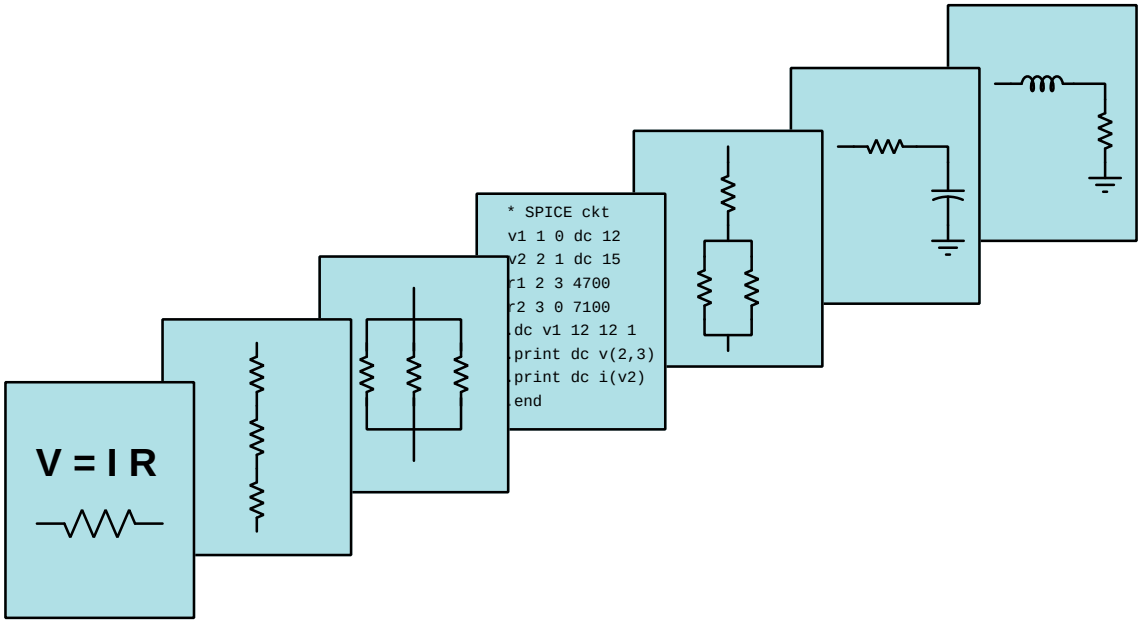
Modular Electronics Learning (ModEL) project



Advanced Filter Circuits

© 2022-2026 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 20 April 2026

This is a copyrighted work, but licensed under the Creative Commons Attribution 4.0 International Public License. A copy of this license is found in the last Appendix of this document. Alternatively, you may visit `http://creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons: 171 Second Street, Suite 300, San Francisco, California, 94105, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.

Contents

1 Introduction **3**
- 1.1 Recommendations for students . . . 3
- 1.2 Challenging concepts related to complex filter networks . . . 5
- 1.3 Recommendations for instructors . . . 6

2 Case Tutorial **7**
- 2.1 Example: 3rd-order low-pass filter design . . . 8
- 2.2 Example: 4th-order high-pass filter design . . . 9

