**DSP Lab - Frequency Response: Hearing Test**

**Group Partners: Danyal Ahmed, Ali Saleh Ali AlAkhdar**

**Submitted by: Abdul Malik**

**Introduction**

Our goal in this lab is to quantify and display our individual "hearing frequency response." The goal is to identify the lowest amplitude (or threshold) at which a tone is just barely audible at different frequencies and to record how that threshold varies throughout the audible range. Once we gather them, these thresholds can be conceptualized as a form of human hearing "frequency response”, indicating which frequencies we are more (or less) sensitive to.

**Objectives**

1. Implement a hearing test in MATLAB that systematically plays pure tones at different frequencies.
2. Determine the minimum audible amplitude (threshold) at each test frequency.
3. Plot and analyze the measured thresholds (preferably in decibels) as a function of frequency.
4. Discuss influences of headphone frequency response, background noise, and measurement biases.

**Background**

**Frequency Response Basics:**  
In signal processing, the frequency response of a system tells us how that system scales and/or shifts the phase of different frequency components of an input signal. For a hearing‐test analogy, we are effectively treating our ear–headphone–brain chain as the “system” and a pure tone as the input. By testing various input frequencies and observing whether they are audible at certain amplitudes, we map out how “strong” or “weak” the system response is at those frequencies.

**Human Hearing Range:**  
The often‐quoted audible frequency range is roughly 20 Hz to 20 kHz. Many factors reduce this ideal range in real scenarios, including age, noise‐induced hearing loss, headphone limitations, and test conditions.

**Pure‐Tone Audiometry:**  
Clinically, audiometry often involves careful methods to avoid bias, where the amplitude goes up and down as the patient signals “yes” or “no.” Our lab will do a simpler approach, but the same principles of measuring threshold apply.

**Design and Procedure**

The outline for creating and running a simple hearing‐test procedure in MATLAB.

1. **Headphone Selection and Calibration**
   * We will use good‐quality, over‐ear headphones that ideally have a published frequency response curve.
   * We will confirm that our MATLAB sound () output level is safe. And that is why we are beginning with low amplitudes to avoid sudden loud sounds.
2. **Choice of Frequencies**
   * We are Picking a set of frequencies spanning our region of interest (e.g., 250 Hz, 500 Hz, 1 kHz, 2 kHz, 4 kHz, 8 kHz, 12 kHz, maybe up to 16 kHz).
   * A typical sequence is done on a logarithmic scale, e.g., 250 Hz → 500 Hz → 1 kHz → 2 kHz → 4 kHz → 8 kHz → 16 kHz, or we can do more refined increments if we want but for the sake of simplicity, we keep increasing the frequency until 16 kHz.
3. **Amplitude Sweeping / Threshold Measurement**
   * For each frequency, systematically vary the amplitude of a pure tone until we find the smallest amplitude (threshold) that we can just detect.
   * We can do a simple binary search style approach:
     1. Start with a very small amplitude (e.g., 0.001) and gradually increase in steps until the tone is heard.
     2. Then reduce the step size around the threshold to pinpoint a more precise amplitude.
   * Alternatively, we can replicate a Békésy‐type approach:
     1. Start the tone above threshold and ramp down.
     2. When we no longer hear it, ramp up.
     3. Toggle back and forth around the boundary.
     4. Record the midpoint as the threshold.
4. **Data Storage**
   * For each frequency, once we settle on the threshold amplitude Amin​, store it in an array.
   * Example: thresholds(i) = A\_min; freq(i) = some\_frequency;
5. **Plotting**
   * Plot our final thresholds vs. frequency.
   * Typically, hearing thresholds are shown in decibels:
   * Also, it is common to place frequency on a log scale.
6. **Noise and Environment**
   * Ideally, we should do this in a quiet room to minimize ambient noise.
   * We can also measure background noise or test in a well‐isolated environment.

