Introduction

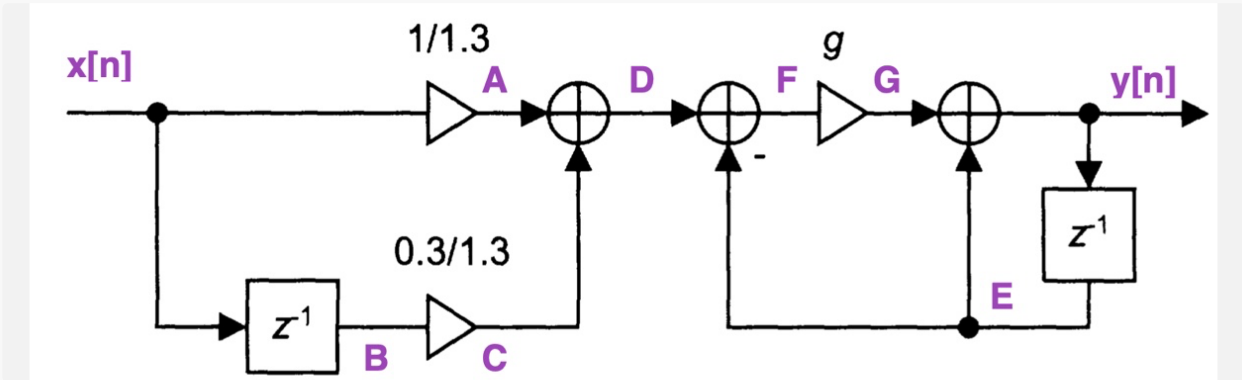
This practical assessment develops oversight on how to implement a resonant lowpass filter, based on Robert Moogs analog subtractive synthesis.

Section 1 Coefficient

This section goes through how to obtain the filter coefficients used in the calculating process.

1. Write equations for the value of the signal at each labelled node.

Figure 1 : Stilson and Smith (1996).



$$A = \frac{x[n]}{1.3}$$

$$x[n] \times \frac{1}{1.3} = \frac{x[n]}{1.3}$$

$$B = x[n-1]$$

Unit delay

$$C = \frac{0.3x[n-1]}{1.3}$$

$$x[n-1] \times \frac{0.3}{1.3} = \frac{0.3x[n-1]}{1.3}$$

$$D = \frac{x[n]+0.3x[n-1]}{1.3}$$

$$C + B$$

$$E = y[n-1]$$

Unit delay

$$F = \frac{x[n]+0.3x[n-1]}{1.3} - y[n-1]$$

The **previous combined** output of $x[n]$ **subtracted** by the **unit delay** of $y[n]$

$$G = g(\frac{x[n]+0.3x[n-1]}{1.3} - y[n-1])$$

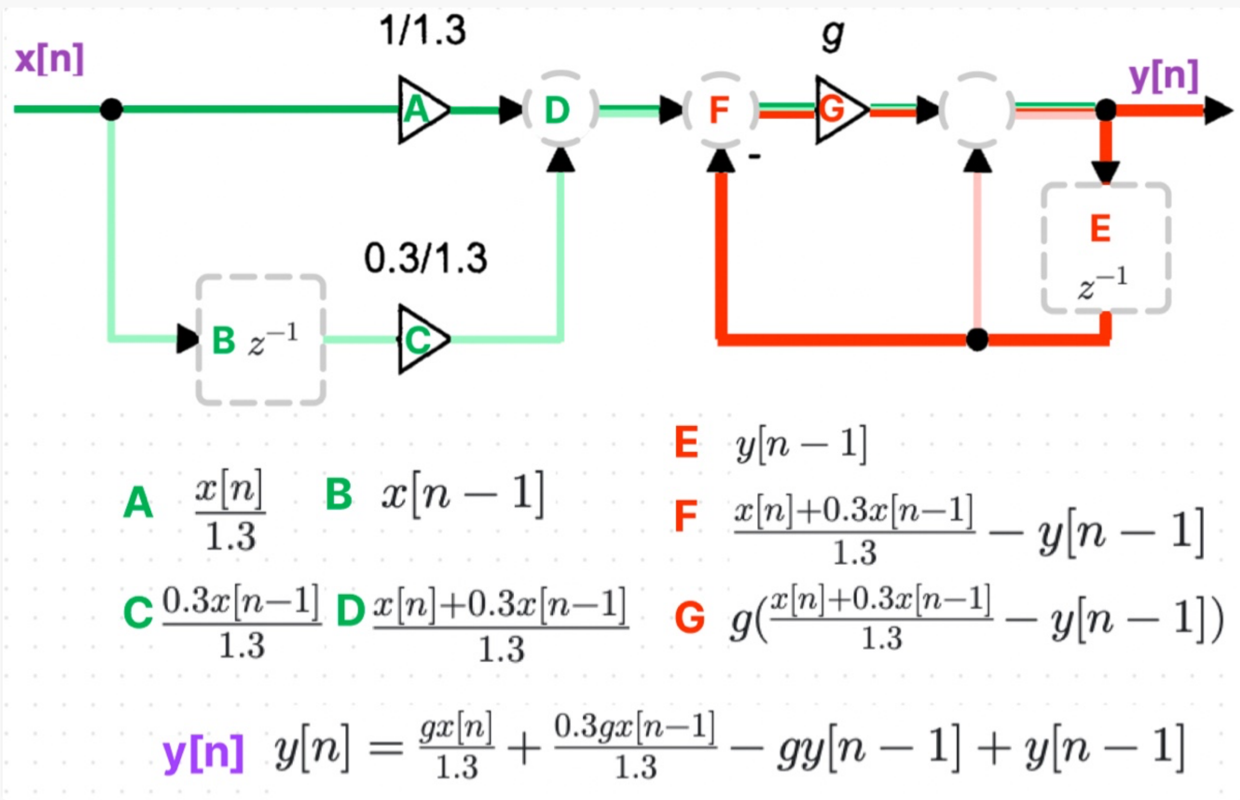
The **previous combined** output of $x[n]$ **subtracted** by the **unit delay** of $y[n]$**multiplied** by g

