Julia Frothingham

Lab 9

PART 1 – RLC CIRCUIT AND RESONANCE

Figure 1. Calculations to determine what size inductor should be used.

$\omega_0 = \sqrt{\frac{1}{LC}}$ $\quad C = 27.84\ nF$

$\omega_0 = (2\pi)(10{,}000\ Hz)$

$\omega_0^2 = \frac{1}{LC}$

$LC\omega_0^2 = 1 \Rightarrow L = \frac{1}{C\omega_0^2} \Rightarrow L = 0.0090985258\ H = 9.1\ mH$

component used: $L = 8.43\ mH$

$\omega_0 = \sqrt{\frac{1}{(27.84\times10^{-9}\ F)(8.43\times10^{-3}\ H)}} = 65275.40894\ rad/s$

$f_0 = \omega_0/2\pi \Rightarrow f_0 = 10388.95\ Hz$

$f_0 = 10.4\ kHz$

I measured the capacitance of the capacitor used and calculated the inductance necessary for a resonant frequency of 10kHz, then examined the available inductors and chose one with a similar value. I measured its inductance and re-calculated the circuit's resonant frequency to ensure it was still within a reasonable range.

Figure 2. Circuit diagram.

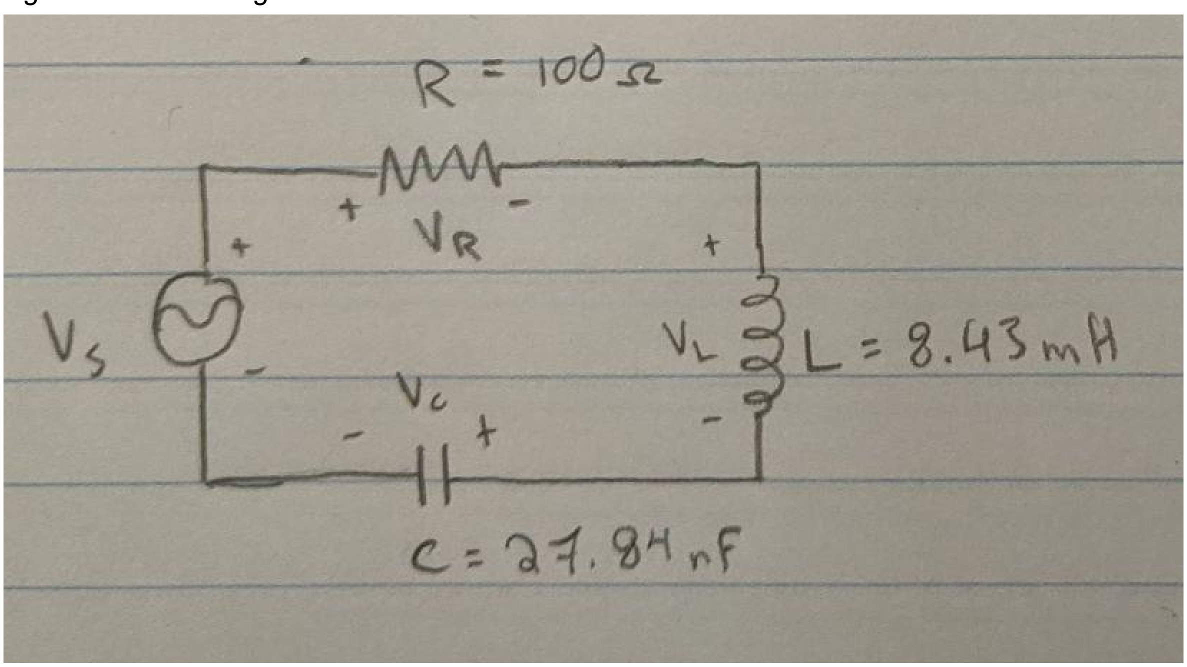
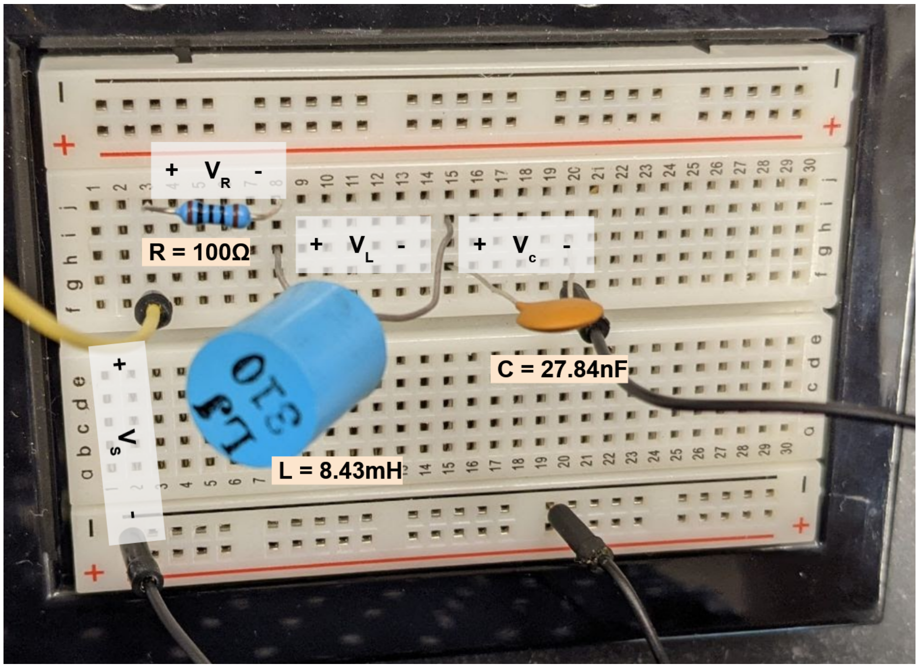


