“Hear here”

Engineering Audio Laboratory Session for Year 12/13

Prof David M Howard, Department of Electronic Engineering,
Royal Holloway, University of London

I have called this lab session **“Hear here”** because it provides a way to create sounds *here* on your computer that you can then *hear* allowing you to explore aspects of your hearing!

First, a safety note: it is essential that you are careful with sound levels, so you do not damage your hearing. Always start any listening with the volume set to a low value and increase it to a comfortable listening level when experimenting with sound.

The work set out below can be done on a PC or a Mac and it involves the use of the freeware program ‘Pure data’ (commonly known as ‘Pd’). Pd can be downloaded from: https://puredata.info/downloads/pure-data where you can select the PC or Mac version as appropriate.
Pd is a graphical programming language that is used by professional audio engineers and music technologists to create:
(a) electronic musical instruments for performance practice and music studios;

(b) special effect sounds for games and augmented/virtual reality; and

(c) test signals for measuring acoustics of spaces and hearing.

The exercises below relate to the exploration of human hearing.

Getting started …
Download and start Pd - the icon for Pd is shown in figure 1.

Figure 1: The screen icon for Freeware program Pd (Pure data)

We shall now explore the sinewave as a sound – it is the building block of audio.

Exercise 1: The sinewave

Once downloaded, start Pd by double clicking on the Pd screen icon. When the main window appears, select **<File><New>** (The way to read my instruction here is - go to the ‘File’ menu and select ‘New’) and an edit window should appear. This is where you work and text that you type is shown in these instructions in blue.

IMPORTANT NOTE: there are two modes of operation in Pd: the **Edit** and **Run** mode. You can tell which one you are in by observing the mouse cursor which changes between an arrow (run mode) and a pointing finger (edit mode). You switch between these either by pressing the **ctrl** and **e** keys together or via the menu as **<EDIT><Edit Mode>**. Check these out now and then go to the **Edit** mode (pointing finger cursor) now.

To create a simple sound, two Pd items are required: a sinewave oscillator and an output to your computer's loudspeakers. Select **<Put><Object>** and click in the Edit window to place the object.

Click within the object and in it type: osc~ 440 - notice that there is a space after 'osc~' and '440' – this is important as it separates the instruction (osc~) from its value (440). (Pd will indicate when it does not recognise something typed into an object by not placing a solid line around the object – do try for yourself.)

This creates a sinewave oscillator at 440 Hz – that is a sinewave that has 440 cycles per second which is the fundamental frequency of the note A above middle C on the piano. This is the note that orchestras tune their instruments to.

Click in the Edit window to place another object below the first and in it type: dac~. (This connects to the loudspeakers of your computer using your computer's 'DAC' or digital to analogue converter.)

To hear the sound, the two objects you have created need to be connected together. Notice a dark rectangle on the lower edge of the 'osc~ 440' object and two dark rectangles on the upper edge of the 'dac~' object (these connect to the left and right loudspeakers of your computer). Place the mouse arrow on the rectangle on the lower edge of the 'osc~ 440' object, click and drag to one of the rectangles on the upper edge of the 'dac~' object and release the mouse button. A connecting 'wire' should appear. Connect another wire from the 'osc~ 440' object to the other rectangles on the upper edge of the 'dac~' object. Two wires are used because the sound output on a computer is in stereo - Your Pd patch should look like that shown in figure 2.

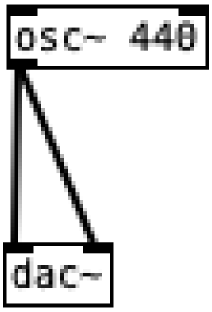


Figure 2: Pd patch for exercise 1.

To start and hear the sound, lower the volume on your computer's loudspeakers and run the patch by selecting **<Media><DSP on>**. You should hear the sound of a 440 Hz sinewave (note that the volume can be controlled with your computer's volume control and you will not hear anything if this is set to its minimum).

To stop the sound select **<Media><DSP off>**.
You have created and listened to a sinewave – the building block of audio.

Exercise 2: Changing the frequency of a sinewave

You are going to control the frequency (heard as a pitch change) of your sinewave using a horizontal slider. Select **<Put><Hslider>** and place it above the 'osc~ 440' object. Connect its output (the rectangle on its lower edge) to the rectangle top left of the 'osc~ 440' object which is its frequency input.