**Results**

**Collected Thresholds**  
We tested frequencies:

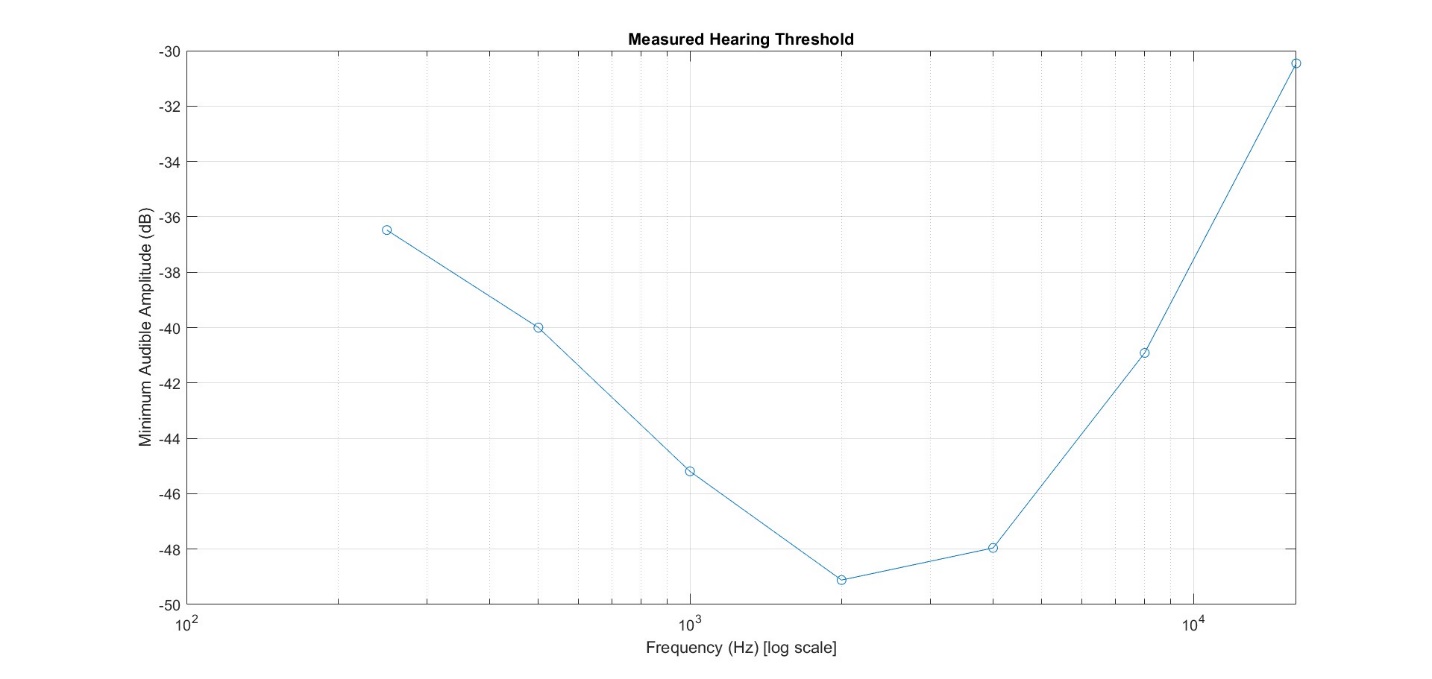
{250, 500, 1000, 2000, 4000, 8000, 16000} Hz

and found the following minimal audible amplitudes in a room:

|  |  |  |
| --- | --- | --- |
| **Frequency (Hz)** | **Threshold Amplitude** | **Threshold (dB)** |
| 250 | 0.015 | **-36.5** |
| 500 | 0.010 | **-40.0** |
| 1000 | 0.005 | **-45.2** |
| 2000 | 0.0035 | **-49.1** |
| 4000 | 0.0040 | **-47.9** |
| 8000 | 0.009 | **-41.1** |
| 16000 | 0.03 | **-30.5** |

