

Fundamentals of Perceptual Audio Coding

Craig Lewiston

INTRODUCTION

Conventional CD and digital audiotape (DAT) systems sample at 44.1 kHz using pulse code modulation¹ (PCM) with a 16-bit sample resolution. This results in a data rate of 705.6 kbits per second (kb/s) for a monaural channel or 1.41 Mbits per second (Mb/s) for stereo (Painter and Spanias, 2000). Though such high data rates are reasonable in audio applications such as CDs and DATs, Internet applications and wireless systems, subject to bandwidth constraints, cannot accommodate such high data rates. However, due to the market penetration of CDs and DATs, people have come to expect “high fidelity” from their audio systems. Therefore, considerable research has gone toward formulating compression algorithms that can satisfy the demand of low data rates without compromising reproduction quality. Collectively, these compression algorithms have been named *perceptual audio coders*. One example of such an algorithm is the Moving Picture Experts Group layer 3 or MPEG-1 layer 3, otherwise known as MP3.

How does MP3 work?

Traditional digital coding is waveform preserving, i.e., the amplitude vs. time waveform of the decoded signal approximates that of the input signal. The difference between input and output waveform is the basic error criterion of coder design.

The central objective of perceptual audio coders is different. Rather than favoring an output signal that faithfully preserves the input waveform, their error criterion favors an output signal that is useful to the human receiver. In short, to represent a signal with a minimum number of bits while producing an audio output at the desired fidelity.

When digitizing a signal, quantization noise is inevitably introduced. Although the outputs of perceptual coders contain considerable amounts of noise and distortion, the

¹ PCM (pulse code modulation) is a digital scheme for transmitting analog data. The signals in PCM are binary; that is, there are only two possible states, represented by logic 1 (high) and logic 0 (low). This is true no matter how complex the analog waveform happens to be. To obtain PCM from an analog waveform at the source (transmitter end) of a communications circuit, the analog signal amplitude is sampled (measured) at regular time intervals. The sampling rate, or number of samples per second, is several times the maximum frequency of the analog waveform in cycles per second or hertz. The instantaneous amplitude of the analog signal at each sampling is rounded off to the nearest of several specific, predetermined levels. This process is called quantization. The number of levels is always a power of 2 -- for example, 8, 16, 32, or 64. These numbers can be represented by three, four, five, or six binary digits (bits) respectively. The output of a pulse code modulator is thus a series of binary numbers, each represented by some power of 2 bits. At the destination (receiver end) of the communications circuit, a pulse code demodulator converts the binary numbers back into pulses having the same quantum levels as those in the modulator. These pulses are further processed to restore the original analog waveform.

noises and distortion are unperceivable to most human listeners. These algorithms reduced bit rates in large part by taking advantage of the human auditory system's inability to hear quantization noise under conditions of auditory masking (Pan, 1995).

Masking is a perceptual property of the human auditory system that occurs whenever the presence of one audio signal makes a temporal or spectral neighborhood of another audio signals imperceptible.

Motivation & Objective

No aspect of auditory psychophysics is more relevant to the design of perceptual auditory coders than masking, since the basic objective of perceptual audio coders is to use the masking properties of sounds to hide quantization noise.

In this lab you will have the opportunity to carry out some psychophysical measurements on yourselves and gain some “ear-on” experience with auditory masking. The experiments should be carried out in pairs, so you can take turns running the experiments.

GENERAL INFORMATION

The experiments take place in the sound booths in the middle of room.

The lab session is subdivided into two parts. In part one, you will be measuring the masking pattern associated with a narrowband noise. In part two, you will be measuring the masking thresholds in the presence of various masker types. The entire lab session will take approximately 2 hours.

The waveforms are created on a PC using Matlab. Sound is generated from those waveforms using a 24-bit digital-to-analog converter (DAC) in the PC. The electrical signal is then fed via a headphone buffer (TDT HB6) to the booth. In the booth, the stimuli are presented via Sennheiser HD580 headphones (located in the booth).

- Before you start the experiment, it is very important to make sure that the wiring and the attenuation settings are correct. Make sure the HB6 switch is set to 6 dB (‘up’ position).
- Log onto the computer.
- Start up **Matlab 6.5**.
- There should be a handheld voltage meter outside the booth. Use this to verify the voltage at both the left and right headphone amplifier outputs. To check the voltage enter:

```
> calibrate('mid',6)
```

This tells the system that you have 6 dB attenuation in the path (from the headphone amplifier).

- On the screen will then appear the voltage you should expect to measure at the output to the headphone buffer. Check that the actual voltage does not differ from the predicted voltage by more than about 10%. (Remember that a barely detectable 1-dB change is already 12%).

The next step is to enter this line of code:

```
> set(0,'RecursionLimit',775)
```

This line of code increases the recursive memory buffer size in MATLAB. **If you do not run this line, the experiment will never finish and you will not be able to record any data!**

MASKING EXPERIMENTS

To limit the scope of this lab, our focus will be on the subject of simultaneous masking. Simultaneous masking refers to the process by which the *simultaneous presence* of one sound (*masker*) elevates the threshold (changes the audibility or sensitivity) of another sound (*target*).

