Jack Walters

ECS7012P Music and Audio Programming

Assignment 1 Report

3a.

$$\frac{(2\pi 1000)^2}{s^2 + \frac{(2\pi 1000)}{\frac{\sqrt{2}}{2}} s + (2\pi 1000)^2}$$

3b.
Low-pass

$$\frac{\omega_0^2}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2}$$

$$\frac{\omega_0^2}{\left(\frac{2}{T}\,\frac{1 - z^{-1}}{1 + z^{-1}}\right)^2 + \frac{\omega_0}{Q}\left(\frac{2}{T}\,\frac{1 - z^{-1}}{1 + z^{-1}}\right) + \omega_0^2}$$

$$\frac{\omega_0^2(1 + z^{-1})^2}{\frac{4}{T^2}(1 - z^{-1})^2 + \frac{2\omega_0}{QT}(1 - z^{-1})(1 + z^{-1}) + \omega_0^2(1 + z^{-1})^2}$$

$$a = \frac{4}{T^2}$$

$$b = \frac{2\omega_0}{QT}$$

$$\frac{\omega_0^2(1 + 2z^{-1} + z^{-2})}{a(1 - 2z^{-1} + z^{-2}) + b(1 - z^{-2}) + \omega_0^2(1 + 2z^{-1} + z^{-2})}$$

$$\frac{\omega_0^2(1 + 2z^{-1} + z^{-2})}{a - 2az^{-1} + az^{-2} + b - bz^{-2} + \omega_0^2 + 2\omega_0^2 z^{-1} + \omega_0^2 z^{-2}}$$

$$\frac{\omega_0^2(1 + 2z^{-1} + z^{-2})}{(a + b + \omega_0^2) + (-2a + 2\omega_0^2)z^{-1} + (a - b + \omega_0^2)z^{-2}}$$

$$\frac{\omega_0^2(1 + 2z^{-1} + z^{-2})}{\left(\frac{4}{T^2} + \frac{2\omega_0}{QT} + \omega_0^2\right) + \left(-\frac{8}{T^2} + 2\omega_0^2\right) z^{-1} + (\frac{4}{T^2} - \frac{2\omega_0}{QT} + \omega_0^2)z^{-2}}$$

Jack Walters

$$\frac{\omega_0^2(1+2z^{-1}+\ z^{-2})}{\frac{(4Q+2\omega_0T+\ \omega_0^2QT^2)+(-8Q+2\omega_0^2QT^2)z^{-1}+(4Q-2\omega_0T+\ \omega_0^2QT^2)z^{-2}}{QT^2}}$$

$$\frac{\omega_0^2QT^2+2\omega_0^2QT^2z^{-1}+\ \omega_0^2QT^2z^{-2})}{(4Q+2\omega_0T+\ \omega_0^2QT^2)+(-8Q+2\omega_0^2QT^2)z^{-1}+(4Q-2\omega_0T+\ \omega_0^2QT^2)z^{-2}}$$

$$b_0=\ \frac{\omega_0^2QT^2}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$b_1=\ \frac{2\omega_0^2QT^2}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$b_2=\ \frac{\omega_0^2QT^2}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$a_0=\ 1$$

$$a_1=\ \frac{(-8Q+2\omega_0^2QT^2)}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$a_2=\ \frac{(4Q-2\omega_0T+\ \omega_0^2QT^2)}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

High-pass

$$\frac{s^2}{s^2+\ \frac{\omega_0}{Q}s+\omega_0^2}$$

$$\frac{\left(\frac{2}{T}\ \frac{1-z^{-1}}{1+z^{-1}}\right)^2}{\left(\frac{2}{T}\ \frac{1-z^{-1}}{1+z^{-1}}\right)^2+\ \frac{\omega_0}{Q}\left(\frac{2}{T}\ \frac{1-z^{-1}}{1+z^{-1}}\right)+\omega_0^2}$$

$$\frac{\frac{4}{T^2}(1-z^{-1})^2}{\frac{4}{T^2}(1-z^{-1})^2\ +\ \frac{2\omega_0}{QT}(1-z^{-1})(1+z^{-1})+\omega_0^2(1+z^{-1})^2}$$