---

$$y[n] = \frac{gx[n]}{1.3} + \frac{0.3gx[n-1]}{1.3} - gy[n-1] + y[n-1]$$

Following each coefficient (b_0, b_1, a_1) through the circuit, they all eventually meet g so have to be **multiplied** with it, for example $\frac{x[n]}{1.3} = g\frac{x[n]}{1.3}$

Figure 2 : Stilson and Smith (1996) (edited using Figma) modified diagram.



Coefficients for the filter are

$$b_0 = \frac{g}{1.3}$$

$$b_1 = \frac{0.3 \times g}{1.3}$$

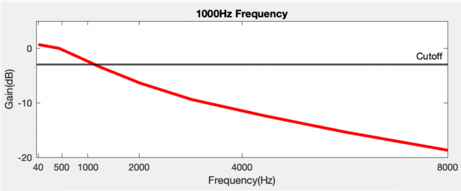
$$a_1 = g - 1$$

Section 2 Graphs

This section walks through the improvements in frequency response, when changing the equation g or applying multiple filter blocks to the filter.

2. At what frequency (approximately) do you find the $-3dB$ point? for $1000Hz$

Figure 3: Graph of filters frequency response at frequency cutoff $1000Hz$ (plotted using Matlab) Gain (dB): Frequency (Hz).

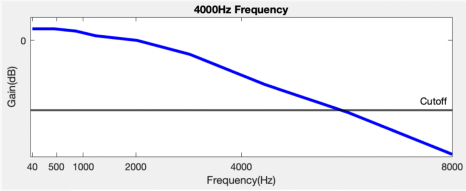


Frequency	Decibels
$8000Hz$	-18.7dB
$6025Hz$	-15.4dB
$4450Hz$	-12.4dB
$3024Hz$	-9.4dB
$2024Hz$	-6.4dB
$1235Hz$	-3.4dB
$866Hz$	-1.8dB
$445Hz$	0dB
$40Hz$	0.7dB

$1235Hz$

3. At what frequency (approximately) do you find the $-3dB$ point? for $4000Hz$

Figure 4: Graph of filters frequency response at frequency cutoff $4000Hz$ (plotted using Matlab) Gain (dB): Frequency (Hz).



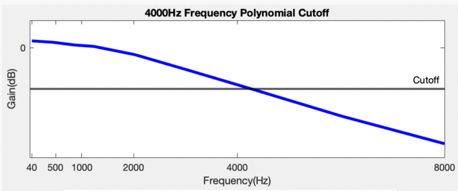
Frequency	Decibels
$8000Hz$	-4.9dB
$6025Hz$	-3.1dB
$4450Hz$	-1.9dB
$3024Hz$	-0.6dB
$2024Hz$	0dB
$1235Hz$	0.2dB
$866Hz$	0.4dB
$445Hz$	0.5dB
$40Hz$	0.5dB

$$6025Hz$$

This answer is **not** entirely **accurate** due to the **cutoff frequency** of the filter being set at $4000Hz$.

4. Change g to the polynomial equation. Leaving the filter frequency at $4000Hz$, run the project again and measure the $-3dB$ point again. Is it more accurate?

Figure 5: Graph of filters frequency response at frequency cutoff $4000Hz$ using a polynomial formula (plotted using Matlab) Gain (dB): Frequency (Hz).

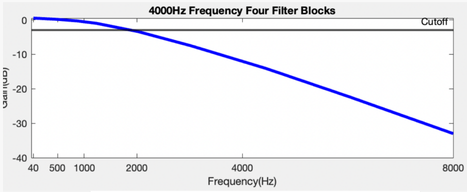


Frequency	Decibels
$8000Hz$	-7dB
$6025Hz$	-5dB
$4450Hz$	-3.2dB
$3024Hz$	-1.6dB
$2024Hz$	-0.5.dB
$1235Hz$	0.1dB
$866Hz$	0.2dB
$445Hz$	0.4dB
$40Hz$	0.5

I would say it is **more accurate**, due to the **cutoff frequency** being **closer** to $4000Hz$, however the $-3dB\ point$ is lower visually, than where the filter starts to drop. I would say this could be due to the longer journey to reach the $-3dB\ point$. In *figure 3* after the filter reaches $-3dB$ the decibels drops are more significant, *figure 6* has an equivalent pattern to *figure 3.*

5. Take a measurement of this new fourth-order filter at the same frequency. Now what is the gain at this frequency? Why?