Figure 3. Built circuit.



Measurements were taken using Scopy's network analyzer, with Channel 1 of the wave generator supplying V_S, Channel 1 of the measurement probes (orange) measuring V_S, and Channel 2 of the measurement probes (blue) measuring V_{out}. Data was collected with $V_{out} = V_R$, $V_{out} = V_C$, and $V_{out} = V_L$.

Figure 4. Measurement setup where $V_R = V_{out}$

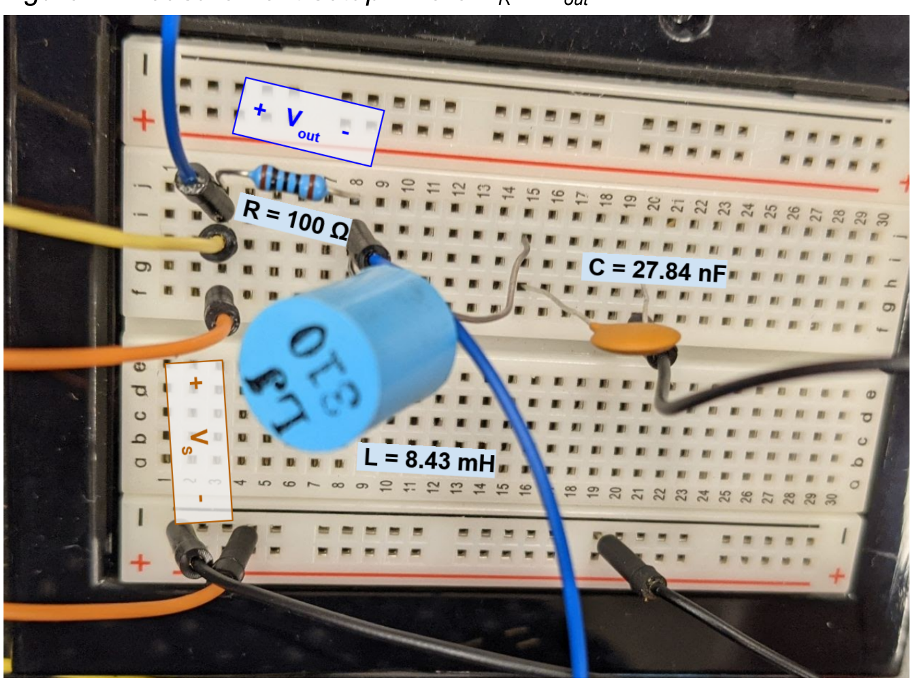


Figure 5. Gain and phase difference plotted against frequency for each component.