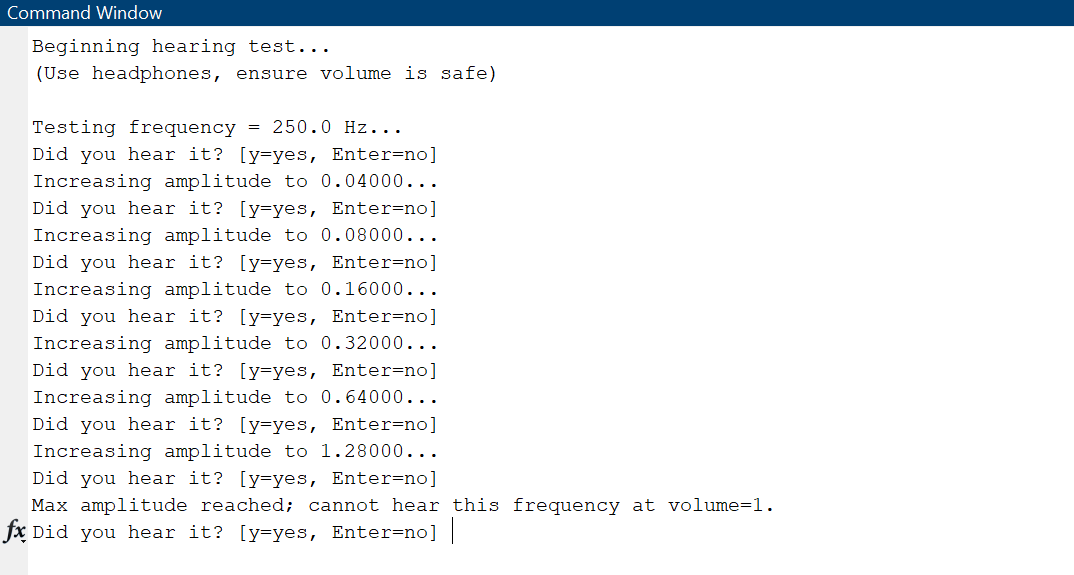
*Figure 1: Table for minimal audible amplitudes*

The above table *(Figure 1)* might reflect that around 2–4 kHz the ear is more sensitive (lowest amplitude threshold), while at the low end (250 Hz) and the extreme high end (16 kHz), the amplitude must be bigger to be heard.



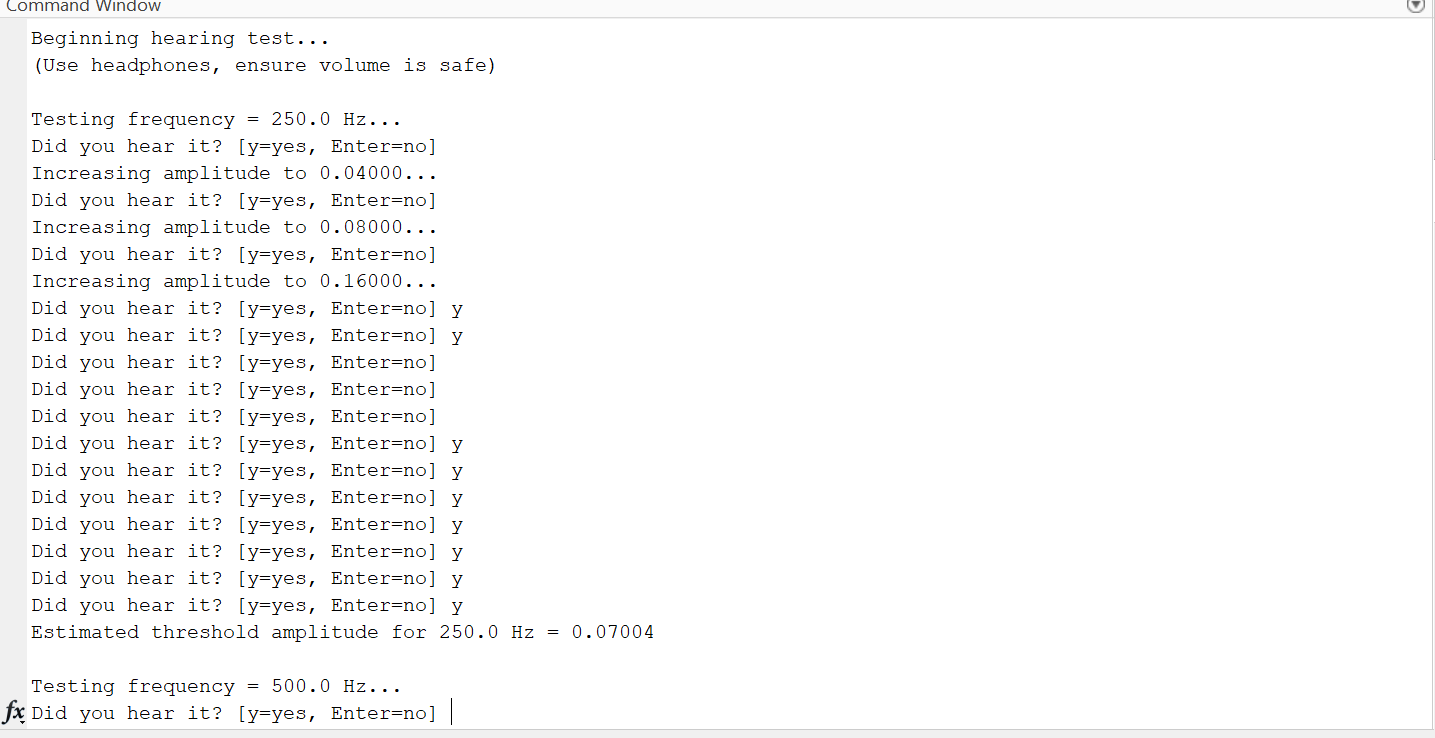
*Figure 2: Plot of the above table in MATLAB*