Before the slider can be used the range of its output values need to be changed because by default, the range is set as 0 to 1 which would be inaudible to the human ear. Place the mouse cursor on the slider and press **ctrl + click** on a Mac or **right click** on windows. Select '**Properties**' and you are going to change the '**Output Range**' values.

We want to set the frequency range from one octave below 440 Hz to one octave above 440 Hz (the musical interval of one octave is achieved by doubling the frequency to go up by an octave and halving the frequency to go down by an octave). Go ahead and change the 'Lower' value to an octave below 440 and the 'Upper' value to an octave above 440. (If you are not sure what values to use, check out the appendix at the end of this lab sheet.)

Finally, we want to know the frequency of the sinewave we are listening to. Select **<Put><Number>** and place it below the slider. Connect a wire from the slider output to the number input. Your Pd patch should look like that shown in figure 3.

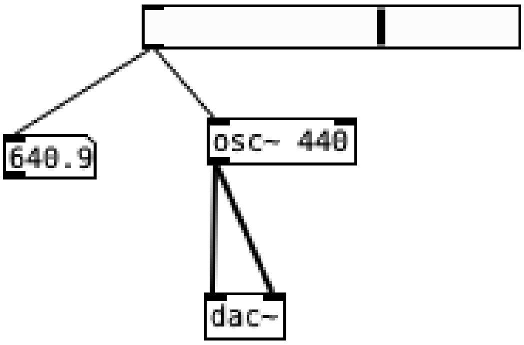


Figure 3: Pd patch for exercise 2.

To run the patch, you should be in **Run** mode which allows the slider value to be changed (recall that you can toggle between run and edit modes using **Ctrl + e**).

Run the patch to test the result (by selecting **<Media><DSP on>**). Listen to what an octave sounds like by setting a start frequency whilst listening to the sinewave and then moving to an octave above by moving the slider. The upwards/downwards pitch change of one octave occurs whenever the frequency of a sinewave doubles/halves.

Exercise 3: Beats

Beating is what you hear when 2 sinewaves are close in frequency (tuning a musical instrument involves reducing beats to zero which means that two sounds have the same frequency).

Starting with the complete patch created in exercise 2, set its range from 410 to 470. Add another sinewave at 440 Hz. We have to do something else to prevent the output signal from distorting – since we are adding two sinewaves together their sum can be double what the output will allow. Therefore, we need to multiply each sinewave output by 0.5. This is done in an object (**<Put><Object>)** with the contents *~ 0.5 which goes between the osc~ and dac~ objects as shown in figure 4. (To delete a connecting wire, click on it and press 'delete'.)
See figure 4 for the final Pd patch.

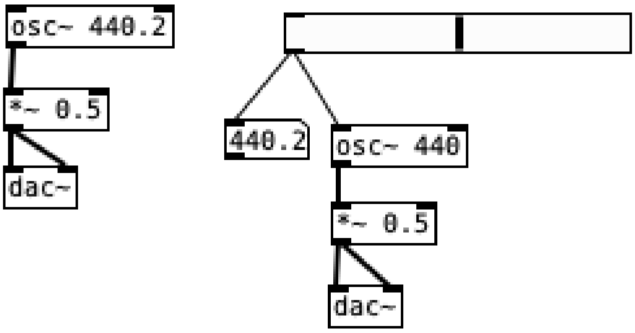


Figure 4: Pd patch for exercise 3.

Run the patch and adjust the slider to find the frequency nearest to 440 that it produces (in my case this was 440.2). Change the value of the fixed-value sinewave (the one without the slider) to that obtained from the slider so that the two frequencies are the same.

Use the slider to adjust the frequency of its sinewave to be close to 438 Hz and listen. Can you hear a regular undulation known as 'beating'? Note the frequency vales of the two sinewaves – the beating should be occurring at the difference frequency.

Use the slider to adjust the frequency of its sinewave above and below 440 Hz until the beating is 'lost' and the two sinewaves become 'fused' into a single 'fluttering' sound (it is not easy to describe this change – have a go and listen to what you hear). Note the frequencies (above and below 440 Hz) at which this happens for you. Note that there is no 'right' answer to this as we all hear slightly differently. **You have done a psychoacoustic (hearing) test.**

Exercise 4: Human hearing range (linear)

Our hearing has a nominal range of 20 Hz to 20,000 Hz (or 20 kHz). We can use Pd to hear sounds across our hearing range. Close your Pd patch from exercise 3 and create the Pd patch shown in figure 5. Set the frequency range slider from 20 Hz to 20 kHz and set its **Width** to 350 and its **Height** to 30 (this is to give more control of this wide frequency range). The **number object** (it is showing a value of 15123 in figure 5) needs to have a greater width to display 20000 – right click on the **number object**, select **Properties** and set the **Width** to 8. You Pd Patch should look like that shown in Figure 5.

