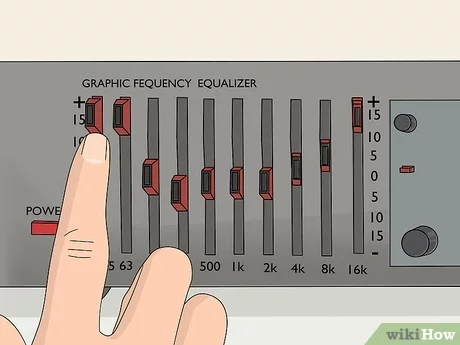
**Sound design**

**Lab 5: Audio Effect Programming**

**Background:**

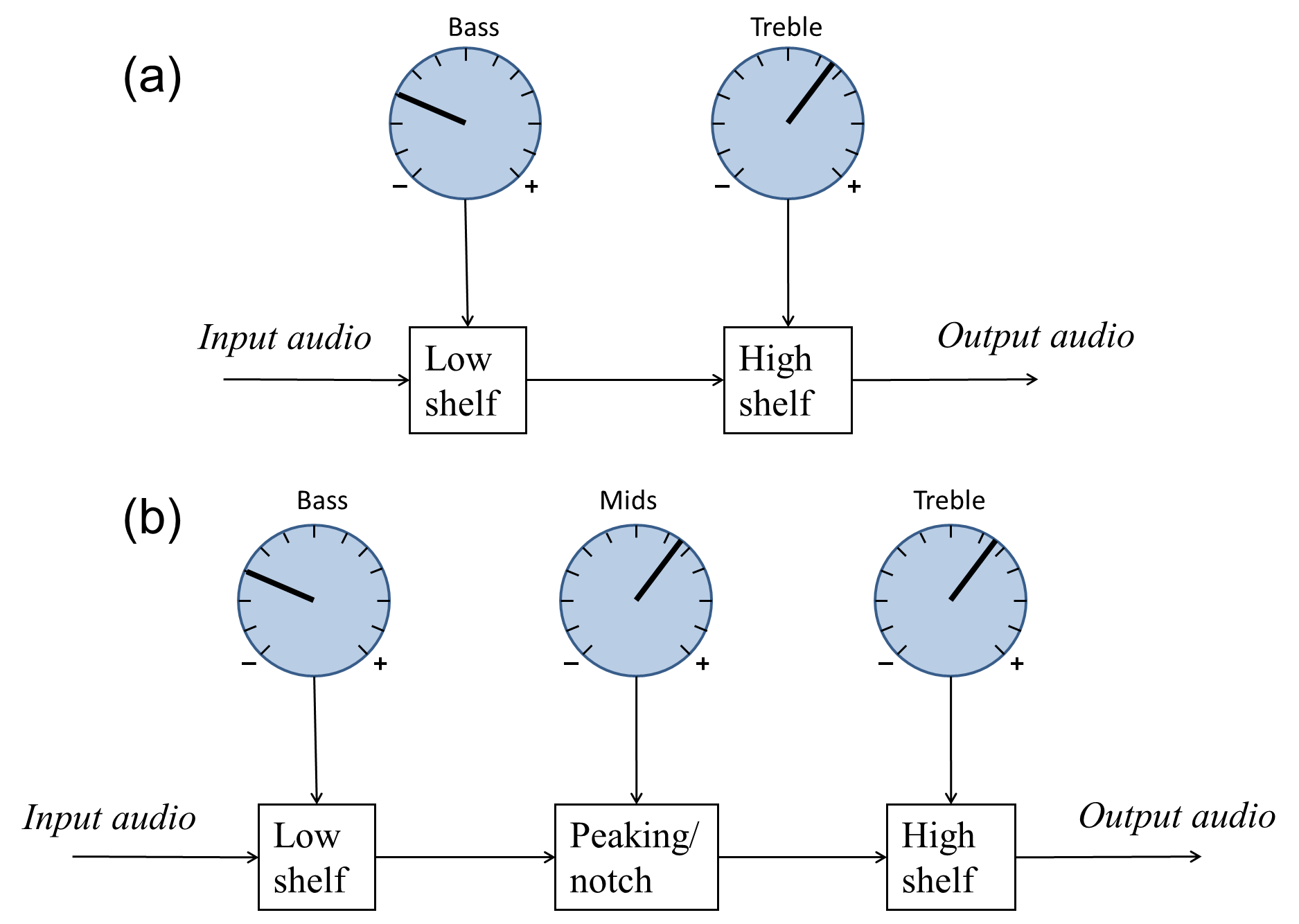
In this lab, you will implement and test a graphic equaliser that you build using the Web Audio API. Source code is not provided. So you will need to do a bit more planning and testing than in other labs.