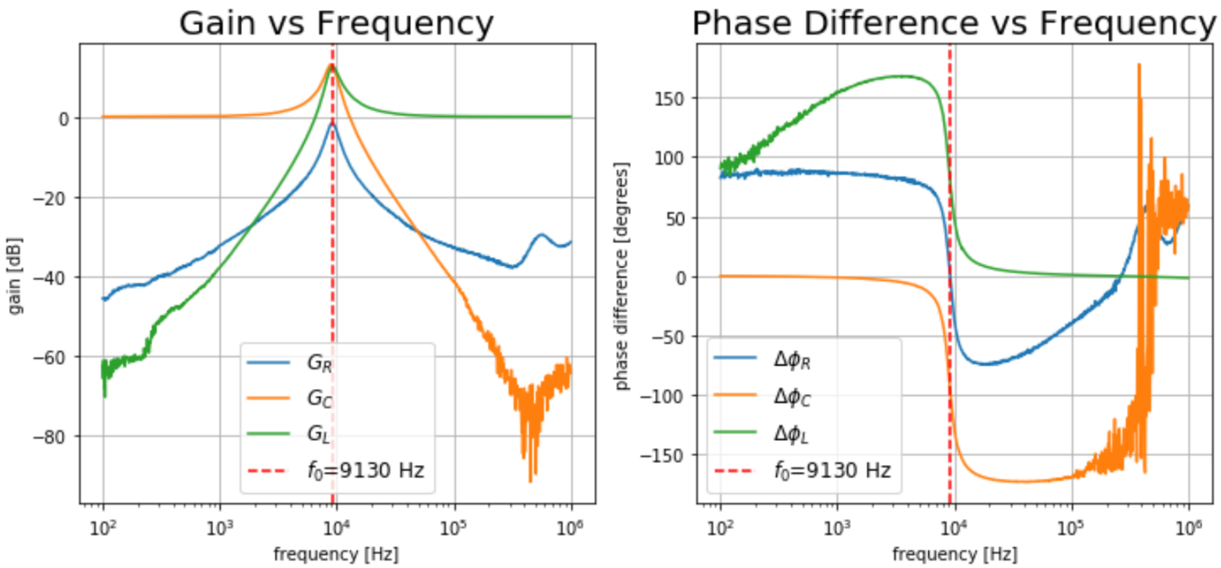
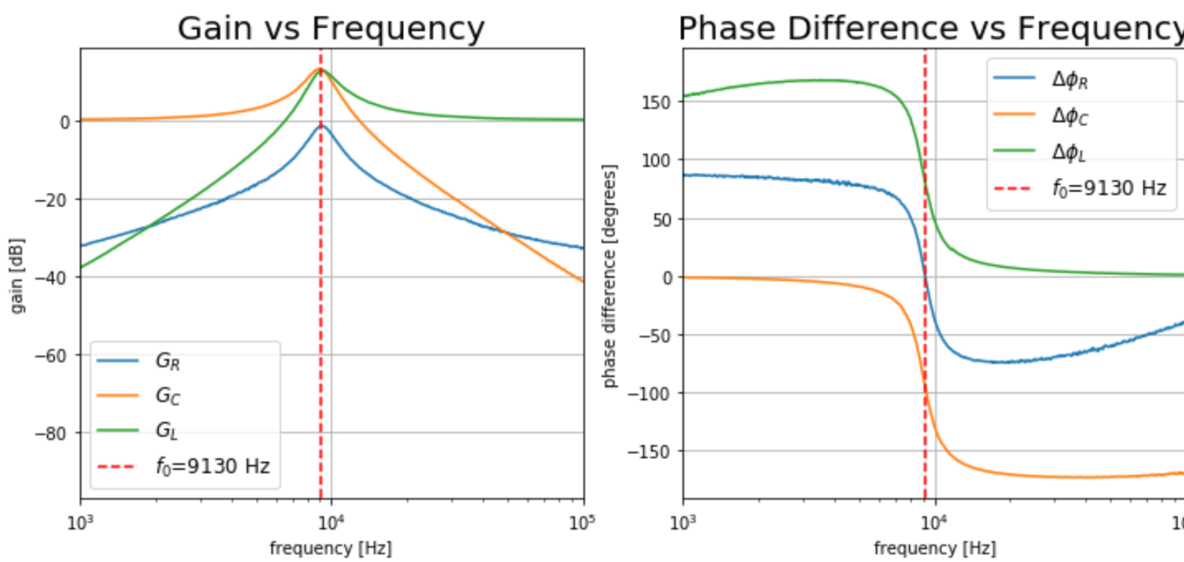


Figure 6. Gain and phase difference plotted against frequency for each component, showing region near fundamental frequency.



