Filter Design & Monte Carlo Analysis

David Collins
C10736685

December 2012

1 Background

1.1 Filter specifications

We have been asked to design a filter which meets the following requirements:
Pass-band: $A_{max} = 1\text{dB}$, $f_p = 1\text{kHz}$
Stop-band: $A_{min} = 15\text{dB}$, $f_s = 4\text{kHz}$
The meaning of these parameters are explained with the aid of Figure 1.

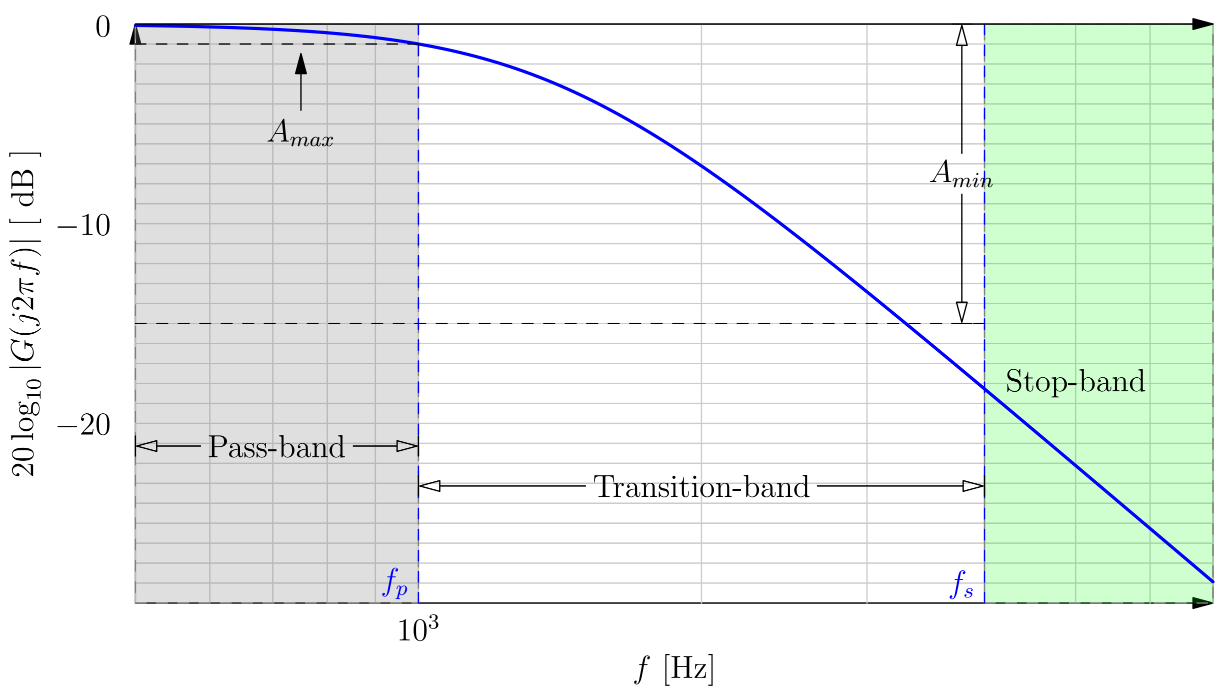


Figure 1: Frequency response of a low-pass filter

- f_p — the frequency at the upper-edge of the pass-band; frequencies below this can be attenuated by no more than A_{max}
- f_s — the frequency at the lower-edge of the stop-band; frequencies above this must be attenuated by at least A_{min}
- A_{max} — the maximum acceptable loss in the *passband*

- A_{min} — the minimum acceptable loss in the *stopband*

1.2 Ideal vs. practical filters

Although an ideal filter is impossible — even in the digital domain — increasing the order of a filter allows one to approach the ideal frequency response. There are practical limitations on the order of the filter however — in the analog domain due to the number of components required and in the digital domain due to the time of computation.

1.3 Butterworth approximation functions

There are various classes of filter functions which could be used for the present purpose. We have chosen to use a Butterworth filter, however. The transfer functions for normalised versions of such filters — for filters of various order — are

$$\begin{array}{ll} \mathbf{n=1:} & \frac{1}{s+1} \\ \mathbf{n=2:} & \frac{1}{s^2+\sqrt{2}s+1} \\ \mathbf{n=3:} & \frac{1}{s^2+s+1}\cdot\frac{1}{s+1} \end{array} .$$

1.3.1 Denormalisation

In the course of designing a filter these transfer functions are *denormalized* by replacing s with $\frac{\epsilon^{1/n}}{\omega_p}s$, where ω_p is the angular passband frequency (in radians per second).

