a Zwicker critical band, meaning that all frequency lines in a given critical band form a sub-block that shares a single scale factor.

Assuming a data rate of $I = 192$ kb/s/ch, answer the following:

i) If the input sound file is encoded as 16 bit PCM, **what is the compression ratio** for this data rate?

ii) At this data rate, **how many bits** do you have available to encode each block of $N/2$ spectral lines? **How many bits per frequency line** does this represent?

iii) If 4 bits are used to encode each scale factor, **how many bits** remain to encode mantissas? **How many bits per frequency line** does this represent?

iv) If, in addition to the scale factors, 4 bits are used for passing the bit allocation for each scale factor band and 4 bits are used for the block header (one block header per MDCT block), **how many bits** remain to encode mantissas? **How many bits per frequency line** does this represent?

1.b) **(3 pts.)**

Repeat question 1.a assuming now **a data rate of $I = 128$ kb/s/ch**.

1.c) **(10 pts.)**

We consider now the input signal used in Assignment #4 sampled at $F_s = 48$ kHz:

$$\begin{aligned} x[n] = {} & A_0 \cos(2\pi 220 n/F_s) + A_1 \cos(2\pi 330 n/F_s) + A_2 \cos(2\pi 440 n/F_s) \\ & + A_3 \cos(2\pi 880 n/F_s) + A_4 \cos(2\pi 4400 n/F_s) + A_5 \cos(2\pi 8800 n/F_s) \end{aligned}$$

where $A_0 = 0.40$, $A_1 = 0.24$, $A_2 = 0.18$, $A_3 = 0.08$, $A_4 = 0.04$, $A_5 = 0.02$.

Please quantize and inverse quantize the $N/2$ frequency lines of the $N = 1024$ MDCT of that signal (windowed using a *Sine or KBD* window) using block floating point. To do so, please use use 4 bits to encode each scale factor (with one scale factor per Zwicker critical band); assume that 4 bits are used to encode each scale factor band bit-allocation; assume also that 4 bits are used for the block header.

For $\underline{\text{the data rate of 128 kb/s/ch}}$, **please allocate** ($\underline{\text{by hand}}$ is OK) **all** the remaining bits as mantissa bits in the 3 following ways:

i) Uniformly distribute the remaining bits for the mantissas.

ii) Distribute the remaining bits for the mantissas to try and keep a constant quantization noise floor (assuming a noise floor 6 dB per bit below the peak SPL line in the scale factor band).

iii) Distribute the remaining bits for the mantissas to try and keep the quantization noise floor a constant distance below (or above, if bit starved) the masked threshold curve, i.e., try to keep a constant noise to mask ratio

Please report in a table the number of mantissa bits allocated for each critical band for each of the above bit allocation strategies. **Also, please submit 3 plots** (one for each of the above bit allocation strategies) showing the MDCT SPL, the Masked Threshold (you can use the code from Assignment #4) and the noise floor computed as the peak SPL value in that scale factor band minus 6dB per allocated mantissa bit (the noise floor can be plotted using, for example, matplotlib.pyplot.step) as a function of frequency (in log scale). Please **make sure you clearly indicate** which table values and plot correspond to which allocation strategy.

1.d) **(10 pts.)**

Repeat question 1.c (i.e., tasks i), ii) and iii)) for a data rate of 192 kb/s/ch. **Report** your results as a table. **Submit three plots** (one per bit allocation strategies) with the same quantities as above.

1.e) **(10 pts.)**

Modify the 3 functions `BitAllocUniform`, `BitAllocConstSNR`, `BitAllocConstNMR` in the provided skeleton file `bitalloc.py` with the same signature of the `BitAlloc()` function in `bitalloc.pyc` that returns the bit allocation values found above as hard-coded arrays.

Use your coder to **listen** to how they sound with the WAV file provided of the *test signal*. **Please answer the following:** How do the different bit allocations (for both data rates) compare with each other?
(**Note:** the 3 functions `BitAllocUniform`, `BitAllocConstSNR`, `BitAllocConstNMR` are not implemented in `bitalloc.pyc`.)

Problem 2: BitAlloc function (60 pts.)

2.a) **(20 pts.)**

Implement your own bit allocation routine `BitAlloc()` in the skeleton file `bitalloc.py`. You may use the strategy of your choice for the bit allocation, for example, you can use the water-filling algorithm (which is a good approximation of the optimization formula presented in the lecture). Please make sure that your function allocates bits to a set of sub-blocks covering the $N/2$ frequency lines of a N-window long MDCT block. The frequency lines in each sub-block will share a single scale factor and will all use the same number of mantissa bits. In your implementation, **make sure of the following**:

- Do not allocate **negative bits**;
- Do not allocate **a single bit**;
- Do not allocate **more than allowed maximum number of mantissa bits** per frequency line (e.g., 16 bits) to any scale factor band as specified in the input argument `maxMantBits`;

2.b) **(20 pts.)**

Incorporate your bit allocation in your coder. Use your bit allocation to reduce the data rate to 128 kb/s/ch and 192 kb/s/ch. **Report** in a table the bit allocation (i.e., the number of mantissa bits allocated for each critical band) you obtain when calling your function. **Submit a plot** showing the MDCT SPL, the Masked Thrshold, and the noise floor. **Please answer the following:**

- **Recall** the compression ratio necessary to compress a 16-bit encoded input sampled at 48 kHz to a data rate of 128 and 192 kb/s/ch. **Which value** should you set `codingParams.targetBitsPerSample` to achieve a data rate of 128 and 192 kb/s/ch for a 16-bit encoded input sampled at 48 kHz?
- For an input test file of your choice, **report** the size of the input file, the size of the compressed output PAC file and the corresponding compression ratio. **Are you getting the compression ratio expected?** (If not, please check/refine your bit allocation function until you are at least getting consistently close to the target compression ratio).

Note: See `pacfile.py` and `codec.py` if you need more information regarding the use of `codingParams.targetBitsPerSample`.

2.c) **(10 pts.)**

Spend some time tuning your bit allocation and SMR calculations to get the best sound you can out of the existing coder using a wide set of excerpts from the SQAM material including harpsichord, castanet, and german male speaker. **Describe briefly** your tuning attempts and their results.

2.d) **(10 pts.)**

Listen for coding artifacts you learned about in class (again using a wide set of excerpts from the SQAM material including harpsichord, castanet, and german male speaker) when varying the target bit rate. **Describe briefly** your observations.

Tip: It is often helpful to lower the bit rate until they are very audible and then listen carefully for them as you increase the bit rate. This training will help you hear artifacts you missed before.

CONGRATULATIONS! You now have a working coder – Review all the pieces of the coder. Carefully review the the specified bitstream format. Start thinking about what you would like to do to improve on the current coder as your class project. (Block switching? Stereo coding? Huffman/Arithmetic coding? SBR-type high-frequency enhancement? ML/AI-based Coding? Other ideas?)

Problem 3: Reading Assignment

Chapters 10 & 14 from the textbook, M. Bosi and R. E. Goldberg, "Introduction to Digital Audio Coding and Standards", KAP Springer 2003.

State, according to Stanford's honor code, whether or not you read the assigned chapter(s). In the next lecture you may or may not be asked to answer a simple question regarding the material presented in the reading assignment.