In the lower frequency portion of the gain graphs, the capacitor dominates. Capacitors in series with resistors are low-pass filters, meaning they allow lower-frequencies to pass through with less loss than higher-frequencies. The reactance of a capacitor is inversely proportional to the frequency, so lower frequencies will result in higher reactance.

In the higher frequency portion of the gain graphs, the inductor dominates. Inductors in series with resistors are high-pass filters, meaning they allow higher-frequencies to pass through with less loss than lower-frequencies. The reactance of an inductor is proportional to the frequency, so higher frequencies will result in higher reactance.

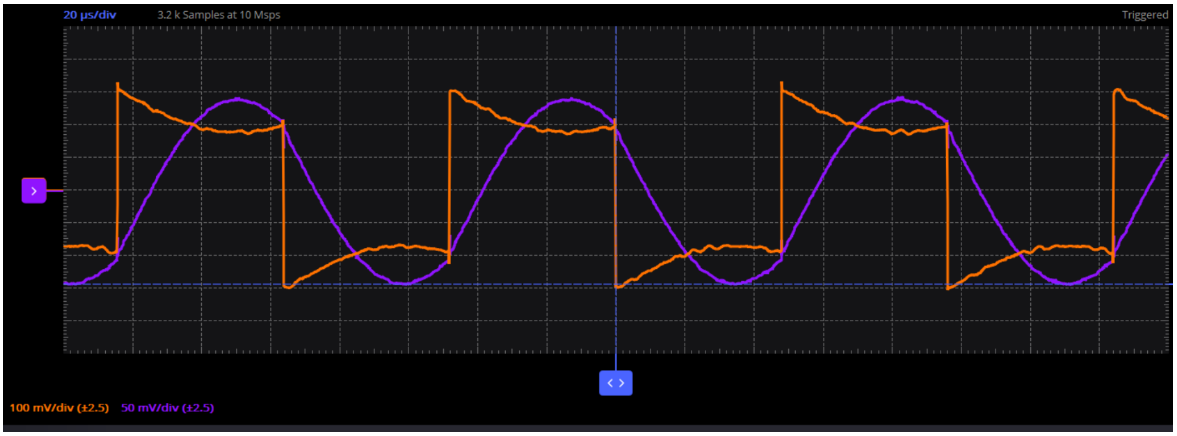
Near the resonant frequency f_0, both V_L and V_C have positive gain. By Kirchhoff's Voltage Law, the sum of each voltage in the circuit must be equal to zero. It does not violate Kirchhoff's Voltage Law for both voltages to have positive gain because V_L and V_C are out of phase with each other by 180° at the fundamental frequency. This means that their voltage contributions cancel each other out and contribute a net sum of approximately 0V towards the KVL sum.

Figure 7. V_L (shown in orange), V_C (shown in purple), and their sum (shown in green) near critical frequency. Measured and plotted in Scopy's oscilloscope.

At the fundamental frequency f_0, the peak V_R gain value is less than 0dB because the real circuit includes some internal resistance from the power source. So the voltage used as V_{in} to the circuit, which drives V_R, is not equal to the reference voltage V_S. At the fundamental frequency, the voltage across the resistor V_{out} may be equal to V_{in}, but since gain is calculated with V_S rather than V_{in}, the gain ratio is not going to exactly equal 1 and so the gain will not be exactly 0dB.

Driving the circuit with a small-amplitude square wave near the resonant frequency produces a sine-wave output from the resistor. However, measuring the input voltage shows a square wave with an unusual shape. The peak amplitude decays over the width of each wave, with the lowest amplitude corresponding to the highest point of the output voltage wave. This would suggest that the power source is loading the circuit - it is itself drawing power because the internal resistance is larger than or close to the resistance of the circuit.

Figure 8. V_{in} (shown in orange) and V_{out} (shown in purple) near f_0. R=100Ω.



If that is the case, then the effects of loading on the circuit should be more noticeable if the resistance of the circuit is decreased, and less noticeable if the resistance of the circuit is increased. Modifying the circuit and measuring the same quantities with different resistance values supports this.