3 Tutorial **11**
- 3.1 Phasor analysis review . . . 12
- 3.2 Basic filter networks . . . 16
- 3.3 Output-limited filter networks . . . 20
- 3.4 Balanced versus unbalanced filter networks . . . 23
- 3.5 Filter network topology . . . 24
- 3.6 Improving roll-off . . . 25
- 3.7 Filter transfer functions . . . 29
  - 3.7.1 Transfer function poles and zeros . . . 33
  - 3.7.2 Transfer function for an RC low-pass filter . . . 36
  - 3.7.3 Transfer function for an LR low-pass filter . . . 37
  - 3.7.4 Transfer function for an RC high-pass filter . . . 38
  - 3.7.5 Transfer function for an LR high-pass filter . . . 39
  - 3.7.6 Transfer function for an RLC series band-pass filter . . . 40
  - 3.7.7 Transfer function for an RLC shunt band-pass filter . . . 42
  - 3.7.8 Transfer function for an RLC series band-stop filter . . . 45
  - 3.7.9 Transfer function for an RLC shunt band-stop filter . . . 47
- 3.8 Filter poles and zeroes . . . 50
  - 3.8.1 Two-pole low-pass filter plots . . . 53
  - 3.8.2 Two-pole high-pass filter plots . . . 54
  - 3.8.3 Two-pole band-pass filter plots . . . 55
  - 3.8.4 Two-pole band-stop (notch) filter plots . . . 56
  - 3.8.5 Two-pole all-pass filter plots . . . 57
  - 3.8.6 Butterworth filter pole locations . . . 58
  - 3.8.7 Bessel filter pole locations . . . 59

3.8.8 Chebyshev filter pole locations . . . 60
3.8.9 Elliptic filter pole/zero locations . . . 62
3.9 Passive filter design . . . 64
3.9.1 Low-pass passive filter design example . . . 67
3.9.2 High-pass passive filter design example . . . 69
3.10 Active filters . . . 71
3.11 Active filter design . . . 75
3.11.1 Low-pass active filter design example . . . 77
3.11.2 High-pass active filter design example . . . 79
3.12 Active filter transfer functions . . . 81
3.12.1 Sallen-Key low-pass filter transfer function derivation . . . 82
3.12.2 Sallen-Key high-pass filter transfer function derivation . . . 87

4 Historical References **89**
4.1 Wave screens . . . 90

5 Derivations and Technical References **93**
5.1 Decibels . . . 94
5.2 IEC standard component values . . . 104
5.3 The meaning of the s variable . . . 106
5.4 Impedance expressed using the s variable . . . 110
5.5 Butterworth passive filter component values . . . 115
5.6 Chebyshev passive filter component values . . . 117
5.7 Operational amplifier imperfections . . . 120
5.7.1 Output voltage limits . . . 121
5.7.2 Input voltage limits and latch-up . . . 122
5.7.3 Input bias current . . . 124
5.7.4 Input offset voltage . . . 128
5.7.5 Frequency compensation . . . 129
5.7.6 Gain-bandwidth product . . . 129
5.7.7 Slew rate . . . 130
5.7.8 Common-mode rejection . . . 131

6 Programming References **133**
6.1 Programming in C++ . . . 134
6.2 Programming in Python . . . 138
6.3 Butterworth passive filter designer using C++ . . . 143

7 Questions **147**
7.1 Conceptual reasoning . . . 151
7.1.1 Reading outline and reflections . . . 152
7.1.2 Foundational concepts . . . 153
7.1.3 Single-pole filters . . . 155
7.1.4 Filter component value scaling . . . 156
7.1.5 Limiting-case analysis of Sallen-Key filter networks . . . 157
7.2 Quantitative reasoning . . . 158

7.2.1 Miscellaneous physical constants . . . 159
7.2.2 Introduction to spreadsheets . . . 160
7.2.3 RC filter component selection program . . . 163
7.2.4 Sallen-Key active low-pass filter design . . . 165
7.2.5 Sallen-Key active high-pass filter design . . . 165
7.3 Diagnostic reasoning . . . 166
7.3.1 Effects of faults in a multi-order filter . . . 167

A Problem-Solving Strategies **169**

B Instructional philosophy **171**
B.1 First principles of learning . . . 172
B.2 Proven strategies for instructors . . . 173
B.3 Proven strategies for students . . . 175
B.4 Design of these learning modules . . . 176

C Tools used **179**

D Creative Commons License **183**

E References **191**

F Version history **193**

Index **194**

Chapter 1

Introduction

1.1 Recommendations for students

Filters are extremely important networks in a variety of electrical and electronic circuits, used to pass and block certain frequencies. Simple filter networks may be made from a single resistor and a single capacitor, or from a single resistor and a single inductor, but higher-performing filters will contain more than just two components. This module focuses on more advanced passive filter network designs as well as active filters (incorporating electronic amplification).