1.4 Loss function

We note, in the case of the second-order filter (n=2), that $1/G(s)$ is

$$\frac{1}{G(s)} = \frac{\epsilon}{\omega_p^2}s^2 + \sqrt{2}\frac{\epsilon^{1/2}}{\omega_p}s + 1.$$

Since $s = \sigma + j\omega$ and $s\Big|_{\sigma=0} = j\omega$,

$$\frac{1}{G(\omega)} = \frac{\epsilon}{\omega_p^2}j^2\omega^2 + \sqrt{2}\frac{\epsilon^{1/2}}{\omega_p}j\omega + 1 = -\epsilon\left(\frac{\omega}{\omega_p}\right)^2 + \sqrt{2\epsilon}\frac{\omega}{\omega_p}j + 1$$

$$= \left(-\epsilon\left(\frac{\omega}{\omega_p}^2 + 1\right) + j\sqrt{2\epsilon}\frac{\omega}{\omega_p}\right),$$

and has magnitude

$$\left|\frac{1}{G(\omega)}\right| = \sqrt{\epsilon^2\left(\frac{\omega}{\omega_p}\right)^4 - 2\epsilon\left(\frac{\omega}{\omega_p}\right)^2 + 1 + 2\epsilon\left(\frac{\omega}{\omega_p}\right)^2}$$

$$= \sqrt{1+\epsilon^2\left(\frac{\omega}{\omega_p}\right)^4} = \sqrt{1+\epsilon^2\left(\frac{\omega}{\omega_p}\right)^{2n}}.$$

This is an expression for the voltage *loss* rather than gain in terms of ϵ, ω, and n. Converting this value to decibels and denoting the result as $A(\omega)$ yields equation 1.

$$A(\omega) = 10\log_{10}\left[1+\epsilon^2\left(\frac{\omega}{\omega_p}\right)^{2n}\right]. \tag{1}$$

(Note that $A(\omega)$ is a real rather than complex value.) This was an informal verification for the case of n = 2, but equation 1 is valid for a filter of any order.

1.5 Filter parameters

By taking equation 1 and solving for the case of $\omega = \omega_p$ it can be shown that

$$\epsilon = \sqrt{10^{.1A_{max}} - 1}. \tag{2}$$

which will be important in matching our filter to the requirements.

We also need to know what order of filter is required. This is found by using equation 3 (which, again, can be derived from equation 1.

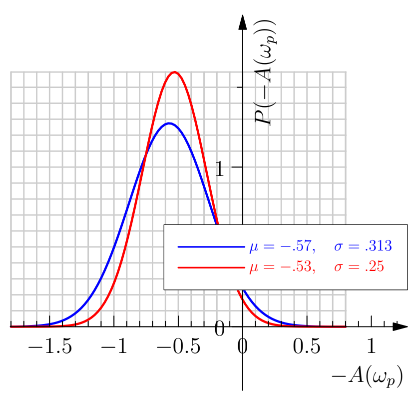
$$n = \frac{\log_{10}\left(\frac{10^{.1A_{min}}-1}{\epsilon^2}\right)}{\log_{10}\left(\frac{\omega_s}{\omega_p}\right)^2} \tag{3}$$

1.6 Component tolerances and random sampling

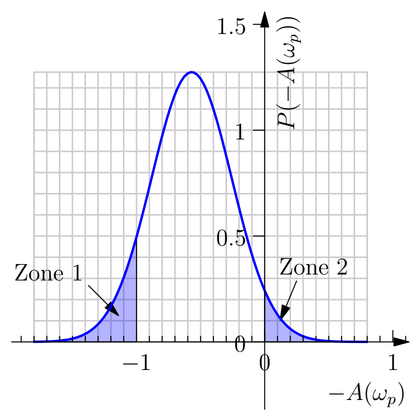
Our filter is to be implemented using an active RC electricial circuit. All electrical resistors and capacitors have a *tolerance*, which is the fraction by which the component value may deviate from its nominal value. A resistor with a nominal value of 1kΩ, for example, is not exactly 1kΩ. It's value, rather, is — e.g. — 1k$\Omega \pm 10\%$, 1k$\Omega \pm 2\%$, or 1k$\Omega \pm 1\%$, depending on the tolerance.

Unsurprisingly, these variations in component values will have a significant effect on the performance of our filter. Instead of performing a single simulation, we will perform a large number of simulations and calculate the mean and standard deviation values. The deviation about the mean can then be considered graphically by plotting equation 4 (Figure 2).

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{4}$$



(a) Distributions with different values of μ and σ



(b) Areas of failure

Figure 2: Probability density function

The gain of our filter must lie between -1 and 0 dB at the passband frequency. The percentage failure of our circuit builds will be determined by calculating the area under the curve that lies outside this range (highlighted in Figure 2b).