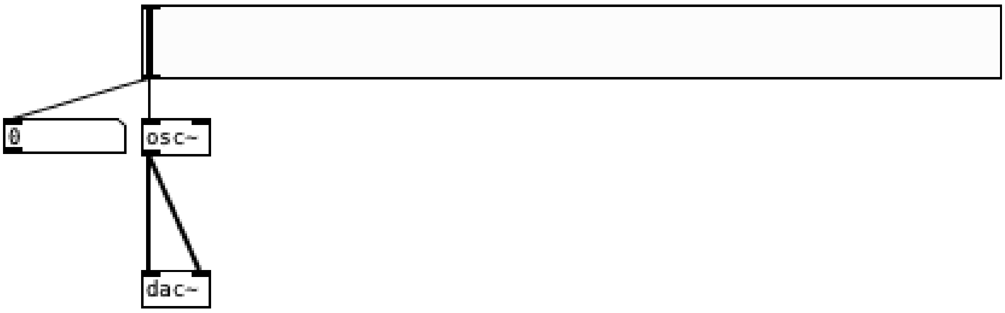


Figure 5: Pd patch for exercise 4.

Run the patch and find the range of frequencies between 20 Hz and 20 kHz that you can hear.

Please note that this is NOT a clinical measurement, you are not in a sound-treated quiet room and you are very unlikely to be using loudspeakers/headphones that operate over the full hearing range. Any hearing concerns you might have should be investigated by an audiologist health professional who will have professional audiological hearing test equipment available.

Exercise 5: Human hearing range (logarithmic)

Notice how the frequency changes when you move the slider in the previous exercise; this is frequency on a ***linear*** scale. The human ear hears frequency on a ***logarithmic*** scale which means that multiplying frequencies by a specific value gives equal steps for hearing. You have already explored this when listening to octaves in exercise 2.

We can experience listening on a logarithmic scale which models how the human ear works in terms of frequency. Add another slider to the Pd patch in figure 5 that has the same parameters as your existing slider (quick way to do this: in edit mode, click on the slider and then 'copy' <**CTRL+C**> and paste <**CTRL+V>**). Then change the slider **Parameter** from **lin** to **log** by clicking on **lin** (note that it changes to 'log for 'logarithmic) and set the **label** to 'logarithmic'. Open the other slider and set its **label** to 'linear'. Add another **number** object with a width of 8 and connect your log slider to it. Finally connect your logarithmic slider to the **osc**. It is fine to have two inputs connected to osc which will update to the last change made. Your patch should look like that shown in figure 6 (the items have been around to keep them tidy – click to select one item or 'rubber band' to select more than one item which can then be moved with the mouse).

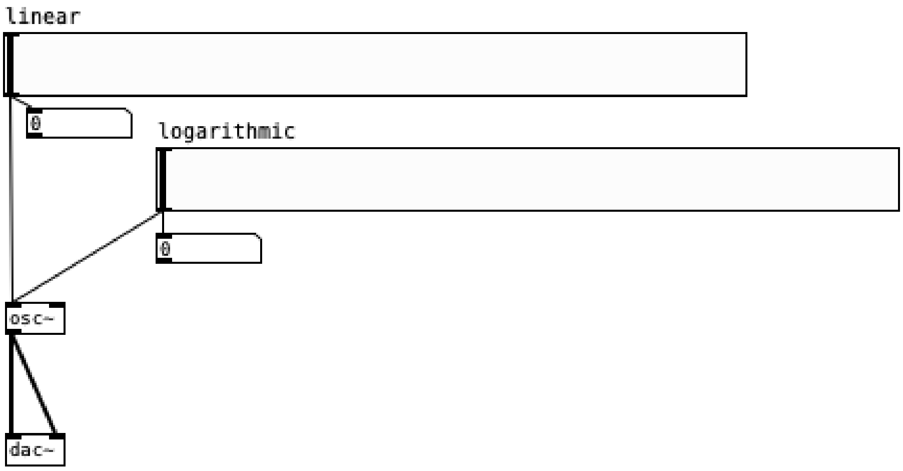


Figure 6: Pd patch for exercise 5.