Important concepts related to filters include **capacitive reactance**, **inductive reactance**, effects of **opens** versus **shorts**, **voltage divider** networks, **cutoff frequency**, **parasitic** properties, **resonance**, **Bode plots**, **roll-off**, **decibels**, **fundamental frequency**, **harmonic frequency**, **exponential functions**, **real** versus **imaginary** versus **complex** numbers, **gain**, **impedance**, **quadratic formula**, **source** versus **load**, **balanced** versus **unbalanced** signals, , and .

Due to the heavy application of math (in the form of *transfer functions*), the reader is urged to apply the mathematical principles as soon as possible. The examples contained in the tutorial showing how transfer functions, poles, and zeroes are derived from basic filter networks serves this purpose well, as they provide opportunities to see if you can derive the functions yourself and then check the correctness of your work against the final results shown. Do not simply read a mathematical example and assume you understand it just because nothing in the presentation seemed confusing. *Until you can perform the analysis yourself without assistance, you haven't mastered it!*

Here are some good questions to ask of yourself while studying this subject:

- How might an experiment be designed and conducted to gather enough data to sketch a Bode plot for a filter network having unknown characteristics? What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?
- How might an experiment be designed and conducted to measure the input impedance of a filter network? What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?

- In what form do inductors store energy?
- In what form do capacitors store energy?
- How does the problem-solving technique of "limiting cases" help us understand filter networks?
- What are some practical applications of filter networks?
- How is "cutoff frequency" defined for a filter network?
- Why are capacitors usually favored over inductors for creating filter networks?
- What is "resonance" and how does it manifest in both electrical and mechanical systems?
- How does a Bode plot differ from an oscillograph?
- How does a pole-zero plot differ from a Bode plot and also an oscillograph?
- What does "roll-off" mean for a filter network?
- How may the roll-off rate of a filter be made more perfect?
- What are some compromises that often accompany increased roll-off for a filter?
- What does the phrase "brick wall" mean for the response of an ideal filter network?
- How do "L", "T", and "π" filter network topologies differ from one another?
- What is a "zero" within an electrical network, and what practical meaning does this term have?
- What is a "pole" within an electrical network, and what practical meaning does this term have?
- What does it mean if a complex frequency has a real value of zero?
- What does it mean if a complex frequency has a real value that is positive?
- What does it mean if a complex frequency has a real value that is negative?
- How does one add algebraic fractions with dissimilar denominators?
- What differentiates complex frequency from simple frequency in an AC circuit?
- What does it mean to *scale* a value?
- What does the "order" value of a filter network represent?
- What advantages do active filters have over passive filters?
- What advantages do passive filters have over active filters?
- How might an experiment be designed and conducted to gather enough data to sketch a Bode plot for a filter network having unknown characteristics?

1.2 Challenging concepts related to complex filter networks

The following list cites concepts related to this module's topic that are easily misunderstood, along with suggestions for properly understanding them:

- **Reasoning from trusted principles** – many students enter college-level study of electronics with an educational background stressing rote memorization at the expense of logical reasoning from trusted principles, and as such tend to find circuit analysis daunting where there is no single procedure or single formula always yielding the correct answer(s). These students also try to rote-memorize circuit configurations rather than use logic to determine what each of those configurations does. In the case of simple filter networks the tendency is to try to memorize the component positions and associate them with labels such as "high-pass", "low-pass", etc. A much better approach is to view each new filter network from the perspective of a voltage divider when subjected to signals of different frequency, using general principles of "opens" and "shorts" to conclude the effects on the output signal as frequency values go to extremes (i.e. DC versus super-high frequency). In other words, apply the problem-solving strategy of *limiting cases* to every filter circuit so as to figure out its function rather than try to memorize it!