$$a=\ \frac{4}{T^2}$$

$$b=\ \frac{2\omega_0}{QT}$$

$$\frac{a(1-\ z^{-1})^2}{a(1-z^{-1})^2+\ b(1-z^{-1})(1+z^{-1})+\omega_0^2(1+z^{-1})^2}$$

$$\frac{a(1-2z^{-1}+\ z^{-2})}{a(1-2z^{-1}+\ z^{-2})\ +\ b(1-z^{-2})+\omega_0^2(1+2z^{-1}+\ z^{-2})}$$

$$\frac{a(1-2z^{-1}+\ z^{-2})}{a-2az^{-1}+az^{-2}+b-bz^{-2}+\ \omega_0^2+2\omega_0^2z^{-1}+\omega_0^2z^{-2}}$$

$$\frac{a(1-2z^{-1}+\ z^{-2})}{(a+b+\ \omega_0^2)+(-2a+2\omega_0^2)z^{-1}+(a-b+\omega_0^2)z^{-2}}$$

$$\frac{a(1-2z^{-1}+\ z^{-2})}{\left(\frac{4}{T^2}+\frac{2\omega_0}{QT}+\ \omega_0^2\right)+\left(-\frac{8}{T^2}+2\omega_0^2\right)z^{-1}+(\frac{4}{T^2}-\frac{2\omega_0}{QT}+\omega_0^2)z^{-2}}$$

$$\frac{a(1-2z^{-1}+\ z^{-2})}{\frac{(4Q+2\omega_0T+\ \omega_0^2QT^2)+(-8Q+2\omega_0^2QT^2)z^{-1}+(4Q-2\omega_0T+\ \omega_0^2QT^2)z^{-2}}{QT^2}}$$

$$\frac{4Q-8Qz^{-1}+4Qz^{-2}}{(4Q+2\omega_0T+\ \omega_0^2QT^2)+(-8Q+2\omega_0^2QT^2)z^{-1}+(4Q-2\omega_0T+\ \omega_0^2QT^2)z^{-2}}$$

$$b_0=\ \frac{4Q}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$b_1=\ \frac{-8Q}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$b_2=\ \frac{4Q}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$a_0=\ 1$$

$$a_1=\ \frac{(-8Q+2\omega_0^2QT^2)}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

$$a_2=\ \frac{(4Q-2\omega_0T+\ \omega_0^2QT^2)}{(4Q+2\omega_0T+\ \omega_0^2QT^2)}$$

Jack Walters

3c.
Low-pass difference equation

$$y[n] = \frac{\omega_0^2 QT^2}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} x[n] + \frac{2\omega_0^2 QT^2}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} x[n-1]$$

$$+ \frac{\omega_0^2 QT^2}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} x[n-2] + \frac{(-8Q + 2\omega_0^2 QT^2)}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} y[n-1]$$

$$+ \frac{(4Q - 2\omega_0 T + \ \omega_0^2 QT^2)}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} y[n-2]$$

High-pass difference equation

$$y[n] = \frac{4Q}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} x[n] + \frac{-8Q}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} x[n-1]$$

$$+ \frac{4Q}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} x[n-2] + \frac{(-8Q + 2\omega_0^2 QT^2)}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} y[n-1]$$

$$+ \frac{(4Q - 2\omega_0 T + \ \omega_0^2 QT^2)}{(4Q + 2\omega_0 T + \ \omega_0^2 QT^2)} y[n-2]$$

3d.

After having initialised the coefficient variables as global variables at the top of the code, their values are calculated inside the calculate_coefficients() function using sampling frequency, Q and cut-off frequency as parameters. This function updates the global variables, which are then used inside the render() function. It is worth noting that to ensure $y[0]$ was normalised to 1, in the render() function every other coefficient was divided by the original $y[0]$ value derived from the bilinear transform.

Shared global coefficient variables:

```
b0 = (pow(Omega0,2)*q*pow(T,2));
b1 = (2 * pow(Omega0,2)*q*pow(T,2));
b2 = (pow(Omega0,2)*q*pow(T,2));

a0 = (4*q + 2*Omega0*T + pow(Omega0,2)*q*pow(T,2));
a1 = (-8*q + 2*pow(Omega0,2)*pow(T,2)*q);
a2 = (4*q - 2*Omega0*T + pow(Omega0,2)*q*pow(T,2));
```

4.