The area is more easily calculated by converting this distribution to a unit normal distribution — i.e. a distribution with $\mu = 0$ and $\sigma = 1$. Qualitatively, this is done by 'stretching' the distribution appropriately, and translating it such that the mean of the distribution is at zero. Quantitatively, we define a parameter z as given by equation 5.

$$z = \left|\frac{x-\mu}{\sigma}\right| \tag{5}$$

The probability distribution, then, is given by equation **??**,

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \tag{6}$$

and the required areas are determined by integrating this expression over the appropriate range.

2 Attempt 1

We begin by assuming that a filter with a loss of 1dB in the pass-band will be acceptable, and derive the filter parameters accordingly. We will then use Spice to perform the simulations.

2.1 Butterworth Approximation Function

We begin by determining the required value of ϵ using equation 2.

$$\epsilon = \sqrt{10^{.1A_{max}} - 1} = \sqrt{10^{1/10} - 1} = .50885$$

The loss in the stop-band is

$$A_{min} = 10 \log_{10} \left[1 + \epsilon^2 \left(\frac{\omega_s}{\omega_p} \right)^{2n} \right].$$

Thus,

$$10^{.1A_{min}} = 1 + \epsilon^2 \left(\frac{\omega_s}{\omega_p} \right)^2,$$

$$\left(\frac{\omega_s}{\omega_p} \right)^{2n} = \frac{10^{.1A_{min}} - 1}{\epsilon^2},$$

and

$$n = \frac{\log_{10} \left(\frac{10^{.1A_{min}} - 1}{\epsilon^2} \right)}{\log_{10} \left(\frac{\omega_s}{\omega_p} \right)^2} = 1.7215.$$

Consequently, we need a *second-order* filter.

2.2 Second-order Butterworth filter

The normalised form of a second-order Butterworth low-pass filter is

$$G(s) = \frac{1}{s^2 + \sqrt{2}s + 1}.$$

To *denormalize* the function we replace s by $\frac{\epsilon^{1/n}}{\omega_p}$. If we denote $\frac{\epsilon^{1/n}}{\omega_p}$ as A, then

$$G(s) = \frac{1}{A^2 s^2 + \sqrt{2}As + 1},$$

or

$$G(s) = \frac{1/A^2}{s^2 + \frac{\sqrt{2}}{A}s + \frac{1}{A^2}}. \tag{7}$$

2.3 Sallen-Key circuit topology

We will implement this filter using the active RC circuit shown in Figure 3. This is known as a *Sallen-Key topology*. The transfer function of the circuit — derived fully in Appendix A — is

$$\boxed{G(s) = \frac{\frac{k}{R_1R_2C_1C_2}}{s^2 + s\left(\frac{1-k}{R_2C_2} + \frac{1}{R_2C_1} + \frac{1}{R_1C_1}\right) + \frac{1}{R_1R_2C_1C_2}}}. \tag{8}$$

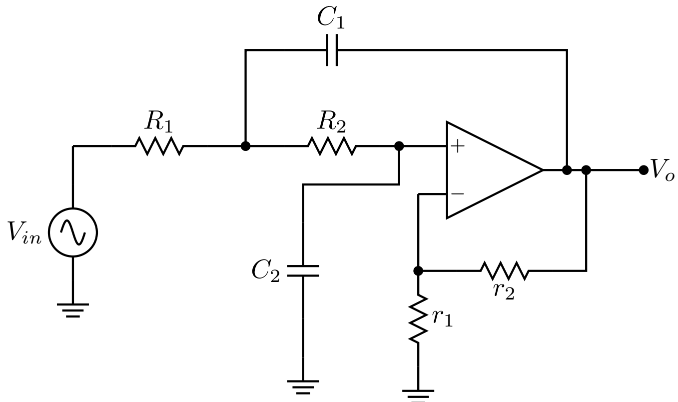


Figure 3: Active RC filter (Sallen-Key topology)

2.4 Coefficient matching

By matching coefficients in equations 8 and 7 we can determine the component values for our active RC circuit.

$$A^2 = R_1R_2C_1C_2$$

$$\frac{\sqrt{2}}{A} = \frac{1-k}{R_2C_2} + \frac{1}{R_2C_1} + \frac{1}{R_1C_1}$$

Let $R_1 = R_2 = 1\text{k}\Omega$. Let $k = 1$.

$$\therefore A^2 = (1 \cdot 10^6\Omega^2)C_1C_2$$

$$\frac{\sqrt{2}}{A} = \frac{1}{(1\text{k}\Omega)C_1} + \frac{1}{(1\text{k}\Omega)C_1} = \frac{2}{(1 \cdot 10^3\Omega)C_1}$$

$$\therefore C_1 = \frac{2}{(1 \cdot 10^3\Omega)}\frac{A}{\sqrt{2}} = 161\text{nF}$$

$$C_2 = \frac{A^2}{(1 \cdot 10^6\Omega^2)C_1} = 80.3\text{nF}$$

$$\therefore R_1 = 1\text{k}\Omega, \quad R_2 = 1\text{k}\Omega, \quad C_1 = 161\text{nF}, \quad C_2 = 80\text{nF}$$

2.5 Results

Table 1 shows the results. The z parameters were calculated using equation 5 and are essential for determining the failure in each zone (Table 2).