- **Decibels** – "decibels" are an attempt to express power ratios (i.e. power gains or attenuation factors) logarithmically rather than linearly, and as such they tend to generate confusion for students less familiar (or unfamiliar) with exponential and logarithmic functions.

- **Practical filter applications** – Bode plots show how filter circuits respond to inputs of changing frequency, but this is not how filters are typically used in real applications. Rarely does one find a filter circuit subjected to only one particular frequency at a time – usually a simultaneous mix of frequencies are seen at the input, and it is the filter's job to select a particular range of frequencies to pass through from that simultaneous mix. Understanding the superposition theorem is helpful for comprehending practical filter applications.

- **Transfer function pole** – a "pole" in any transfer function is some value of s where that function's value (i.e. the gain of the system) goes to infinity. This is where we have an output signal even with no input signal, usually the result of some energy-storing component in the system releasing its energy from a prior moment in time where we *did* have an input signal to charge it up with energy, s representing the complex frequency of that output signal.

- **Transfer function zero** – a "zero" in any transfer function is some value of s where that function's value (i.e. the gain of the system) goes to zero. This is where we get no signal out of the system even with signal coming in, s representing the complex frequency of that input signal.

- **Complex frequency** – if a complex number is used as an exponential value (i.e. e raised to the power of a complex number), the "real" portion of that complex number represents a growth/decay rate while the "imaginary" portion of that complex number represents a sinusoidal oscillation rate. Specifically $s = \sigma + j\omega$ where s is the complex number, σ is the growth/decay rate, and ω is the oscillation rate. The proper unit of measurement for both is "per second" which is equivalent to "radians per second" for the oscillation rate.

1.3 Recommendations for instructors

This section lists realistic student learning outcomes supported by the content of the module as well as suggested means of assessing (measuring) student learning. The outcomes state what learners should be able to do, and the assessments are specific challenges to prove students have learned.

- **Outcome** – Demonstrate effective technical reading and writing

  Assessment – Students present their outlines of this module's instructional chapters (e.g. Case Tutorial, Tutorial, Historical References, etc.) ideally as an entry to a larger Journal document chronicling their learning. These outlines should exhibit good-faith effort at summarizing major concepts explained in the text.

  Assessment – Students show how quantitative results were obtained by the author in the Tutorial chapter's examples.

  Assessment – Students present their "Conceptual Toolbox" Theory Journal entries listing important concepts applied in the Tutorial text(s). A "Conceptual Toolbox" serves as inspiration when solving problems, each concept suggesting a potential problem-solving action. Challenge students to collaboratively generate a Conceptual Toolbox for the day's topic during the class session and then use it when solving problems together, thus fostering the habit of reasoning from principles rather than recalling procedures!

- **Outcome** – Design a multi-order passive filter network

  Assessment – Sketch a schematic diagram and choose component values for a multi-order filter with a given characteristic and polynomial type (e.g. third-order, low-pass, Butterworth).

- **Outcome** – Independent research

  Assessment – Locate filter network datasheets (e.g. SAW-type filters) and properly interpret some of the information contained in those documents including pass characteristic(s), frequency ranges, power ratings, etc.

  Assessment – Read and summarize in your own words reliable source documents on the subject of filter design. Recommended readings include tutorials and primers published by electronic component manufacturers.

  Assessment – Download and run filter-design software (or use web-based applications) to choose components and topologies for various filter types.

Chapter 2

Case Tutorial

The idea behind a *Case Tutorial* is to explore new concepts by way of example. In this chapter you will read less presentation of theory compared to other Tutorial chapters, but by close observation and comparison of the given examples be able to discern patterns and principles much the same way as a scientific experimenter. Hopefully you will find these cases illuminating, and a good supplement to text-based tutorials.