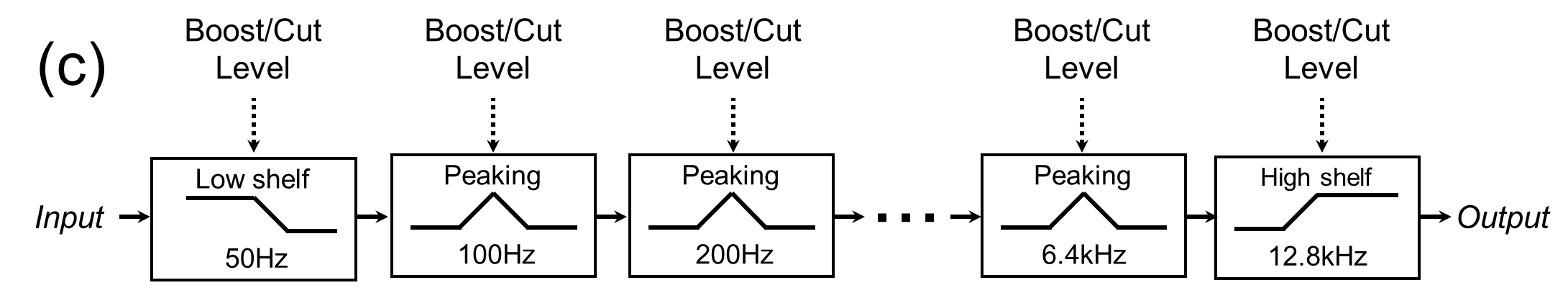
A graphic equaliser is an audio effect that allows one to adjust the gains of different frequency ranges, known as frequency bands, in a signal. The simplest one features just controls for the low frequency content (bass) and for the high frequency content (treble). Often though, they will feature lots of sliders to allow different gains on narrow frequency ranges.



*A couple of images of a graphic equaliser. The audio is divided into different frequency bands. In this case, they have centre frequencies of approximately 31.25 Hz, 62.5 Hz, 125 Hz, … 8000 Hz, 16000 Hz. The centre frequencies and Q values of each frequency band are fixed but the user can adjust the gains on each band.*

A simple way to construct a graphic equaliser is to use low shelving and high shelving filters for the lowest and highest frequency bands, and peaking filters for all the other bands. The filters are connected in series, though the order of the filters does not matter.





*(a) Bass and treble tone controls implemented as a low shelving filter and high shelving filter placed in series. (b) Three tone controls, including a peaking filter to adjust the midrange.(c) A graphic equaliser with many peaking filters to adjust the gain on different bands.*

The centre frequencies of the filters are usually given a spacing based on octaves. That is, they are logarithmic, not linear, in frequency. An octave is a factor of 2 (or 21). 1/3 octave spacing is based on a factor of 21/3 ~ 1.26.

So 1 octave spacing starting at 100Hz would have centre frequencies at 100 Hz, 200Hz, 400Hz, 800Hz, 1.6kHz, ... And 1/3 octave spacing starting at 100Hz would have centre frequencies at 100Hz, 126Hz, 159Hz, 200Hz, 252Hz, 318Hz, 400Hz, …

**Instructions:**

You will design, code and test a simple graphic equaliser. You can use the BiquadFilter nodes, or any other nodes you wish, as well as creating your own AudioWorklets. You may need to experiment with different Q values and centre frequencies so that each filter can effectively isolate just one frequency band. You don't need to use ten bands, like in the figures above. A minimum of 3 bands is sufficient.

The interface should allow the user to adjust the gain on each frequency band. The lowshelf, highshelf and peaking filters in the Web Audio API have a gain parameter, in decibels, so you can have a slider directly controlling each filter's gain parameter.

Support will be provided in the lab sessions, but this can also be done on your own time. You should customize the interface and comment your code to highlight that it is your own.

To do analysis, you should look at the spectrum of the output where noise is used as the input audio stream. The magnitude spectrum of noise should be flat, and it can be shaped by the graphic equaliser. You may wish to include other plots, such as the spectrogram or the phase spectrum, or the graphic equaliser applied to noise.

**Extra credit:**

As shown below, a graphic equaliser can also be built using a lowpass filter, bandpass filters and a highpass filters. These filters try to remove frequency content outside of a frequency band. So they need to be connected in parallel. That is, the input audio is connected to each filter, a gain is applied and then the outputs of the gains are all summed together.

Shape

Description automatically generated with medium confidence

These filters do not have a gain parameter, so gain nodes should be used. The interface should take gain values in decibels, and convert it to linear values for use as the gain parameter of each gain node.

For the extra credit, you should implement a graphic equaliser as shown. Provide source code and produce similar plots to what was done with the original design, showing that they can give similar results.

**Turn In:**

Using the online submission system, submit a ZIP archive containing:

• **Commented** **Code** for your implementation of a graphic equalizer (plus the extra credit implementation, if you tried it). Please make your code legible and easy to understand!

• **Audio files** of your graphic equaliser in action (at least two processed samples), *plus* original version of any source material used.).

• **PDF Report**, 3 or 4 sides of A4 total, explaining the graphic equaliser that you have created, and how any existing code was modified. Include at least two plots showing the result of applying the equaliser to noise and any other audio you used. You can analyse the result in SonicVisualiser or any other tool to generate plots.