Figure 6: Graph of filters frequency response at frequency cutoff $4000Hz$ using the four filter blocks (plotted using Matlab) Gain (dB): Frequency (Hz).



Frequency	Decibels
$8000Hz$	-33.0dB
$6025Hz$	-22.3dB
$4450Hz$	-14.1dB
$3024Hz$	-7.5dB
$2024Hz$	-3.4dB
$1235Hz$	-1dB
$866Hz$	-0.3dB
$445Hz$	0.2dB
$40Hz$	0.5dB

$-14dB$ at $4497Hz$

You are putting the frequency through **multiple filter blocks**, therefore when you add up all of the cutoff points they become one.

$$-3.4 \times 4 = -13.6 = -14dB$$

Section 3 Analysing Nonlinearity

This section talks about the sensitivity of the non linearity function `neon_tanf()`.

6. What happens to the shape of the blue signal as you increase the amplitude? (Time domain).

The **blue signal adds new harmonics**, as you increase the amplitude above $0dB$. You can see this, if you look carefully at Figure 7; the **smaller sine wave** shapes **are inserted in-between** *the original blue sine-wave*. The higher the amplitude above $0dB$, the more defined the sandwiched blue sine wave becomes.

7. What do you see when the sine wave has a low amplitude? What happens as you increase the amplitude? Why? (FFT decibels)

When the sine wave has the **lowest amplitude** there is only **one harmonic**; the higher the amplitude the higher the increase of harmonics.

You are **crossing** the **threshold** of **nonlinearity** ("oversampling by at least a factor of 2" (Välimäki and Huovilainen, 2006)) adding new harmonics up and down the spectrum.

Figure 7: Oscilloscope of the time domain (top) and the frequency domain (bottom) in decibels.

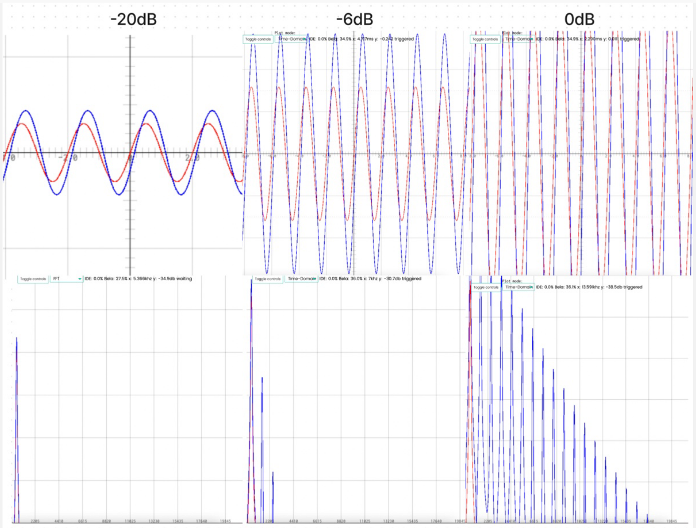
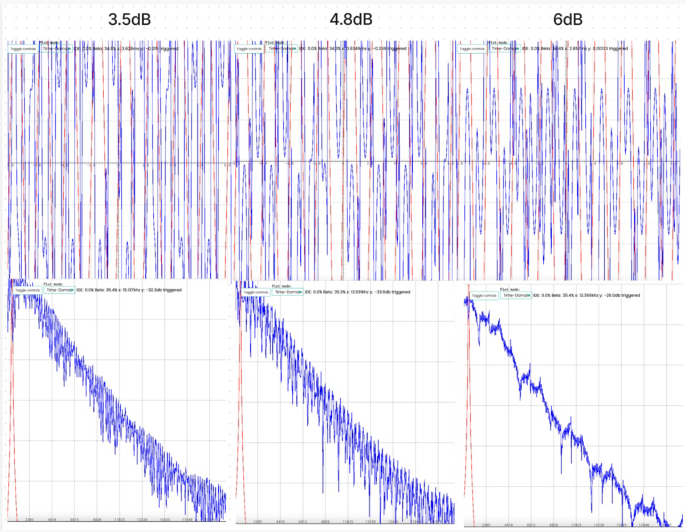