Move each slider in turn and notice how the sound you hear and the displayed frequency values change for the linear and logarithmic sliders. In particular, notice how the low frequency values are 'spread out' for the logarithmic slider. Note how you can hear the sinewave across most of the range of the logarithmic slider but not the linear slider.

If you are familiar with octaves, listen out for how they vary on the linear and logarithmic scales and compare with the linear and logarithmic octave plots starting at 20 Hz in figure 7.

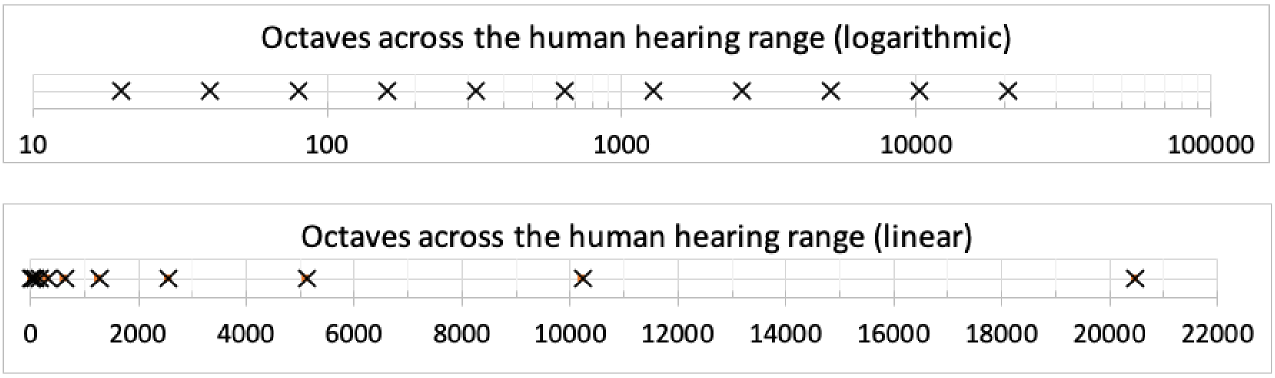


Figure 7: Octaves from 20 Hz on logarithmic (upper) and linear (lower) scales across the human hearing range (20 – 20,000 Hz).

Exercise 6: Musical chord (major triad)

Musical chords are made up of more notes whose frequencies are set to appropriate ratios to each other. We have already met the octave where the frequency ratio is a doubling to go up by 1 octave. A major triad has three notes and the frequencies of the upper two are tuned to 1.25 (major third) and 1.5 (perfect fifth) times the lower note frequency (see appendix 2 for an explanation of these multiplication factors). Notice that the **tilde** symbol (~) is not used in this case because we are not multiplying the waveform itself. (Pure data shows audio signal connections with thick connecting lines such as those below each **osc~** item in figure 6 and thin lines elsewhere, for example the four lines from the slider in figure 6, where the data update rate is much slower.)

Close your Pd patch from exercise 5 and create the Pd patch shown in figure 8. Create the three note Pd Patch shown in figure 5 noticing that the three output sinewaves are each multiplied by 0.3 (the *~ 0.3 objects) to ensure the overall output is not greater than 1 which could cause output distortion. Set the range of the frequency slider from 220 to 880.

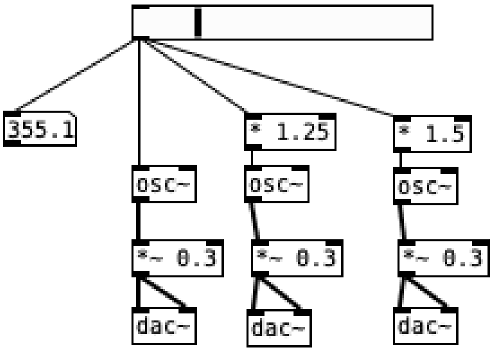


Figure 8: Pd patch for exercise 5.