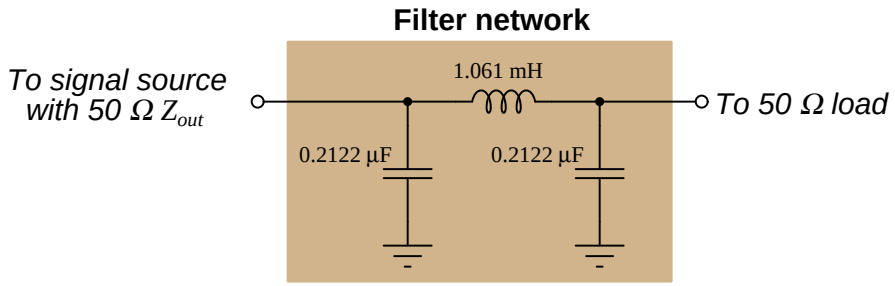
These examples also serve well as challenges following your reading of the other Tutorial(s) in this module – can you explain *why* the circuits behave as they do?

2.1 Example: 3rd-order low-pass filter design

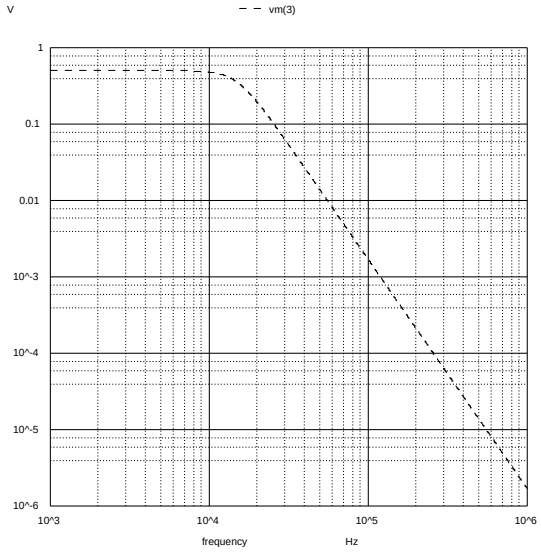
Design parameters:

- Low-pass characteristic
- Ladder topology, unbalanced
- 3rd order Butterworth response
- $f_{cutoff} = 15$ kHz
- Z_{in} and $Z_{load} = 50\ \Omega$

Schematic diagram:



SPICE analysis:

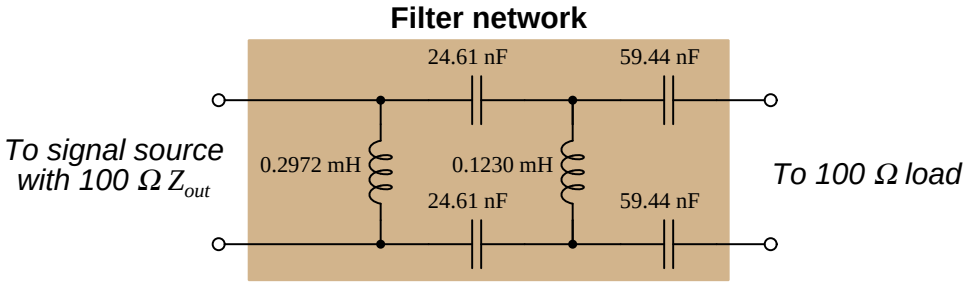


2.2 Example: 4th-order high-pass filter design

Design parameters:

- High-pass characteristic
- Ladder topology, balanced
- 4th order Butterworth response
- f_{cutoff} = 70 kHz
- Z_{in} and Z_{load} = 100 Ω

Schematic diagram:



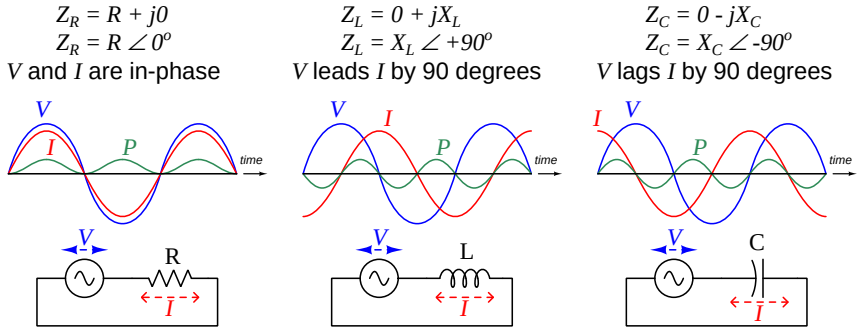
Chapter 3

Tutorial

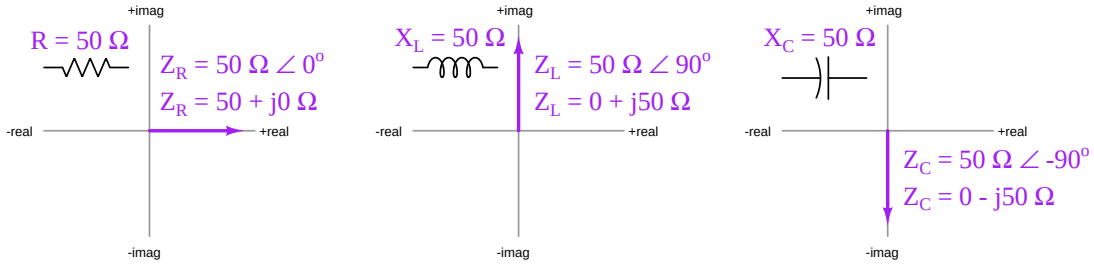
3.1 Phasor analysis review

When analyzing any AC circuit, it is not enough to simply quantify every voltage and current in terms of amplitude (e.g. how many Volts or Amperes) as we do in DC circuits. Instead, we must consider both the amplitude of each signal as well as the amount of phase shift separating them. A mathematically elegant way of accomplishing both is to use *complex numbers* which may be expressed either in rectangular form (e.g. $a + jb$) or polar form (e.g. $m\angle\theta$). When we use complex numbers to represent any AC circuit quantity, we call that value a *phasor*.

An illustrative example is how we characterize the *impedance* of passive components. Resistors function by dissipating energy in the form of heat, with voltage and current waveforms being perfectly in-phase. Inductors and capacitors, however, function by absorbing and releasing energy rather than dissipating, and as such we find voltage and current waveforms shifted by one-quarter of a cycle (i.e. 90^o) for each, voltage leading current for an inductor and voltage lagging current for a capacitor:



Graphical expressions of complex-number resistance and reactance values are called *phasor diagrams*. Three such diagrams illustrate the difference between a 50 Ω resistor versus an inductor having 50 Ω of reactance and a capacitor also having 50 Ω of reactance. Each has 50 Ohms of impedance, but each of these impedances has a different phase angle defined by the phase shift between the component's voltage and current:



The ratio of voltage to current in a DC network is resistance ($R = \frac{V}{I}$), and in an AC circuit where phase shifts exist is impedance ($Z = \frac{V}{I}$). Both are measured in unit of the Ohm (Ω). The ratio of current to voltage is the reciprocal of these quantities: for DC networks we call it *conductance* ($G = \frac{I}{V}$) and for AC it is known as *admittance* ($Y = \frac{I}{V}$). We measure both in the unit of *Siemens*[1] (S).