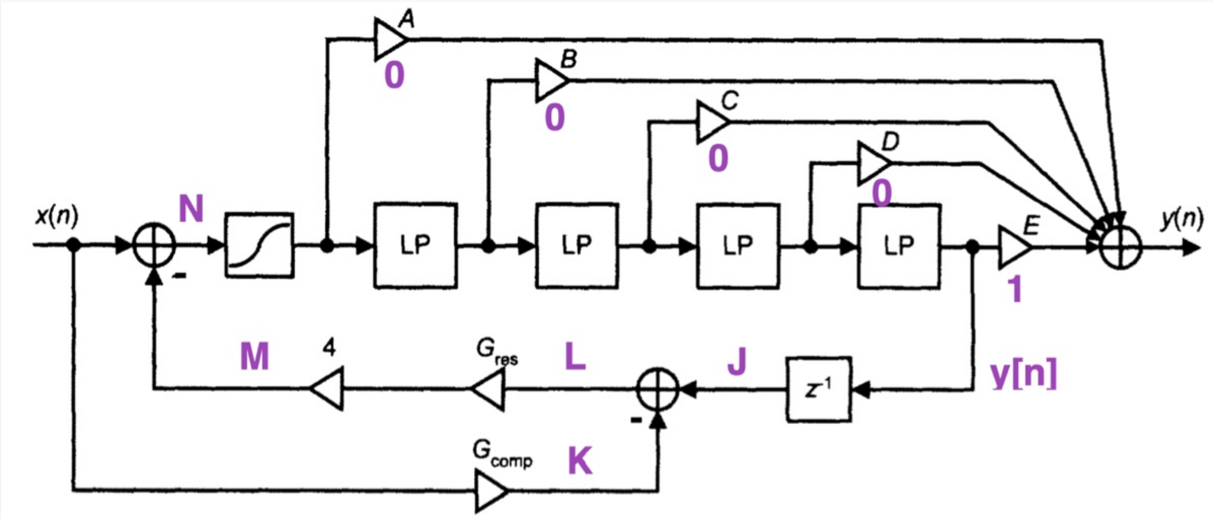
Figure 8: Oscilloscope of the time domain (top) and the frequency domain (bottom) in decibels.



Section 4 Feedback Calculation

This section finds the formula used to calculate the feedback, by putting $x[n]$ through the block diagram first then $y[n]$ and subtracting for the result.

Figure 9: Huovilainen (2006).



---

8. write out the formula for **each letter** in terms of x, y, G_{res}, G_{comp}.

$$j = y[n-1]$$

Unit delay

$$K = x[n]g_{comp}$$

$Input$ goes through g_{comp}

$$L = y[n-1] - x[n]g_{comp}$$

$Input$ goes through Unit Delay goes in first **subtracted** by K

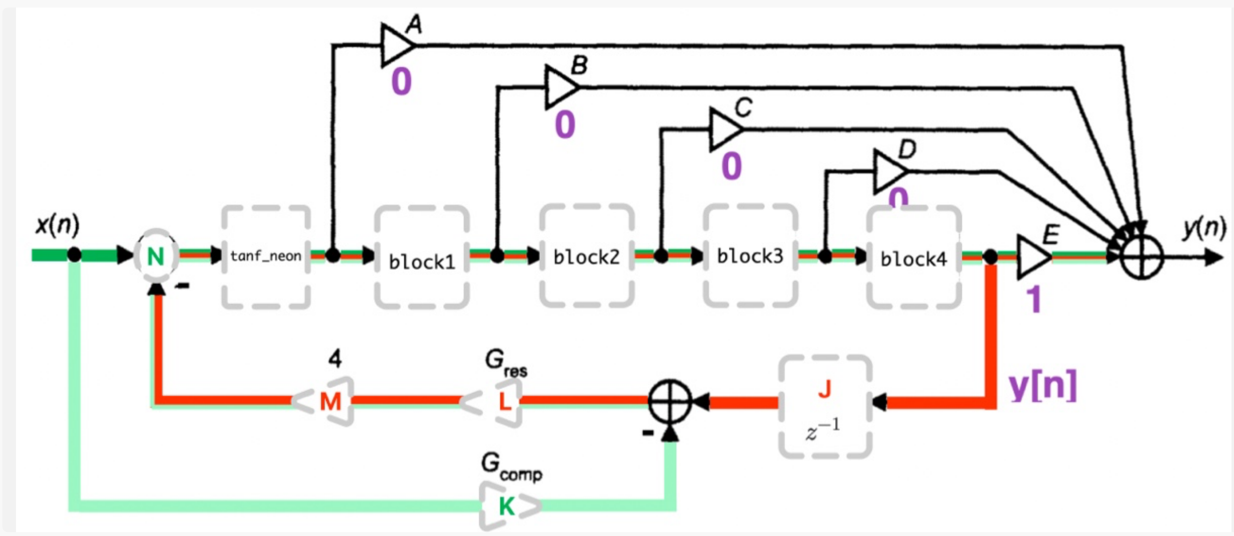
$$M = 4g_{res}(y[n-1] - x[n]g_{comp})$$

The combined values of J & K goes through $4g_{res}$

$$N = x[n] - (4g_{res}(y[n-1] - x[n]g_{comp}))$$

The $input$ is **subtracted** by the combined value M

Figure 10: Huovilainen (2006) (edited using Figma) modified diagram.



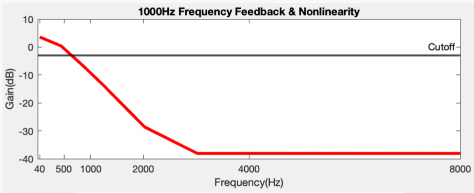
Section 5 Peaks & Harmonics

This section focuses on the harmonics added by the non linearity function and the peaks of the signal in the oscilloscope.