Figure 9. V_{in} and V_{out} for the same circuit near f_0 where R=20Ω (left) and R=470Ω (right).

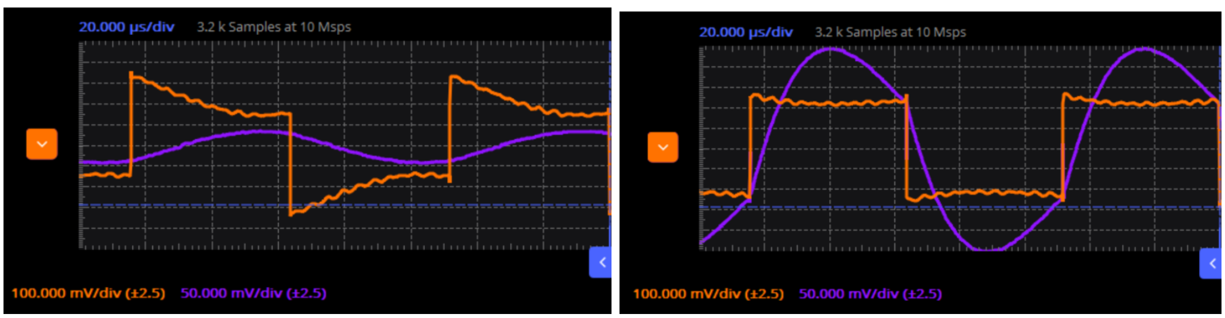
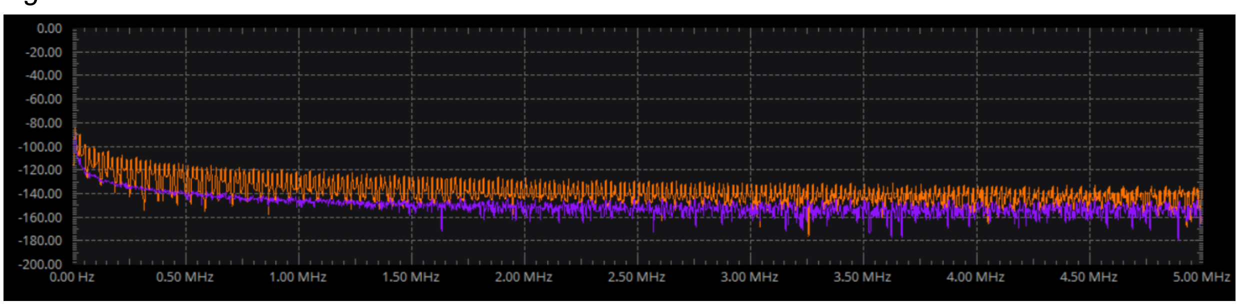


Figure 10. FFT for the circuit where R=100Ω.



The FFT shows that overall, the amplitude of lower-frequency components is higher than the amplitude of higher-frequency components. There are many, mostly regularly-spaced frequencies present in both V_{in} and V_{out}. This is because the input square wave is represented by a Fourier series consisting of many components with frequencies that are multiples of the fundamental frequency and different amplitudes. It makes sense that the lower frequency components, those closest to the frequency of the square wave, would have higher amplitudes than the extremely high frequency components. It also makes sense that V_{out} would have lower overall amplitudes compared to V_{in}, because V_{out} is less than V_{in}.

PART 2 – RLC RESONANCE quantified by Quality factor Q

Figure 11. Gain and phase difference plotted against frequency for several resistances.