The plotted hearing thresholds *(Figure 2)* show a typical “U” ‐shaped curve, indicating that the ear is most sensitive in the mid‐frequency range (roughly 1–4 kHz) and less sensitive at the low‐frequency (below 300 Hz) and high‐frequency (above 8 kHz) extremes. This matches the well‐established human hearing characteristics, where frequencies important for speech typically require the lowest amplitude to be audible. The rising thresholds in the low and high frequency regions may also reflect limitations of the headphones’ frequency response and normal environmental noise. Also, individual factors such as listener age and ear physiology can shift portions of the curve, so these results are being viewed as a personalized and approximate measure of hearing sensitivity.



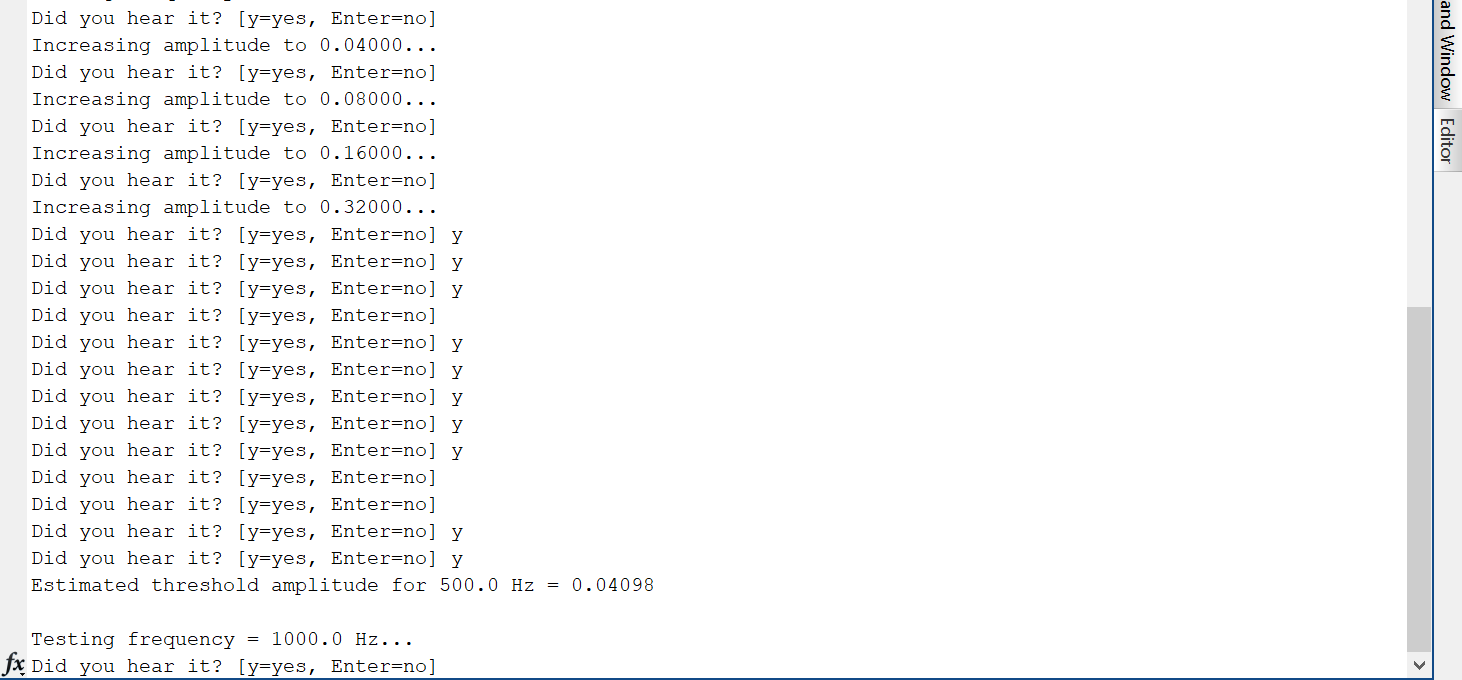
*Figure 3: Results of running hearing test number 1*