---

9. What frequency has the peak magnitude response? What is the gain (in decibels) at the peak?

Figure 11: Graph of filters frequency response at $4000Hz$ using feedback and the nonlinearity function (plotted using Matlab) Gain (dB): Frequency (Hz).



Frequency	Decibels
$8000Hz$	-38.1dB
$6025Hz$	-38.1dB
$4450Hz$	-38.1dB
$3024Hz$	-38.1dB
$2024Hz$	-28.6 dB
$1235Hz$	-13.5dB
$866Hz$	-6.8dB
$445Hz$	0.4dB
$40Hz$	3.6dB

$$40Hz$$

$$3.6dB$$

---

10. *To test whether this adjustment works, run the project with a sawtooth oscillator input. (Try a low oscillator frequency, say 100Hz, and an amplitude of 0.3, for a clear result.) How does the sound change when you sweep the filter frequency up and down?*

When you start at $100Hz$ for the cutoff frequency, you get one **low amplitude harmonic** in both the frequency and time domain. As you **increase** the cutoff frequency the **amplitude and harmonics rise** forming a similar shape to its red counter part (a sawtooth signal).

---

11. Once you have finished implementing this, run the project again with a sawtooth wave input and the oscilloscope set to FFT mode. Set the oscillator to around 100Hz, amplitude 0.3, and the filter to 2000Hz and resonance 0.9. Include a screenshot of the plot in your report. (Hint: in the oscilloscope controls there is a button "Pause plotting" that you can use to freeze the display for taking a screenshot.) Can you see the resonant peak?

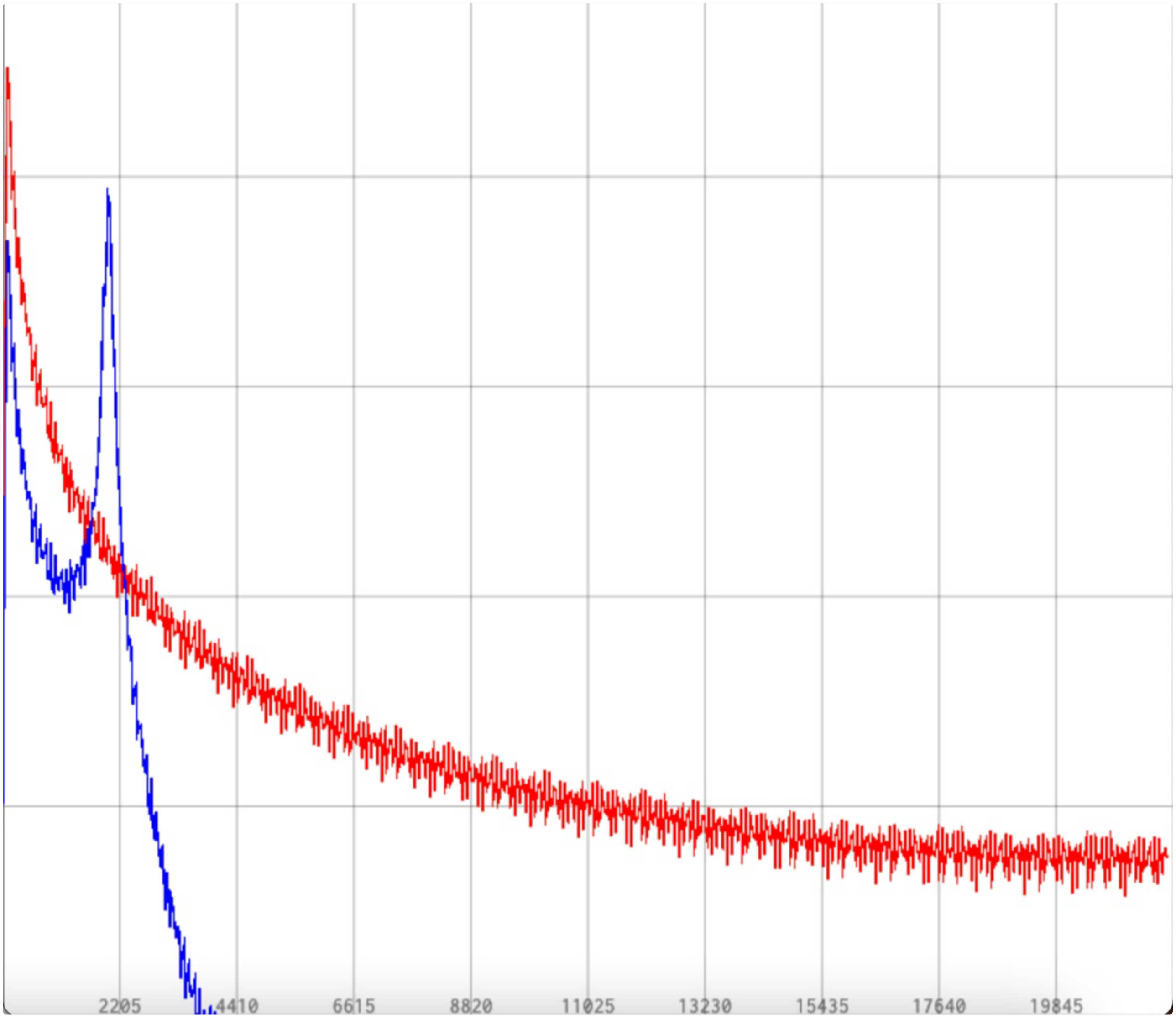
Figure 12: Oscilloscope of the Frequency domain in decibels.