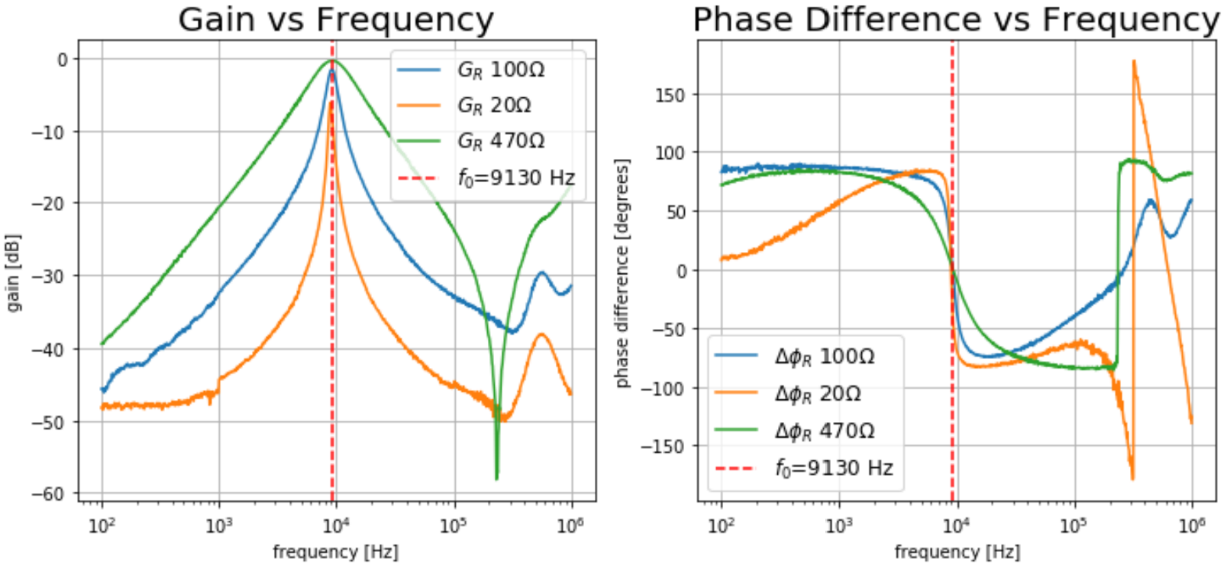
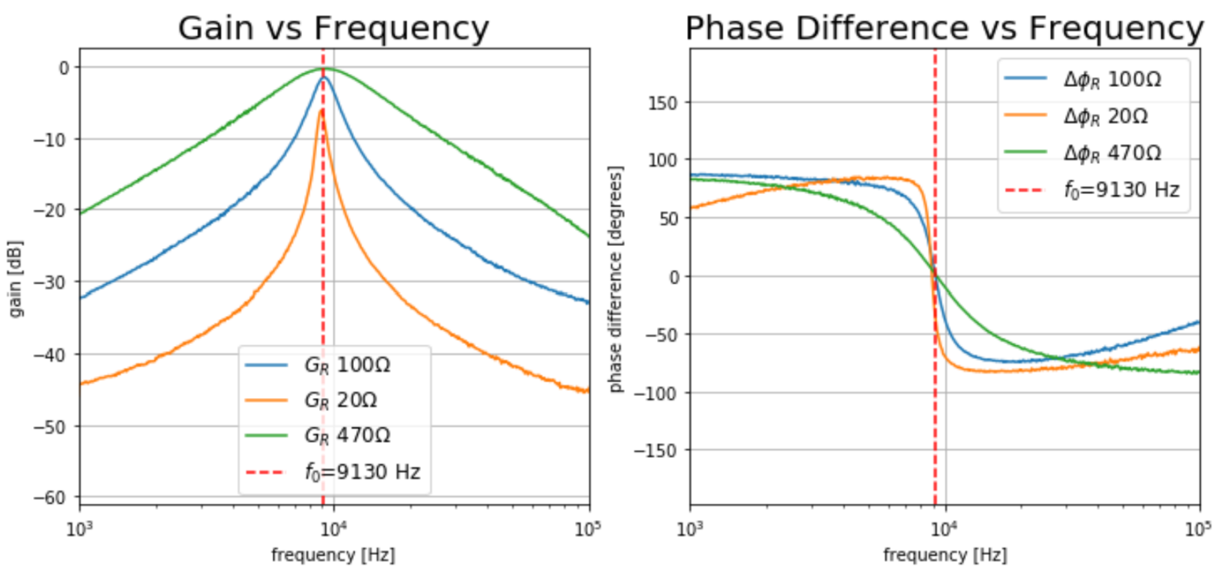


Figure 12. Gain and phase difference plotted against frequency for each resistance, showing region near fundamental frequency.