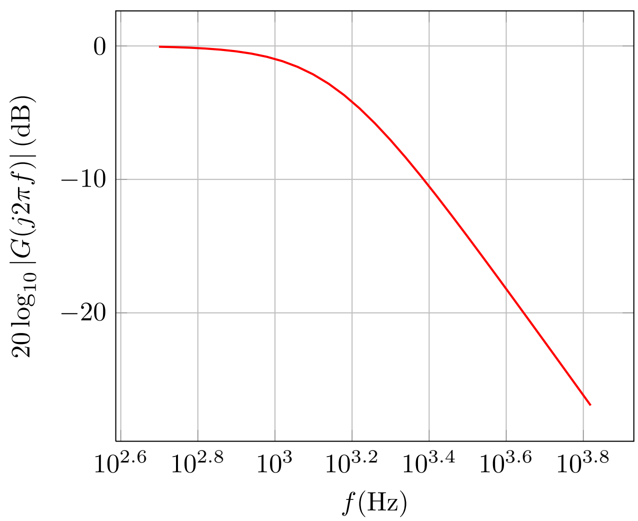


Figure 4: Magnitude frequency response of 2nd-order filter (attempt 1)

tol_R [%]	tol_C [%]	Passband				Stopband		
		μ	σ	$z_{-1\text{dB}}$	$z_{0\text{dB}}$	μ	σ	z
10	20	-1.020	0.653	.03009	1.5626	-18.20	1.62	1.977
10	10	-.9821	0.358	.04995	2.7421	-18.24	0.993	3.264
2	10	-.9858	0.334	.04246	2.9475	-18.30	0.718	4.592
1	10	-.9815	0.322	.05738	3.0438	-18.28	0.710	4.620
1	5	-.9788	0.160	.13285	6.1284	-18.31	0.363	9.121
1	2	-.9760	0.065	.37272	14.898	-18.29	0.152	21.664
1	1	-.9737	0.036	.72969	26.963	-18.29	0.101	32.710

Table 1: Results for 2nd-order filter

2.6 Analysis

Based on the results shown in Table 1 we can calculate the percentage failure for each set of tolerances. This is shown in Table 2. Evidently, the failure rate is unacceptable regardless of what component tolerances are used.

The failure is due to our assumption that designing our filter with a loss of 1dB in the pass-band would be unproblematic. In retrospect this was foolish. 1dB is the *maximum* acceptable loss in the pass-band. Considering that the acceptable range of loss in the pass-band is between 0 and 1 dB, we should design the filter to have a loss in the *middle* of this range — i.e. 0.5 dB. This will be our approach in our second attempt at filter design (Section 3).

tol_R [%]	tol_C [%]	Zone 1 [%]	Zone 2 [%]	Zone 3 [%]	**Total [%]**
10	20	48.8	5.9	2.4	57.1
10	10	48.1	0.3	0.1	48.5
2	10	48.3	0.2	0.0	48.5
1	10	47.7	0.0	0.0	47.7
1	5	44.8	0.0	0.0	44.8
1	2	35.5	0.0	0.0	35.5
1	1	23.3	0.0	0.0	23.3

Table 2: Failure zones for 2nd-order filter

3 Attempt 2

We learnt in our first attempt that — in order to meet the design requirements — we should set the loss in the pass-band to approximately 0.5 dB. If $A_{max} = 0.5$, then

$$\epsilon = \sqrt{10^{.1A_{max}} - 1} = \sqrt{10^{.05} - 1} = .349.$$

To obtain the desired loss in the stop-band, we use equation 3 . If we select a stop-band loss only equal to the requirement, certain circuit builds will fail to meet the requirements due to variations in component values. Consequently, we will aim for a larger loss value, e.g. 20 dB. Then,

$$n = \frac{\log_{10}\left(\frac{10^{.1(20)}-1}{\epsilon^2}\right)}{\log_{10}\left(\frac{\omega_s}{\omega_p}\right)^2} = 2.42$$

Thus, a 3rd-order filter is needed.

The frequency response required is shown in Figure 5, and be compared with that of the second-order filter implemented previously.