(Oscillator Frequency: 100

Oscillator Amplitude: 0.2635

Filter Cutoff: 2000

Resonance: 0.9)

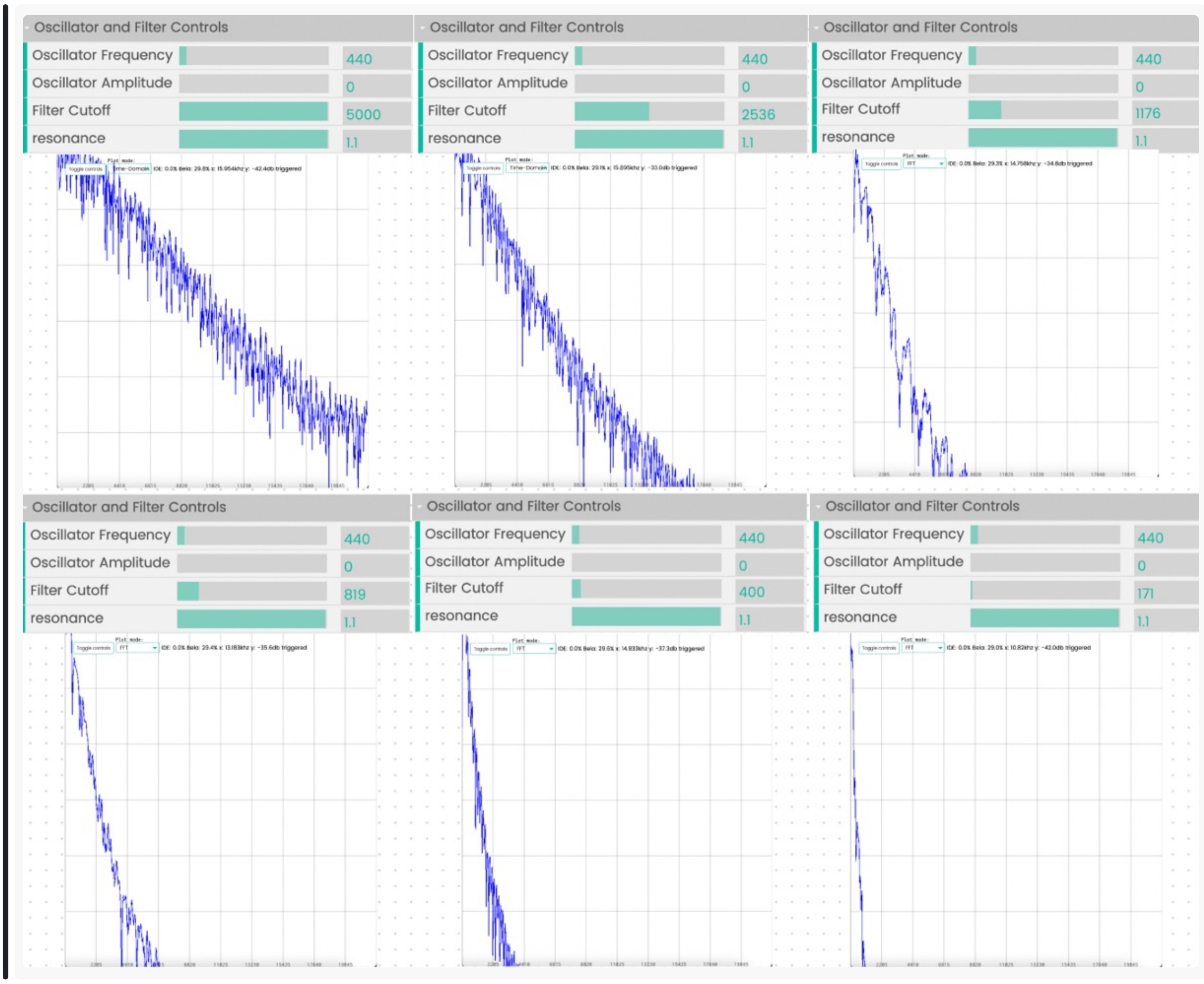


If you go **below** 0.2635 oscillator amplitude you can see the peak, if you don't, you get in and out noise at 0.264 to 0.265 then constant noise from 0.27.

---

12. Now try changing the range of the resonance slider to go from 0 all the way up to 1.1. Run the project again and push the resonance all the way up to the maximum. Try adjusting the filter cutoff frequency, and then turn the oscillator amplitude down to 0. Something unusual happens. What is it, and why do you think it happens? (Hint: this was a feature of the original analog Moog VCF too!)

Figure 13: Oscilloscope of the Frequency domain in decibels, displaying its filter controls.