Run the patch and observe the effect of what musicians would call a 'glissando major triad'. Change the major third to a minor third, for which a multiplying factor of 1.2 is used, and listen to the result. (see appendix 2 for guidance)

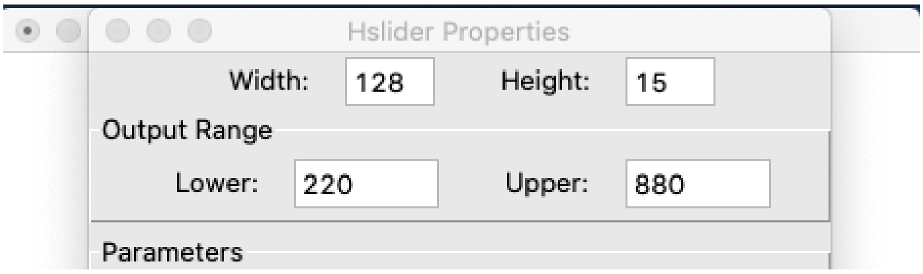
Troubleshooting tips:

- Have you left a space between * or ~ and the number input? If not, then the box will not be complete and so you cannot add any "wires" to other objects.
- Can't find where to change the slider frequencies? Right click on the slider and select "Properties". There you will see "Output range", and you can change the lower and upper limits for the slider.
- Can't move the slider? You need to be in **Run** mode to operate the slider (recall that you can toggle between run and edit modes using **Ctrl + e**).
- Can't read the output frequency of the slider? Right click on the box and choose properties. Then, change the width to 8 to make sure you can read it all the way up to 20,000Hz.

APPENDIX 1:

Setting the frequency values for Exercise 2

The '**Output Range**' values for the slider should go from one octave below 440 Hz to one octave above 440Hz (the musical interval of one octave is achieved by doubling the frequency to go up by an octave and halving the frequency to go down by an octave). The lower frequency will be half of 440 (i.e. 220) and the upper frequency will be double 440 (i.e. 880). The slider setting should look like figure Ap1.1.


Hslider Properties

Width: 128 Height: 15

Output Range

Lower: 220 Upper: 880

Figure Ap1.1: Settings for the slider in exercise 2.

APPENDIX 2:

Frequencies of tones in the harmonic series

Different sounds that have a musical pitch associated with them are made up of a number of sinewaves, but in terms of frequency, it is a restricted set of sinewaves that might be used. The frequencies of the sinewaves that make up a pitched note have to be integer multiples of the so-called 'fundamental frequency' which is the lowest frequency component. If the full set of possible sinewave components (the full set is commonly found in the output from musical instruments but with varying levels which is the basis on which we hear the differences between instruments playing the same pitched note), then for the tuning note A above middle C (fundamental frequency 440 Hz), the frequencies of its component sinewaves will be: (n * 440) Hz where n is an integer (1, 2, 3, 4, 5 ..). These sinewave components are referred to as harmonics which are numbered by their n value. So, for the A above middle C, the harmonic numbers, fundamental frequencies and musical interval above the fundamental are shown in figure Ap2.1.

harmonic	fundamental frequency	musical interval above fundamental
1	440 Hz (1 * 440)	-
2	880 Hz (2 * 440)	octave
3	1320 Hz (3 * 440)	octave + fifth
4	1760 Hz (4 * 440)	2 octaves
5	2200 Hz (5 * 440)	2 octaves + major third
6	2640 Hz (6 * 440)	2 octaves + perfect fifth
7	3080 Hz (7 * 440)	2 octaves + flat seventh
8	3520 Hz (8 * 440)	3 octaves

Figure Ap2.1: Harmonics, fundamental frequencies and musical intervals of the harmonics above the fundamental for the note A above middle C.

It is possible to calculate the frequency ratios for the major and minor triads from figure Ap2.1. For example, the major third is the interval between the 4th and 5th harmonic, so its frequency ratio is 5/4 = 1.25 (note that this value is used in Figure 8 for Exercise 6). The minor third is the interval between the 5th and 6th harmonic
so its frequency ratio is 6/5.

APPENDIX 3:

Taking Pd further:

The Pd manual is available under **<Help><Pd Help>**. A set of tutorials and other information is available under **<Help><Browser…>** and click on **"Pure Data/"** and explore the examples etc.

Further reading:

Howard, D.M. and Angus, J.A.S. (2014) Acoustics and Psychoacoustics,Focal Press, 5th Ed.

Puckette, M. The theory and technique of electronic music, http://msp.ucsd.edu/techniques/latest/book.pdf

Books about Pd: https://puredata.info/docs/BooksAboutPd/

David M Howard
Department of Electronic Engineering
Royal Holloway, University of London