Such a filter can be obtained by simply appending a first-order RC circuit to the output of the sallen-key circuit. The result is shown in Figure 6 .

$$A = \frac{\epsilon^{1/n}}{\omega_p} = \frac{\epsilon^{1/3}}{2\pi(1000\text{Hz})} = 1.12 \cdot 10^{-4}$$

3.1 Coefficient matching

The normalised form of a third-order Butterworth low-pass filter is

$$G(s) = \frac{1}{(s^2 + s + 1)(s + 1)}.$$

To denormalize the function we replace s with As, where — as in the case of the 2nd-order filter — $A = \frac{\epsilon^{1/n}}{\omega_p}$. Then,

$$G(s) = \left(\frac{1/A^2}{s^2 + \frac{1}{A}s + \frac{1}{A^2}}\right) \cdot \left(\frac{1/A}{s + 1/A}\right).$$

Based on the transfer function of the RC circuit, it is evident that

$$\frac{1/A}{s + 1/A} = \frac{\frac{1}{R_3C_3}}{s + \frac{1}{R_3C_3}}.$$

$$\therefore A = R_3C_3$$

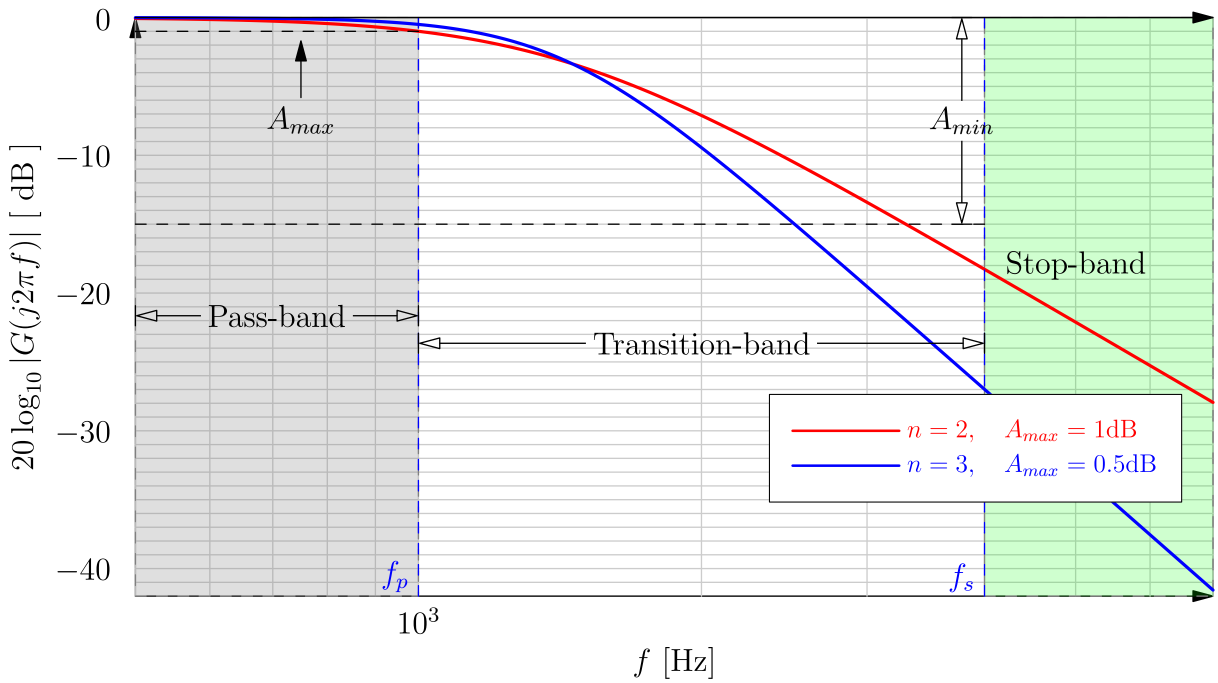


Figure 5: Comparison of Butterworth filters of different order and pass-band gain

Let $R_3 = 1\text{k}\Omega$.

$$\therefore C_3 = \frac{A}{1\text{k}\Omega} = 1.12 \cdot 10^{-7}\text{F} = 112 \cdot 10^{-9}\text{F} = 112\text{nF}$$

Also,

$$R_1 R_2 C_1 C_2 = A^2 \tag{9}$$

and

$$\frac{1}{A} = \frac{1-k}{R_2 C_2} + \frac{1}{R_2 C_1} + \frac{1}{R_1 C_1} = \frac{1}{R_2 C_1} + \frac{1}{R_1 C_1}.$$

$$\therefore \frac{R_1 + R_2}{R_1 R_2 C_1} \tag{10}$$

If we let $R_1 = R_2 = 1\text{k}\Omega$ and substitute this value into equation 10, we find that

$$C_1 = 224\text{nF}.$$

Substituting this value in turn in to equation 9 yields

$$C_2 = 56.1\text{nF}.$$

Thus, the component values for our third-order filter are

R_1	R_2	R_3	C_1	C_2	C_3
1 kΩ	1 kΩ	1 kΩ	224 nF	56.1 nF	112 nF

3.2 Results

The z values in Table 3 were used to calculate the fail values shown in Table 4.