As you adjust the frequency cutoff with the resonance at 1.1, you are able to **control** the **amount** of **nonlinearity** on its own with out the original sound.

I think it happens due to the sensitivity of the nonlinearity function, separately to the frequencies amplitude.

Optional Section 6

This section is an add-on looking at how changing the amplitude values at A, B, C, D and E changes the filter from two pole to four pole and low pass to either high or band pass filters.

Low-Pass

Figure 14: Huovilainen (2006) (edited using Figma) modified diagram Low pass.

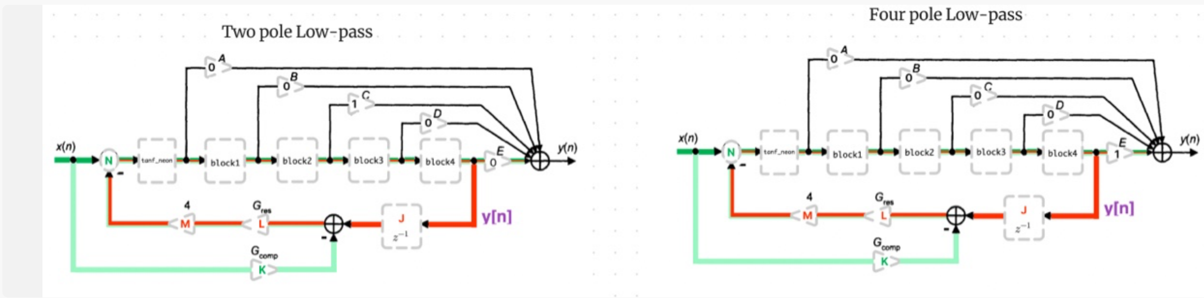
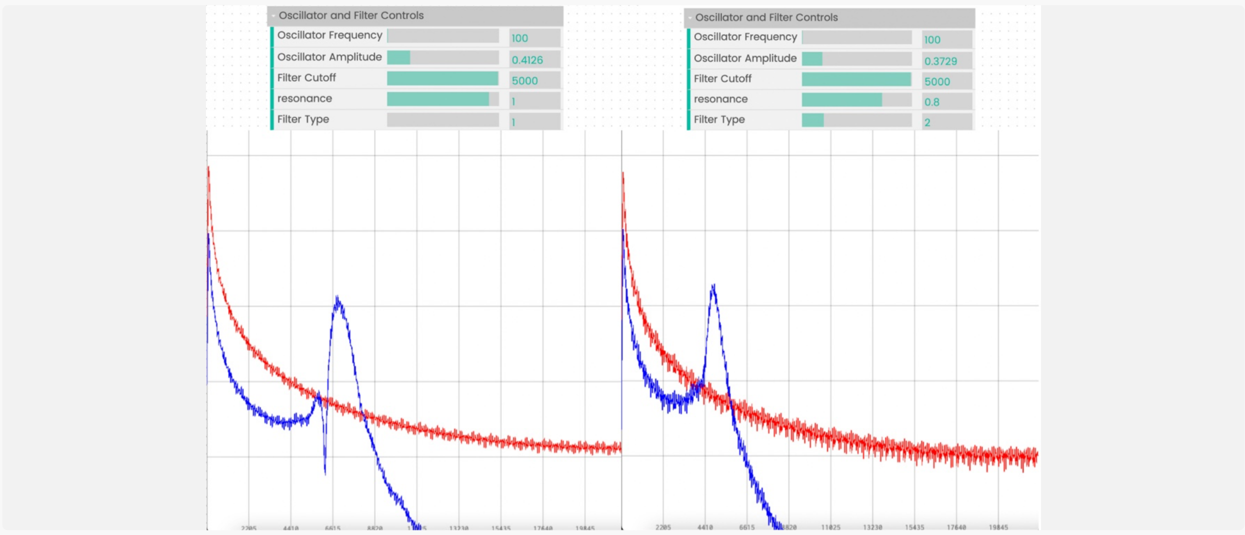


Figure 15: Oscilloscope of the Frequency domain in decibels, displaying its filter controls (Low Pass).



Band-Pass

Figure 16: Huovilainen (2006) (edited using Figma) modified diagram Band pass.