The three gain graphs all display a similar shape - at lower frequencies, they increase, with increasing slope, up to a peak at the resonant frequency, and then decrease, with decreasing slope, at higher frequencies. They differ in terms of gain size and slope. The graph for R=470Ω has gain values higher than the other two graphs, and the peak gain is closest to 0dB of the three graphs, but its slope is the shallowest and its peak is widest. In contrast, the graph for R=20Ω has the lowest overall gain values and a peak gain furthest from 0dB, but it has the steepest slope and the narrowest peak. The graph for R=100Ω is between these two extremes in all respects.

The circuit with a resistance R=470Ω would produce the largest amplitude V_{out} at the resonant frequency. The plots show that it has the largest gain at the resonant frequency, so it also comes the closest to the original amplitude of V_{in}.

For an input source with a component at f_0 and another component at $f_0/2$, the circuit with a resistance R=20Ω would be most selective in filtering out the lower frequency. Assuming selectivity is the difference in gain (in dB) between f_0 and $f_0/2$, the 20Ω plot shows the greatest selectivity. It has the steepest slope approaching the resonant frequency and the larger magnitude difference between f_0 and $f_0/2$.

The quality factor Q can be found in two ways: one is by its definition, which establishes that $Q = \omega_0 / \Delta\omega$ where $\Delta\omega$ is the width of the gain vs frequency plot measured from the two points where the gain is 3dB lower than the peak gain. The other way Q can be found is with the relation $Q = \omega_0 L / R$.

If possible, using the definition of Q is a more reliable method. With a complete graph and dataset of gain vs frequency, it is fairly simple to find Q.

First, I wrote Python code to find the resonant frequency for each set of data collected. This ensures that the correct angular frequency ω_0 will be used.

Figure 13. Python code used to find the resonant frequency for each set of data collected.

```python
# find resonant frequency
f_r_100 = Vr_100_freq[np.argmax(Vr_100_gain)]
print('Resonant frequency, resistor, 100 Ohms:', np.round(f_r_100, 3), 'Hz')
f_c_100 = Vc_100_freq[np.argmax(Vc_100_gain)]
print('Resonant frequency, capacitor, 100 Ohms:', np.round(f_c_100, 3), 'Hz')
f_l_100 = Vl_100_freq[np.argmax(Vl_100_gain)]
print('Resonant frequency, inductor, 100 Ohms:', np.round(f_l_100, 3), 'Hz')
f_r_20 = Vr_20_freq[np.argmax(Vr_20_gain)]
print('Resonant frequency, resistor, 20 Ohms:', np.round(f_r_20, 3), 'Hz')
f_r_470 = Vr_470_freq[np.argmax(Vr_470_gain)]
print('Resonant frequency, resistor, 470 Ohms:', np.round(f_r_470, 3), 'Hz', '\n')

f_avg = (f_r_100 + f_c_100 + f_l_100 + f_r_20 + f_r_470)/5
print('Average resonant frequency:', np.round(f_avg,3), 'Hz')
```

```
Resonant frequency, resistor, 100 Ohms: 9161.4 Hz
Resonant frequency, capacitor, 100 Ohms: 8994.02 Hz
Resonant frequency, inductor, 100 Ohms: 9161.4 Hz
Resonant frequency, resistor, 20 Ohms: 8911.48 Hz
Resonant frequency, resistor, 470 Ohms: 9418.33 Hz

Average resonant frequency: 9129.326 Hz
```

Next, I wrote Python code to find $\Delta\omega$ for each set of collected resistance data. For each dataset, the code determines the maximum gain, then finds the indices of the gain values that are nearest to maximum gain minus 3dB. The sensitivity of what is meant by "near" was manually adjusted until the code produced the indices of two frequencies, one above and one below the resonant frequency. $\Delta\omega$ is equal to the difference between the frequencies at those indices.

Technically speaking, the code shown in Figure 14 actually finds Δf, not Δω, as the conversion from Hz to radians/sec does not happen until later.