Failure in the stopband is neglibile for all resistor and capacitor tolerances.

The final circuit is shown in Figure 9.

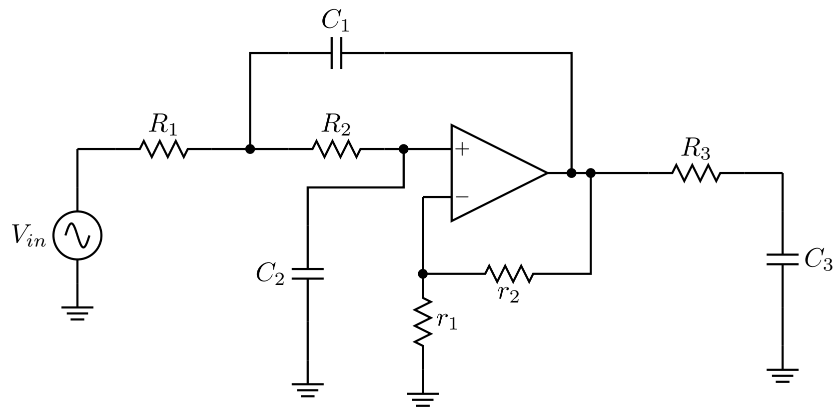


Figure 6: Active RC circuit implementation of a third-order filter

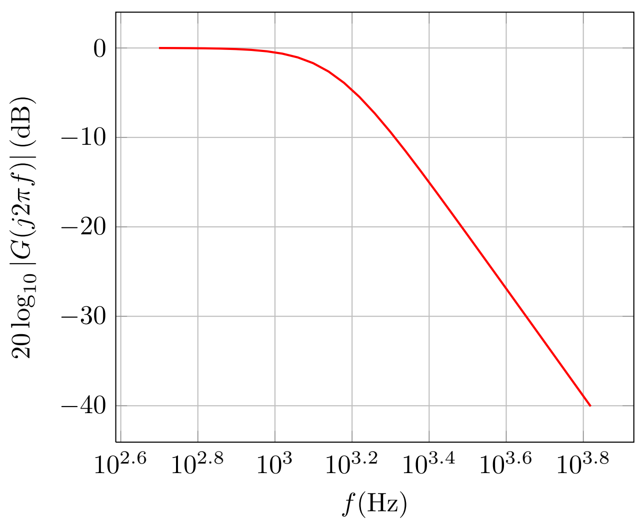
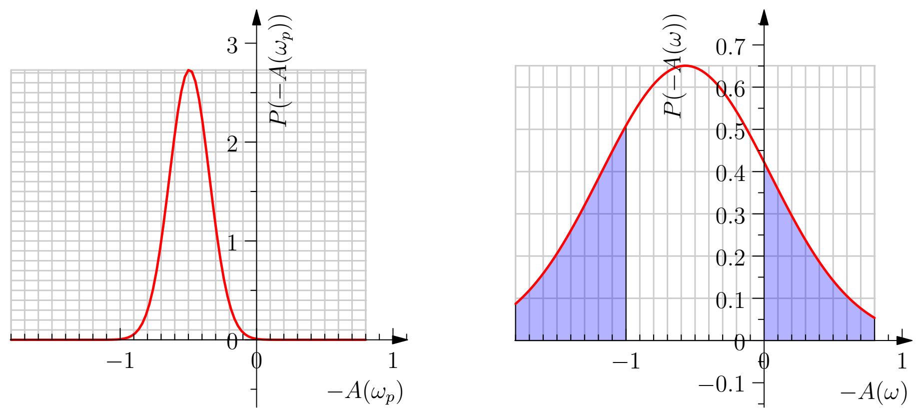


Figure 7: Magnitude frequency response of 3rd-order implementation

tol_R [%]	tol_C [%]	Passband				Stopband		
		μ	σ	$z_{-1\text{dB}}$	$z_{0\text{dB}}$	μ	σ	z
10	20	-.57	.613	.7007	.9293	-26.82	1.979	5.973
10	10	-.494	.321	1.576	1.537	-26.90	1.273	9.351
2	10	-.501	.295	1.694	1.700	-26.98	.910	13.167
1	10	-.514	.295	1.647	1.742	-26.95	.892	13.399
1	5	-.492	.146	3.490	3.378	-27.01	.457	26.256
1	2	-.495	.060	8.373	8.215	-27.04	.196	61.297
1	1	-.494	.032	15.627	15.264	-27.04	.120	100.340

Table 3: Results for 3rd-order filter



(a) $tol_R = 1\%$, $tol_C = 5\%$ (b) $tol_R = 10\%$, $tol_C = 20\%$

Figure 8: Failure rates in the passband area for different component tolerances

tol_R [%]	tol_C [%]	Zone 1 [%]	Zone 2 [%]	Zone 3 [%]	**Total [%]**
10	20	24.2	17.7	0.0	**41.9**
10	10	5.7	6.2	0.0	**11.9**
2	10	4.5	4.5	0.0	**9.0**
1	10	5.0	4.1	0.0	**9.1**
1	5	0.0	0.0	0.0	**0.0**