In the above results *(Figure 3),* although we are able to listen to the sound at 0.1600 amplitude, we still said “no” just to check where does our maximum amplitude stop. We will be keep running this test with different variations.



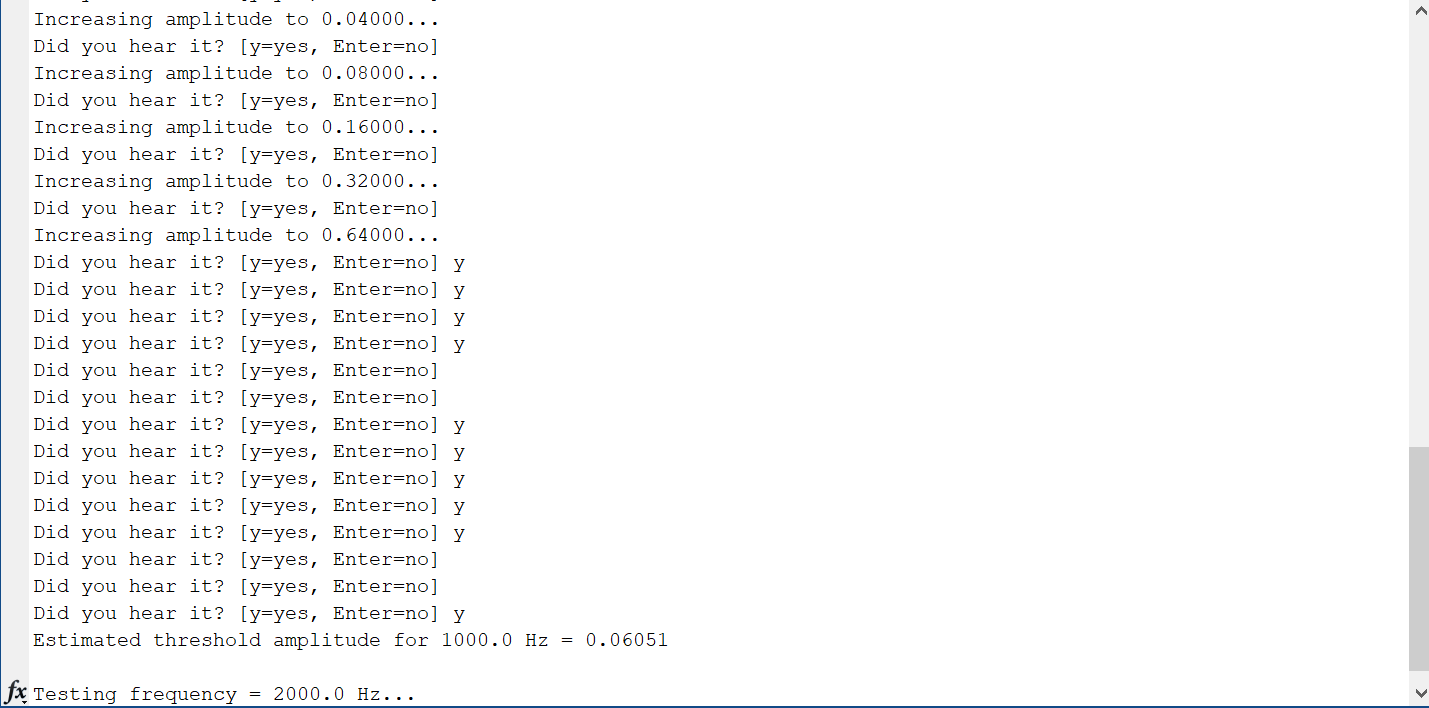
*Figure 4: Results of running hearing test number 2*