Figure 14. Python code used to find Δω for each set of resistance data collected.

```python
# find delta omega

print('20 OHM RESISTOR')
peak_r20 = np.max(Vr_20_gain)
r20_omega_indices = np.asarray(np.where(np.abs(Vr_20_gain - (peak_r20 - 3)) < 0.5))
print('frequncy indices', r20_omega_indices)
r20_delta_omega = Vr_20_freq[r20_omega_indices[0,1]] - Vr_20_freq[r20_omega_indices[0,0]]
print('f2:',Vr_20_freq[r20_omega_indices[0,1]], 'Hz')
print('f:', f_r_20, 'Hz')
print('f1:', Vr_20_freq[r20_omega_indices[0,0]], 'Hz')
print('delta f:', r20_delta_omega, 'Hz')
print('\n')

print('100 OHM RESISTOR')
peak_r100 = np.max(Vr_100_gain)
r100_omega_indices = np.asarray(np.where(np.abs(Vr_100_gain - (peak_r100 - 3)) < 0.175))
print('frequncy indices', r100_omega_indices)
r100_delta_omega = Vr_100_freq[r100_omega_indices[0,1]] - Vr_100_freq[r100_omega_indices[0,0]]
print('f2:',Vr_100_freq[r100_omega_indices[0,1]], 'Hz')
print('f:', f_r_100, 'Hz')
print('f1:', Vr_100_freq[r100_omega_indices[0,0]], 'Hz')
print('delta f:', r100_delta_omega, 'Hz')
print('\n')

print('470 OHM RESISTOR')
peak_r470 = np.max(Vr_470_gain)
r470_omega_indices = np.asarray(np.where(np.abs(Vr_470_gain - (peak_r470 - 3)) < 0.07))
print('frequncy indices', r470_omega_indices)
r470_delta_omega = Vr_470_freq[r470_omega_indices[0,1]] - Vr_470_freq[r470_omega_indices[0,0]]
print('f2:',Vr_470_freq[r470_omega_indices[0,1]], 'Hz')
print('f:', f_r_470, 'Hz')
print('f1:', Vr_470_freq[r470_omega_indices[0,0]], 'Hz')
print('delta f:', r470_delta_omega, 'Hz')
print('\n')
```

```
20 OHM RESISTOR
frequncy indices [[483 492]]
f2: 9331.9 Hz
f: 8911.48 Hz
f1: 8588.83 Hz
delta f: 743.0699999999997 Hz


100 OHM RESISTOR
frequncy indices [[479 502]]
f2: 10233.2 Hz
f: 9161.4 Hz
f1: 8277.86 Hz
delta f: 1955.3400000000001 Hz


470 OHM RESISTOR
frequncy indices [[445 537]]
f2: 14130.3 Hz
f: 9418.33 Hz
f1: 6050.37 Hz
delta f: 8079.929999999999 Hz
```

Finally, I wrote Python code to calculate Q for each collected set of resistance data. It converts the resonant frequency f_0 and Δf in Hz found earlier to ω_0 and $\Delta\omega$ in radians/sec, then takes their ratio. Since it is a ratio and both the numerator and denominator are scaled by the same factor, this conversion is not strictly necessary, but it's good practice.

Figure 15. Python code used to calculate quality factor Q for each set of resistance data.

```
# find Q
# Q = omega / delta omega

print('20 OHM RESISTOR')
omega_r20 = f_r_20 * (2 * np.pi)
delta_omega_r20_rad = r20_delta_omega * (2 * np.pi)
q_r20 = omega_r20 / delta_omega_r20_rad
print('Q =',q_r20, '\n')

print('100 OHM RESISTOR')
omega_r100 = f_r_100 * (2 * np.pi)
delta_omega_r100_rad = r100_delta_omega * (2 * np.pi)
q_r100 = omega_r100 / delta_omega_r100_rad
print('Q =',q_r100, '\n')

print('470 OHM RESISTOR')
omega_r470 = f_r_470 * (2 * np.pi)
delta_omega_r470_rad = r470_delta_omega * (2 * np.pi)
q_r470 = omega_r470 / delta_omega_r470_rad
print('Q =',q_r470, '\n')
```

```
20 OHM RESISTOR
Q = 11.992786682277583

100 OHM RESISTOR
Q = 4.68532326858756

470 OHM RESISTOR
Q = 1.1656449993997473
```

The circuit application determines whether it is better for an RLC circuit to have a larger or smaller resistance. Smaller resistance RLC circuits will have much better selectivity, but a tradeoff is that the effects of loading on the circuit are much more pronounced, and the amplitude of the output voltage will be smaller. For applications in which selectivity is more important than signal strength, smaller resistance is preferable. However, for applications that do not require the best selectivity, a higher-amplitude signal may be preferable, and so a larger resistance would be better.