Table 4: Failure Rates for 3rd-order Filter

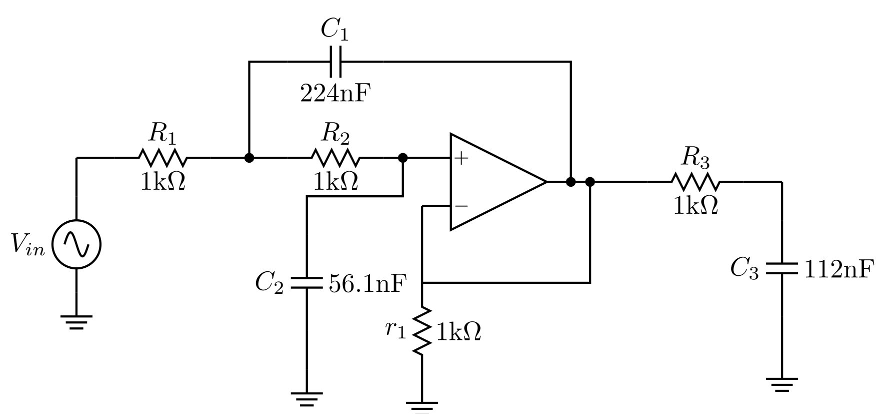


Figure 9: Final circuit

4 Cost Analysis

A brief cost analysis was performed — on the assumption that surface mount components would be used. It was also assumed that the manufacturer would have discretion over which component packaging to use (0603, 0402, 0805, etc.). Ceramic capacitors were chosen.

tol_R	tol_C	Resistor unit cost [cent]	Capacitor unit cost [cent]	Total cost [euro]
1 %	5 %	.7	1	.58m
1 %	2 %	.7	2.5	1.06m
1 %	1 %	.7	2.6	1.06m

As can be seen from the table, using capacitors of lower tolerance could significantly increase the costs of production. Considering that 5% capacitors already meet the specification, there is no justification for using them.

The total cost is that of the passive components — i.e. excluding the operational amplifier, PCB production, soldering, or assembly (as these were assumed to be independent of the resistor / capacitor tolerances chosen).

5 Summary

We have learnt a number of things from these attempts at filter design:

1. The filter should be designed to meet the requirements with a comfortable margin.
2. Although an ideal filter is not possible, requirements can usually be met by selecting a filter of the appropriate order.
3. Although digital filters are increasingly used nowadays, analog filters still play important roles in many instances.
4. Component tolerances have a significant bearing on filter performance (or, indeed, on the performance of any electrical circuit). Their effect should be considered when planning large production runs, and Monte Carlo analysis is an effective way of doing this.

A more comprehensive analysis would have also accounted for the characteristics of the amplifier.

A Transfer Function of 2nd-order Active RC Filter

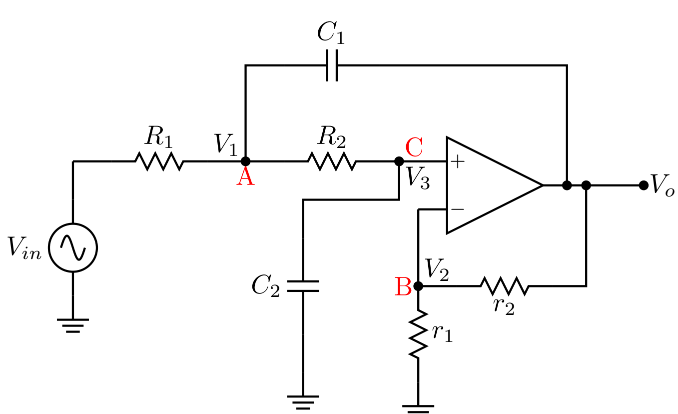


Figure 10: Active RC filter

Referring to Figure 10, nodal analysis can be used to find the voltage transfer function of the circuit. We want to find the ratio of the output voltage V_o to the input voltage V_i.