Some examples[2] of component impedances and admittances are shown here:

- A 570 Ω resistor at any frequency will have the following impedance and admittance values:

  $Z = 570\ \Omega \angle 0^o$ *(polar form)* $= 570 + \text{j}0\ \Omega$ *(rectangular form)*

  $Y = 0.0017544\ \text{S} \angle 0^o$ *(polar form)* $= 0.0017544 + \text{j}0\ \text{S}$ *(rectangular form)*

- A 3.5 H inductor at a frequency of 120 Hz will have the following impedance and admittance values:

  $Z = 2.639\ \text{k}\Omega \angle +90^o$ *(polar form)* $= 0 + \text{j}2.639\ \text{k}\Omega$ *(rectangular form)*

  $Y = 0.00037894\ \text{S} \angle -90^o$ *(polar form)* $= 0 - \text{j}0.00037894\ \text{S}$ *(rectangular form)*

- A 0.01 μF capacitor at a frequency of 3 kHz will have the following impedance and admittance values:

  $Z = 5.305\ \text{k}\Omega \angle -90^o$ *(polar form)* $= 0 - \text{j}5.305\ \text{k}\Omega$ *(rectangular form)*

  $Y = 0.00018850\ \text{S} \angle 90^o$ *(polar form)* $= 0 + \text{j}0.00018850\ \text{S}$ *(rectangular form)*

The utility of phasor representation in AC circuits is that with all signal and component values expressed in phasor form we find most of the foundational principles learned for DC circuit analysis still apply in AC circuits. Quantities that add in series DC networks (e.g. voltage V, resistance R) add as phasor quantities in AC networks (e.g. voltage V, impedance Z); additive quantities in parallel DC networks (e.g. current I, conductance G) add as phasor quantities in AC networks (e.g. current I, admittance Y). With phasor quantities, Ohm's Law, Kirchhoff's Voltage Law, and Kirchhoff's Current Law still hold true in AC networks just as they do for DC.

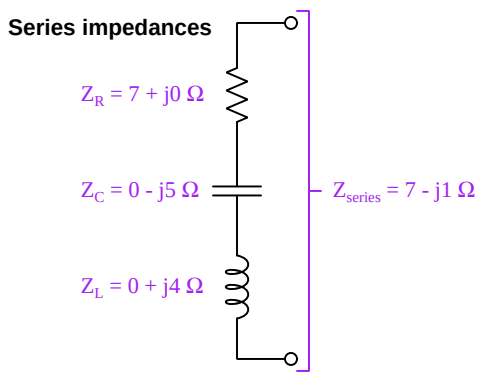
[1] Prior to the adoption of German engineer Werner von Siemens' surname as the unit of measurement for conductance and admittance, the unit of the *Mho* served quite well. This, of course, was a sort of pun on the spelling of *Ohm*, since "mho" is "ohm" spelled backwards, intended to represent the fact that the reciprocal of any Ohm value yields a value in Mhos.

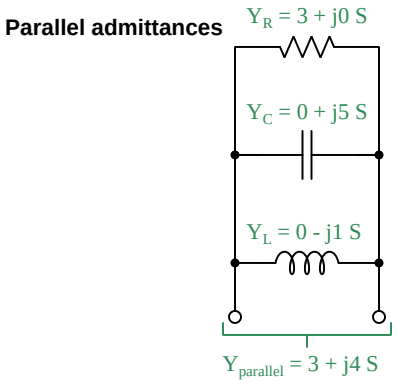
[2] Try calculating these impedance and admittance values from the given component values, to check your understanding. This is a good learning strategy to apply when reading any mathematical text: work through the presented examples on your own to see if you achieve the same results! Please note that when you apply either the $X_L = 2\pi f L$ formula or the $X_C = \frac{1}{2\pi f C}$ formula using your calculator to compute reactance, the result will *only* be a reactance value and not a (complex) impedance value. In order to attach the desired phase angle to your computed reactance value, you will have to perform the additional step of multiplying that reactance by a *unit phasor* which is nothing more than the quantity of 1 with the correct phase angle. For example, a capacitive reactance of 5.305 kΩ would be multiplied by $1 \angle -90^o$ to yield a capacitive *impedance* of 5.305 k$\Omega \angle -90^o$.