Each filter outlines its own copy of the difference equation calculated in step 3.

Filter 1 difference equation computation (inside the render() for() loop) (filter 2 is structured in the same way only differing by state variable names):

```
float F1out = ((
      (b0 * in) + (b1 * F1gLastInput) + (b2 * F1gLastLastInput))
      -
      ((a1 * F1gLastOutput) + (a2 * F1gLastLastOutput)))
      /a0;

F1gLastLastInput = F1gLastInput;    // x[n-2] = x[n-1]
F1gLastInput = in;    // x[n-1] = x[n]

F1gLastLastOutput = F1gLastOutput;    // y[n-2] = y[n-1]
F1gLastOutput = F1out;    // y[n-1] = y[n]
```

Sample storage variables:

Sample number	Filter 1	Filter 2
$x[n]$	in	F1out
$x[n-1]$	F1gLastInput	F2gLastInput
$x[n-2]$	F1gLastLastInput	F2LastLastInput
$y[n]$	F1out	out
$y[n-1]$	F1gLastOutput	F2LastOutput
$y[n-2]$	F1gLastLastOutput	F2LastLastOutput

Upon the first iteration of the for() loop at time step 0, the difference equation is only supplied with value $x[n]$ (variable in for 1st filter, F1out for 2nd filter) and despite all coefficient values being immediately available, all other inputs and outputs sum to 0 as the variables for storing input and output values of $[n-1]$ and $[n-2]$ have not been updated yet so are still in their initialisation state of 0.

When the for() loop iterates for a 2nd time, $x[n-1]$ has been set to $x[n-2]$, $x[n]$ has been set to $x[n-1]$, $y[n-1]$ has been set to $y[n-2]$ and $y[n]$ has been set to $y[n-1]$. Now when the difference equation is calculated in this 2nd for() loop iteration, a new $x[n]$ is processed, as well as $x[n-1]$ and $y[n-1]$, which were $x[n]$ and $y[n]$ in the previous for() loop iteration.

By the time of the 3rd for() loop iteration, all sample variables have been updated again in the same way as previously described. When the difference equation is calculated in this 3rd for() loop iteration, what was $x[n]$ in the 1st for() loop, is now $x[n-2]$ and what was $y[n]$ in the 1st for() loop, is now $y[n-2]$.

Throughout the whole process described above, the output F1out of the 1st filter is used as $x[n]$ for the 2nd filter within the for() loop, and the output out of the 2nd filter is used as the input value parameter for audioWrite().

5.

To take audio from Bela's *audio input* jack, the `in` variable is set to Bela's `audioRead()` function inside of `render()` instead of it being the array `gSamplebuffer[gReadPointer]` which contains the sawtooth wave.

-6dB in relative amplitude

$10^{\frac{-6}{20}} = 0.501$

Input sine wave = 920mV
920 * 0.501 = 460.92mV

1000Hz cut-off in code
Frequency of sine wave output from Bela board when its amplitude is at ~460.92mV = 1053Hz

800Hz cut-off in code
Frequency of sine wave output from Bela board when its amplitude is at ~460.92mV = 843Hz

500Hz cut-off in code
Frequency of sine wave output from Bela board when its amplitude is at ~460.92mV = 525Hz

1500Hz cut-off in code
Frequency of sine wave output from Bela board when its amplitude is at ~460.92mV = 1578Hz

2000Hz cut-off in code
Frequency of sine wave output from Bela board when its amplitude is at ~460.92mV = 2091Hz

3000Hz cut-off in code
Frequency of sine wave output from Bela board when its amplitude is at ~460.92mV = 3130Hz