At node A, Kirchoff's current law yields the following.

$$\frac{V_1}{R_1} - \frac{V_i}{R_1} + (V_1 - V_o)sC_1 + \frac{V_1 - V_3}{R_2} = 0 \tag{11}$$

The voltage at the two input terminals on the op-amp are effectively the same. Consequently,

$$V_3 = V_2 = V_o \frac{r_1}{r_1 + r_2} = V_o \frac{r_1}{r_1 + r_1(k-1)} = \frac{1}{k} V_o.$$

Substituting this value for V_3 back into equation 11, we have

$$\frac{V_1}{R_1} - \frac{V_i}{R_1} + (V_1 - V_o)sC_1 + \frac{V_1 - \frac{V_o}{k}}{R_2} = 0 \tag{12}$$

$$\frac{V_3 - V_1}{R_2} + V_3 sC_2 = 0$$

$$\frac{\frac{V_o}{k} - V_1}{R_2} + \frac{V_o}{k} sC_2 = 0$$

$$\frac{V_o}{kR_2} + \frac{sC_2}{k} V_o = \frac{V_1}{R_2}$$

$$\therefore \frac{V_1}{R_2} = V_o \left(\frac{1}{kR_2} + \frac{sC_2}{k} \right)$$

$$\therefore V_1 = \frac{V_o}{k}(1 + sR_2C_2)$$

Substitute into 12 .

$$\frac{\frac{V_o}{k}(1 + sR_2C_2)}{R_1} - \frac{V_i}{R_1} + \left(\frac{V_o}{k}(1 + sR_2C_2) - V_o \right) sC_1 + \frac{\frac{V_o}{k}(1 + sR_2C_2) - \frac{V_o}{k}}{R_2} = 0$$

$$\therefore \frac{\frac{V_o}{k} + \frac{V_o}{k} sR_2C_2}{R_1} - \frac{V_i}{R_1} + \left(\frac{V_o}{k} + \frac{V_o}{k} sR_2C_2 - V_o \right) sC_1 + \frac{\frac{V_o}{k} sR_2C_2}{R_2} = 0$$

$$\frac{V_o}{kR_1} + \frac{V_o sR_2C_2}{kR_1} - \frac{V_i}{R_1} + \frac{V_o sC_1}{k} + \frac{V_o s^2 R_2C_1C_2}{k} - V_o sC_1 + \frac{V_o}{k} sC_2 = 0$$

$$\therefore V_o + V_o sR_2C_2 - kV_i + V_o sR_1C_1 + V_o s^2 R_1R_2C_1C_2 - kV_o sR_1C_1 + V_o sR_1C_2 = 0$$

$$\therefore V_o \left(1 + s(R_1C_1(1-k) + R_1C_2 + R_2C_2) + s^2 R_1R_2C_1C_2 \right) = kV_i$$

$$\therefore G(s) = \frac{V_o(s)}{V_i(s)} = \frac{k}{s^2 R_1R_2C_1C_2 + s(R_1C_1(1-k) + R_1C_2 + R_2C_2) + 1}$$

$$\therefore \boxed{G(s) = \frac{\frac{k}{R_1R_2C_1C_2}}{s^2 + s\left(\frac{1-k}{R_2C_2} + \frac{1}{R_2C_1} + \frac{1}{R_1C_1} \right) + \frac{1}{R_1R_2C_1C_2}}}$$

B SPICE Analysis

Ngspice was used to perform circuit simulations. The netlisting is shown in Listing 1, and the script used to facilitate the Monte Carlo analysis is shown in Listing 2.

A large number of simulations were performed for each pair of tolerances, in order to allow values of the mean and deviation to settle or converge. A single-frequency AC analysis was performed on each run rather than an AC sweep over a range of frequencies. This was purely to reduce the analysis time.

Listing 1: SPICE netlist

```
3rd-order low-pass filter

.include 'opamp741.inc'

.param R = 1k
.param C1 = 224n
.param C2 = 56.1n
.param C3 = 112n

V1 in 0 dc 0 ac 1
R1 in 2 {R}
R2 2 + {R}
R3 - 0 {R}
R4 3 - .0001
C2 + 0 {C2}
C1 2 3 {C1}
x1 + - 3 opamp741
R5 3 out {R}
C3 out 0 {C3}

.control
        ac dec 25 500 7k
        wrdata 3rd-order-filter vdb(out)
.endc

.end
```

Listing 2: SPICE script

```
.include '3rd-order-filter.cir'

.options nopage noacct

.control
        let runs = 1000
        let run = 0
        let passband = unitvec(runs)
        let stopband = unitvec(runs)

        define unif(nom, var) (nom + nom*var * sunif(0))

        while run < runs
                alter R1 = unif(1e+3, .01)
```

```
            alter R2 = unif(1e+3, .01)
            alter R3 = unif(1e+3, .01)
            alter R5 = unif(1e+3, .01)
            alter C1 = unif(224e-9, .01)
            alter C2 = unif(56.1e-9, .01)
            alter C3 = unif(112e-9, .01)

            ac lin 1 1k 1k
            print vdb(out)
            let passband[run] = vdb(out)

            ac lin 1 4k 4k
            print vdb(out)
            let stopband[run] = vdb(out)

            let run = run + 1
        end

        print passband >results/3rd-order/passband-R1-C1.dat
        print stopband >results/3rd-order/stopband-R1-C1.dat

.endc

.end
```