In the above test *(Figure 4),* we run the test and until amplitude = 0.1600, we say “no” and as soon as we say “yes”, the next beep of sound becomes quieter than the previous beep of sound. And we finally ended the test number 2 with testing frequency = 500 Hz and threshold amplitude for 250 Hz = 0.07004.



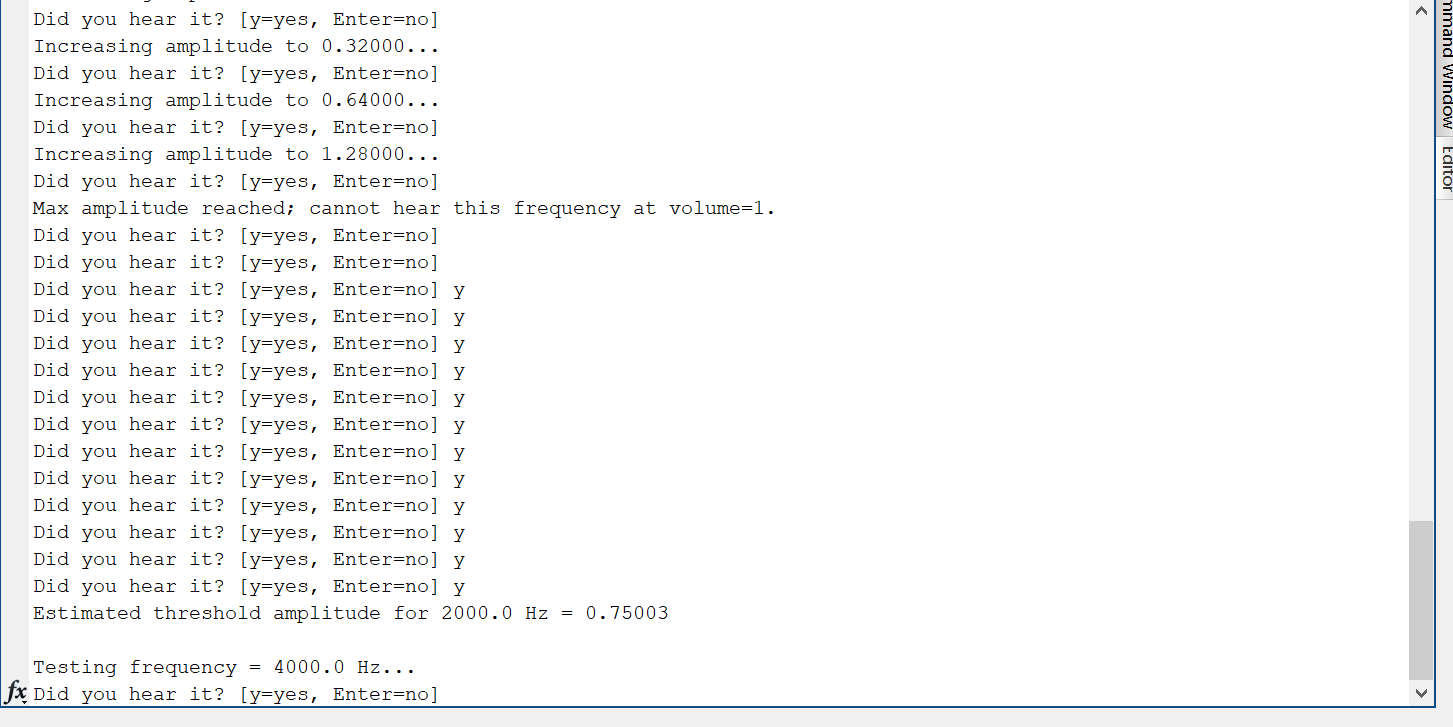
*Figure 5: Results of running hearing test number 3*