The hearing threshold in the presence of a masking signal is called the *masked threshold*. The *masked threshold* is the threshold intensity, I_T , of a target signal at frequency, f_T , in the presence of a masking stimulus with the intensity, I_M . When the masker intensity is set equal to zero, the masked threshold is just the probe intensity at the hearing threshold.

This lab will involve measuring a masking pattern and various masking thresholds. The lab is divided into two parts. Part one, will involve measuring the masking patterns for a narrowband noise centered at 1 kHz. Part two, will involve measuring the masking thresholds in the presence of different masker types.

Part 1: Masking pattern

Overview & Objective

In this part of the lab, you will measure your absolute hearing threshold in quiet and in the presence of a masker. Since measuring hearing thresholds for the complete range of audible frequencies can take a very long time, we will be using the Method of Adjustment, also known as the Békésy tracking method (after the famous scientist Georg von Békésy).

The Békésy tracking method works by repeatedly playing a target tone that is sweeping across a specified frequency range. The subject's task is control the loudness of the target tone by continuously pressing/releasing a button such that the tone is maintained at the just detectable level of hearing.

Stimuli

The frequency range for the target tone is 100 Hz to 8 kHz. The starting value for target tone intensity is 70 dB SPL.

The masker stimulus is a narrowband noise with a bandwidth from 950 to 1050 Hz and a spectrum level of 70 dB SPL.

The experimental parameter is the level of tones. The level is specified in dB SPL.

Method

Your task is to conduct 4 runs of the experiment for each member of your group (8 runs total for a group of 2). The first two runs will measure the hearing threshold in quiet (one run ascending frequency sweep, one run descending frequency sweep), and the second two runs will measure the hearing threshold in the presence of the masker (one run ascending frequency sweep, one run descending frequency sweep). The stimuli will be presented to only one ear. You have the choice of which ear to use. However, it is important that the same ear be used for all four runs for each subject. The different conditions that define these parameters (ear, quiet/masker, ascending/descending) are as follows:

- 1 – Left Ear, w/o Noise, Descending frequency sweep (high to low)
- 2 – Left Ear, w/o Noise, Ascending frequency sweep (low to high)
- 3 – Right Ear, w/o Noise, Descending frequency sweep (high to low)
- 4 – Right Ear, w/o Noise, Ascending frequency sweep (low to high)
- 5 – Left Ear, w/ Noise, Descending frequency sweep (high to low)
- 6 – Left Ear, w/ Noise, Ascending frequency sweep (low to high)
- 7 – Right Ear, w/ Noise, Descending frequency sweep (high to low)
- 8 – Right Ear, w/ Noise, Ascending frequency sweep (low to high)

So, for a person using their right ear, they would want to run conditions 3,4,7 and 8.

To start this experiment, enter the following line in the Matlab command window:

```
> bsy_main('Mask_pattern','xyz','mid','30','cond');
```

Note that all the arguments are in single quotes and are separated by commas. The first argument is the experiment name; the second is for your initials (e.g., 'jas' for John Adam Smith); the third is the booth name ('mid' for the MID booth, and 'front' for the FRONT booth), the fourth is the amount of attenuation set on the TDT PA4 (this needs to be set to '30' for all experiments), and the fifth is the condition under examination (see above for condition list).

Entering in the above command should result in a GUI response box appearing, which gives you instructions about what to do next. At the end of each run the screen will inform you to press “e” to end. After pressing “e”, you will need to start another run by entering the command line above, but will need to change the condition and/or subject.

The level of target tone should begin at an easily detectable level. By pressing on the *space bar* you can lower the level of the target tone. As the subject, your task is to maintain the tone at the just detectable level. Two repetitions for each ear will be run.

Data storage

The results from your experiment are stored in a file named “*mask_pattern_xyz_cond-1.dat*”, where *xyz* should be your initials and *cond* is the condition you ran. After you finish the first part of the experiment, you will want to copy these files to a floppy diskette for further analysis and lab write-up.

You can also perform a quick analysis (graph & average) of your data while you are in the booth. Once you have finished running the four runs for each subject, enter the following line:

```
> Plot_All_Rep
```

This program will take your four data files and graph the original data, then average the quiet runs and the masker runs and plot those averages. You will be prompted four times to select files for analysis. For the first and second file prompts, you will want to select the two quiet runs first, and for the fourth and fifth file prompts, select the two masker runs next. Four MATLAB plots will then pop up. **You can save each of these, and use them in your analysis.**

Part 2: Masking thresholds

Objective

In this part of the lab, you will be measuring the masking effect of different masker types in order to investigate the source of the asymmetry of simultaneous masking, as described in the lecture.

Stimuli

| Condition | Target | Masker |
|-----------|----------------|------------------|
| 1 | Gaussian noise | Tone (1 kHz) |
| 2 | Gaussian noise | Gaussian noise |
| 3 | Gaussian noise | Multiplied noise |
| 4 | Gaussian noise | Low-noise noise |

Gaussian noise was generated by digitally filtering a broadband Gaussian noise with a filter centered at 1 kHz with a bandwidth of 20 Hz.