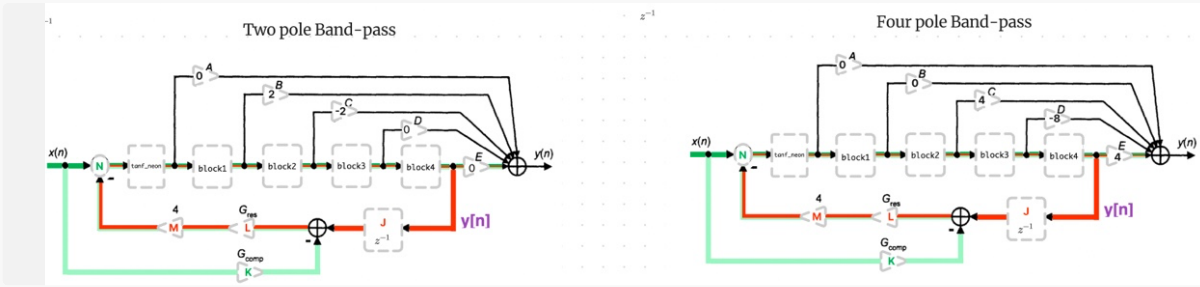
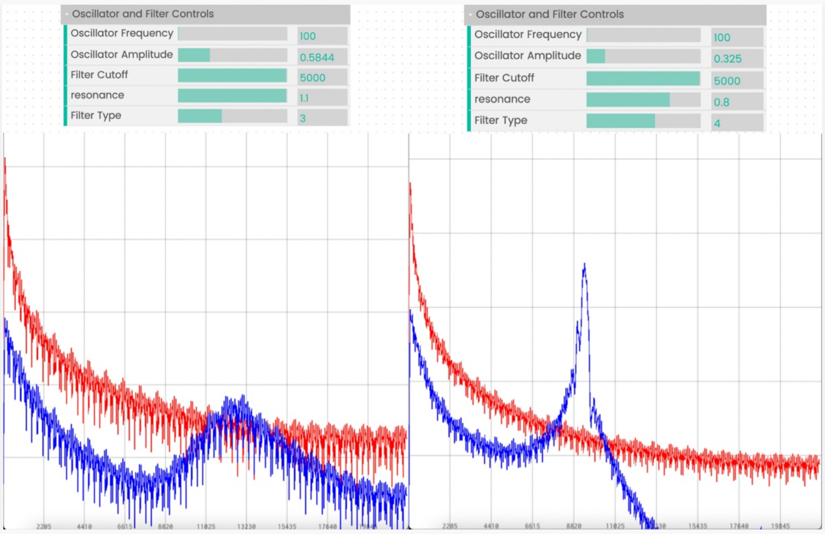


Figure 17: Oscilloscope of the Frequency domain in decibels, displaying its filter controls (Band Pass).



High-Pass

Figure 18: Huovilainen (2006) (edited using Figma) modified diagram High pass.

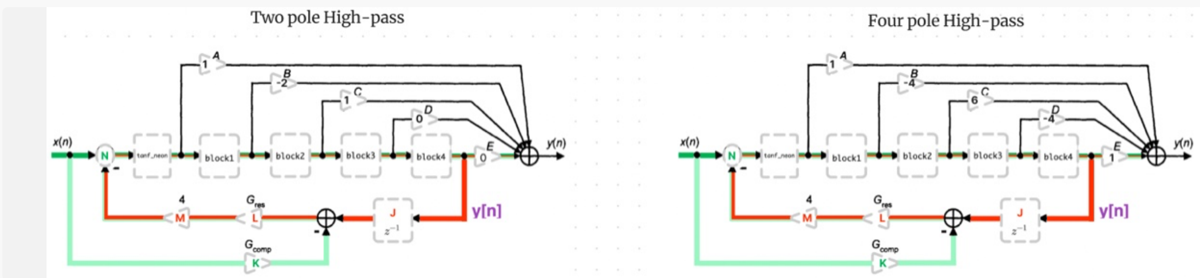
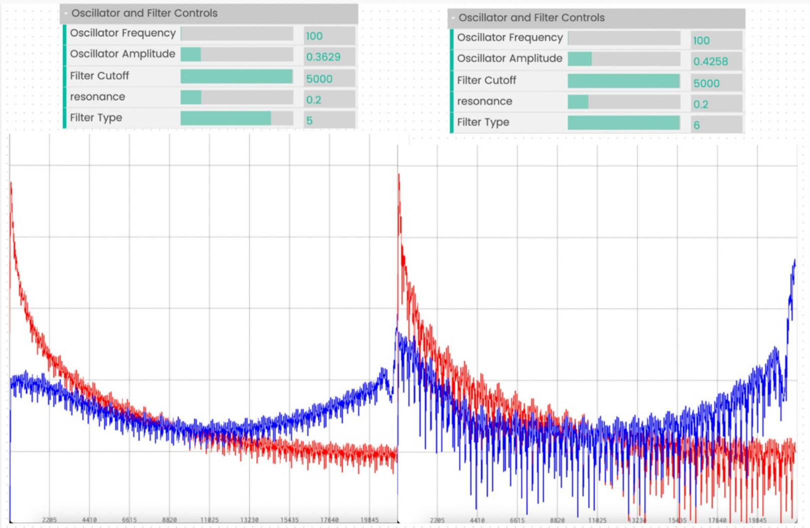


Figure 19: Oscilloscope of the Frequency domain in decibels, displaying its filter controls (High Pass).



Conclusion

In conclusion, the digital representation of the analog filter; has its strengths in the richness of its harmonics and its weaknesses in the sensitivity of the nonlinearity function. Each step of this assessment adds improvements to previous iterations, allowing for visual comparisons of the benefits and disadvantages of the different implementations. For further improvement if possible, I would say to reduce the effect of the nonlinearity filter. This could be done by just setting the minimum and maximum thresholds lower for the specified filter type.

References

Välimäki, V. and Huovilainen, A., 2006. Oscillator and filter algorithms for virtual analog synthesis. Computer Music Journal, 30(2), pp. 2.