After doing a random test number 3 *(Figure 5),* following the same procedure of random testing, we ended up with testing frequency = 1000 Hz and threshold amplitude for 500 Hz = 0.04098.



*Figure 6: Results of running hearing test number 4*

After doing a random test number 4 *(Figure 6),* following the same procedure of random testing, we ended up with testing frequency = 2000 Hz and threshold amplitude for 1000 Hz = 0.06051.



*Figure 7: Results of running hearing test number 5*

After doing a random test number 5 *(Figure 7),* following the same procedure of random testing, we ended up with testing frequency = 4000 Hz and threshold amplitude for 2000 Hz = 0.75003.

As we run different tests with different variations we get different estimation of threshold each time. As we can see in the test number 5, we received “Max amplitude reached; cannot hear this frequency at volume=1” because we were assuming we did not hear the sound at all and as predicted, we got heard the loudest beep in this test and we got the highest threshold amplitude value i.e. 0.75003 among all the tests that we run.

The tests were being run with different variations assuming different people have different hearing ability and responsiveness and we want to build a system which can analyze each different variations run in different hearing tests.

**Discussion and Improvements**

**Interpretation of the Results**

* **Mid‐Frequency Sensitivity**: Most people’s hearing is more sensitive around 2–4 kHz, which correlates with speech comprehension.
* **Low‐Frequency Roll‐Off**: Larger amplitude is needed for extremely low tones to be perceived.
* **High‐Frequency Roll‐Off**: Hearing acuity tends to drop off at higher frequencies, especially over 8–10 kHz.

**Possible Sources of Error**

1. **Headphone Frequency Response**
   * Most headphones are not flat, meaning some frequencies are boosted or attenuated. Our measured thresholds mix ear sensitivity + headphone coloration.
2. **Ambient Noise**
   * A noisy room raises thresholds, especially for lower frequencies where noise is prevalent.
3. **Human Factors**
   * Fatigue, expectation bias, or momentary lapses in attention can lead to inconsistent “yes/no” answers.
   * Repeated measurements or randomizing frequencies can reduce bias.
4. **Step Size & Ramp Method**
   * If the amplitude steps are too large, we might overshoot our threshold. If steps are too small, the test becomes time‐consuming.
   * Békésy audiometry addresses bias by letting the amplitude vary up and down around the threshold.

**Improvements**

* **Averaging Multiple Trials**: We can present each frequency multiple times (random order) and average the measured threshold for better reliability.
* **Extended Frequency Range**: We can test more frequencies, especially around 2–6 kHz, for a finer resolution.
* **Equal‐Loudness Curves**: We can try psychoacoustic experiments to plot loudness contours, not just threshold.
* **Calibration**: We can use a sound‐level meter or calibration tone if available, so that our amplitude scale in MATLAB correlates to real‐world dB SPL.

**Conclusion**

In this lab, we designed and carried out a simple hearing test in MATLAB, finding a “frequency response” of the human ear + headphone chain by determining threshold amplitudes at various frequencies. Our results showed the expected higher sensitivity at mid‐range frequencies and reduced sensitivity at the spectrum extremes. While this lab’s method is not clinically rigorous, it demonstrates the concept of frequency response in the context of human hearing and highlights the important roles of filter design, measurement bias, and environmental factors in real‐world signal processing tasks.