Multiplied-noise is generated by multiplying a sinusoid at 1 kHz with a modulator. The modulator consisted of a low-pass Gaussian noise with a cutoff frequency of 10 Hz at an rms value of -10 dB (relative to amplitude 1) to which a dc component of value 1 was added (Dau *et al.*, 1999).

Low-noise noise is generated in a way described by (Kohlrausch *et al.*, 1997). It represents an efficient way of generating a bandpassed noise with a smoothed temporal envelope. The generation started with a Gaussian noise signal with a rectangular power spectrum. The following steps were iterated ten times: The envelope of the noise was calculated, representing the absolute value of the analytic signal, and the time waveform was divided by this envelope on a sample-by-sample basis and then restricted to its original bandwidth of 20 Hz by zeroing the corresponding components in the power spectrum. Iteration of the procedure leads to a decreasing amount of spectral splatter after each division by the envelope. The power spectrum within the passband is slightly different from that at the beginning of the iteration (for details, see (Kohlrausch *et al.*, 1997).

Method

To start this experiment, enter the following line in the Matlab command window:

```
> afc_main('MaskThresh', 'xyz', 'mid', '0', 'block1')
```

Note that all the arguments are in single quotes and are separated by commas. The first argument is the experiment name; the second is for your initials (e.g., 'js' for John Smith); the third is the booth name ('mid' in this case), the fourth is the amount of

attenuation set on the TDT PA4 (0 because the PA4s are not included in the circuit), and the fifth is the name of the data set you are collecting (leave this to block1).

Entering in the above command should result in a GUI response box appearing, which gives you instructions about what to do next. At the end of each run you have the option to start a new run, or to end the session. If you end or if you interrupt the program (by, for instance, closing the response window) in between runs, you can start where you left off simply by reentering the line given above – the program will know what conditions you still have to do by looking at the control file.

This is a 2-interval, 2-alternative forced-choice procedure. The signal level begins at what should be an easily detectable level and is varied adaptively according to a 2-down 1-up rule, tracking the 70.7% correct point on the psychometric function. The signal level is initially varied in steps of 8 dB. After the first two reversals, the step size is reduced to 4 dB. After a further two reversals, the step size is reduced to its minimum value of 2 dB. Each run is terminated after six more reversals, and the threshold is defined as the mean level at the last six reversals. Two repetitions of each condition will be run. The program randomizes the presentation order of the conditions.

Each trial consists of two 200-ms noise bursts. The task is to decide which of the two intervals also contains the target signal. All the stimuli are gated with 10-ms ramps to avoid spectral “splatter”.

Data storage

The results from your experiment are stored in a file named “*Mask_threshold_xyz_block1.dat*”, where *xyz* should be your initials.

DATA ANALYSIS

Once the data have been collected, you should copy your data file onto a diskette (*not provided*) and complete the analysis elsewhere. The full version of Matlab is available on Athena. The data analysis may be carried out in pairs, so if you have no experience with Matlab, team up with someone who knows about these things!

WRITING UP

Your lab report should describe the experiment and the results (include plots of the data), and should cover the following points:

- 1) Describe the fundamental concepts behind digital audio & perceptual audio encoders (e.g. quantization & quantization noise, sub-band coding & bit allocation, tone & noise masking thresholds, etc.).
- 2) Describe the methods of Experiment 1 and the results you obtained. Explain how the threshold results obtained relate to the masking thresholds used in perceptual audio encoding.
- 3) Describe the methods of Experiment 2 and the results you obtained, highlighting the amplitude and phase characteristics of the two “modified” noises used. Based on your data, indicate which component (amplitude or phase) contributes to the asymmetry of simultaneous masking observed.

The write-up should be done independently, although discussion in the pursuit of learning is highly encouraged (feel free to email and/or come by my office to discuss the lab). If you needed help with the analysis from another student, please state this in the lab report. It will not be held against you. However, the proper person/people should be acknowledged in your report.

REFERENCES

- Dau, T., Verhey, J., and Kohlrausch, A. (1999). "Intrinsic envelope fluctuations and modulation-detection thresholds for narrow-band noise carriers," *J. Acoust. Soc. Am.* **106**, 2752-2760.
- Kohlrausch, A., Fassel, R., van der Heijden, M., Kortekaas, R., van de Par, S., Oxenham, A. J., and Püschel, D. (1997). "Detection of tones in low-noise noise: Further evidence for the role of envelope fluctuations," *Acta Acustica* **83**, 659-669.
- Peter Noll, MPEG Digital Audio Coding Standards, Chapter in: IEEE Press/CRC Press "The Digital Signal Processing Handbook" (ed.: V.K. Madisetti and D. B. Williams), pp. 40-1 - 40-28, 1998
- Painter, T., and Spanias, A. (2000). "Perceptual coding of digital audio," *Proceedings of the IEEE* **88**, 451-513.
- Pan, D. (1995). "A tutorial on MPEG/audio compression," *IEEE Multimedia Journal*