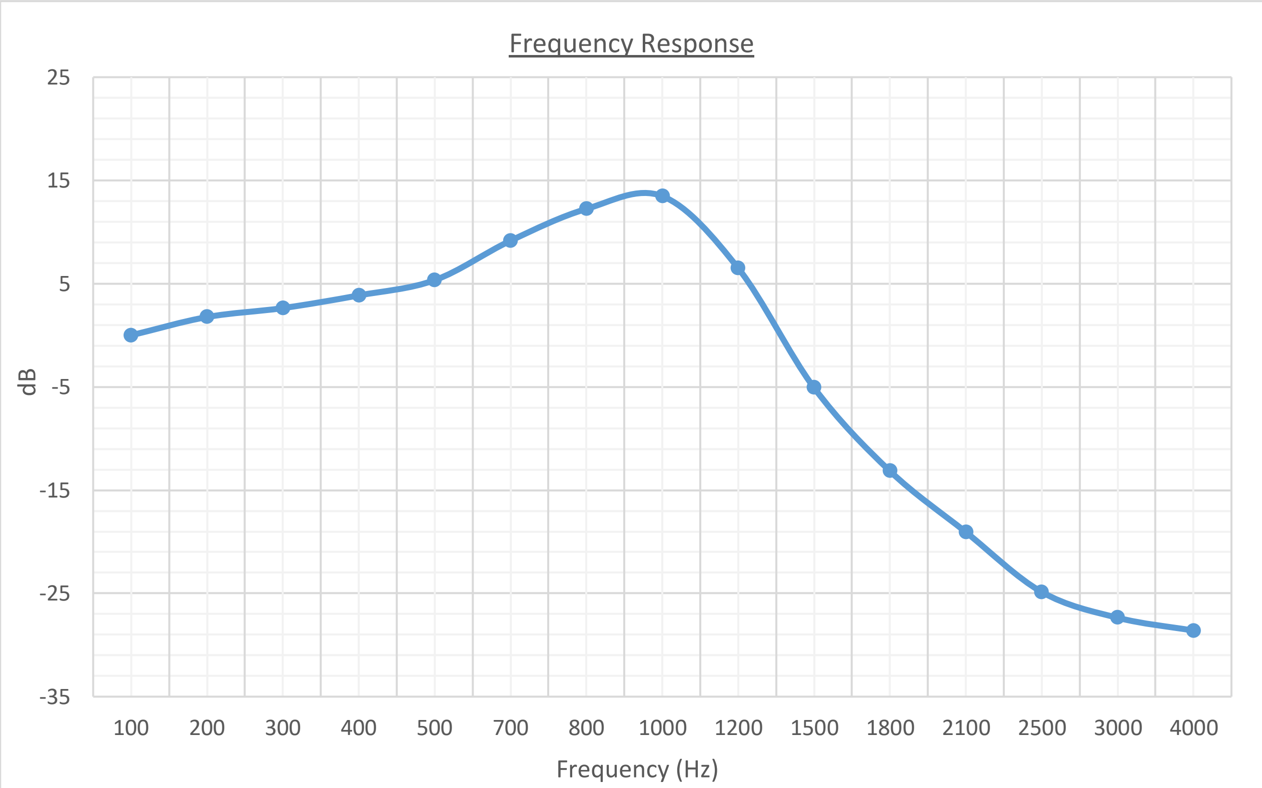

Figure 1 – Frequency response of the filter with cut-off at 1000Hz, and Q at 2

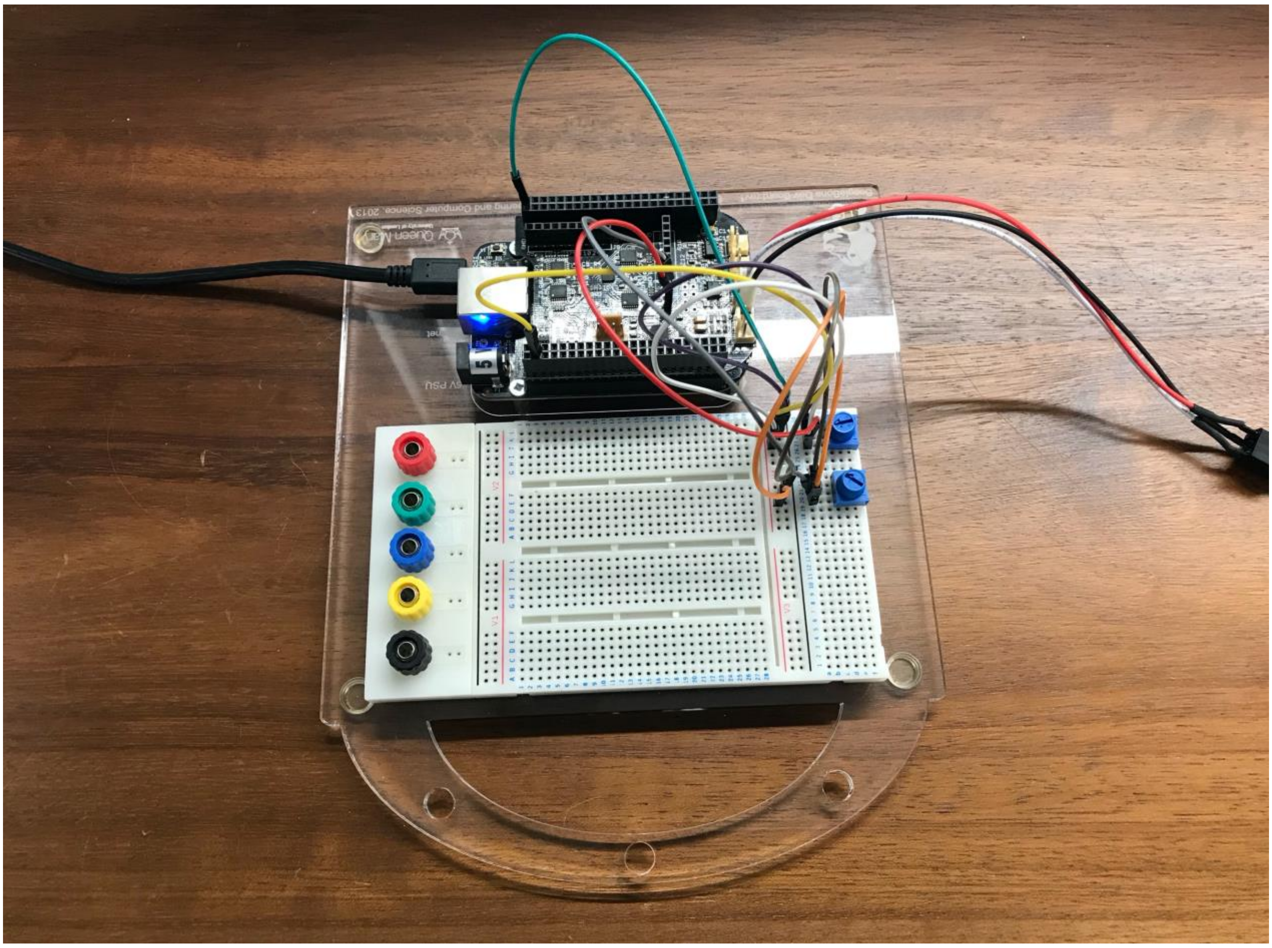

Figure 2 – Potentiometers wired up to the bread board, Bela 3.3V, ground and analog IN ports

The 4th order low-pass biquad filter designed in this assignment has successfully displayed behaviour typical of a low pass filter. At higher Q values, the resonance just below the filter cut-off is larger than at lower Q values and the frequency response exhibits sharper roll-off. It may have been desirable to encapsulate the filter in a header filer, with the `calculate_coefficient` functions being contained as methods.

Despite displaying a frequency response analogous to other 4th order filters of similar cutoffs and Q values, the readings on the previous page of this report show that regardless of what frequency the filter cut-off is set to in the code, the actual frequency at which the relative amplitude of the low-passed sine wave to the input is 0.501 (-6dB), is marginally higher than it should be by a factor of roughly 1/20. One design element I struggled to reconcile with the validity of the Bela-output sine wave's amplitude when being read in the oscilloscope was the fact that Bela's headphone, DAC, ADC, and stereo PGA gain can be adjusted through its IDE settings. I found that to ensure Bela's output signal matched the lab signal generator control sine wave's amplitude in the `passthrough` example in Bela's example code, I needed to decrease the gain of the headphone level by -6dB and increase the gain in both PGA channels by 10dB. Furthermore, initially when testing the amplitude of the Bela-output sine wave in the oscilloscope, its amplitude nearly doubled that of the control when the control was safely in the passband at 100Hz. After some adjustment of gain levels in the IDE settings I managed to achieve a frequency response that seemed typical to a typical 2nd order system's cut-off and Q setting. Later I found that the issue corrected itself once the Bela board had been powered on and off a few times, and that now I didn't need to adjust any gain levels at all. This was definitely a better position to be in as that I didn't need to estimate gain levels myself, however it did cast some doubt over Bela's output gain consistency for the remainder of the experiment.