It is equally valid to express any phasor quantity in either polar or rectangular form. However, unless we have access to an electronic calculator capable of performing complex-number arithmetic, we find certain arithmetic operations much easier to perform with one notation more than the other. Specifically, addition and subtraction are simplest when phasors are in rectangular form, while multiplication and division are simplest when phasors are in polar form.

Addition of rectangular-form complex numbers consists simply of adding their real components together to find the real component of the sum, and doing the same with the imaginary components. Expressing this algebraically, $(a + jb) + (x + jy) = (a + x) + j(b + y)$. Subtraction follows much the same pattern: $(a + jb) - (x + jy) = (a - x) + j(b - y)$.

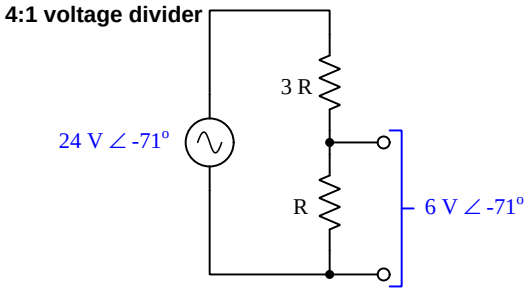
Here are some practical examples of rectangular-form phasor arithmetic where the calculations are simple enough to perform without a calculator:

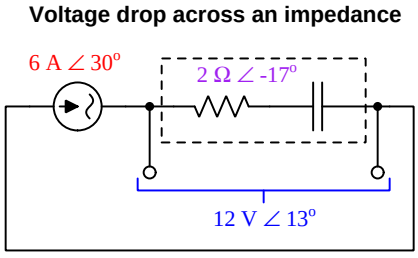




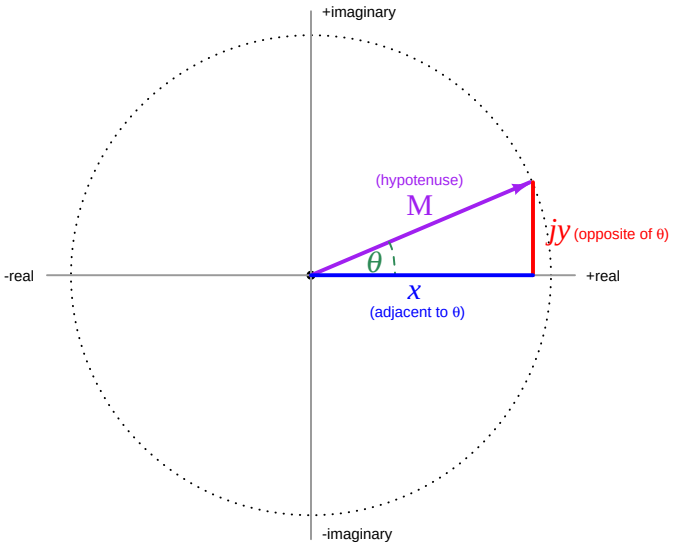
Multiplication of polar-form complex numbers consists simply of multiplying their magnitudes together to find the magnitude of the product, and adding the angles to find the angle of the product. Expressing this algebraically, $(a\angle b) \times (x\angle y) = (a \times x)\angle(b + y)$. Division follows a similar pattern: $(a\angle b) \div (x\angle y) = (a \div x)\angle(b - y)$.

Here are some practical examples of polar-form phasor arithmetic where the calculations are simple enough to perform without a calculator:





When analyzing AC circuits without the use of a complex-number calculator, we invariably must convert between rectangular and polar forms in order to prepare the phasor values for addition/subtraction or multiplication/division, respectively. Both conversions are most easily understood in terms of a right triangle, seeing the rectangular form's real and imaginary components as the adjacent and opposite sides, and the polar form's magnitude and angle as the hypotenuse:



Converting rectangular $(x + jy)$ into polar $(A\angle\theta)$:

$$A = \sqrt{x^2 + y^2} \qquad \theta = \arctan \frac{y}{x}$$

Converting polar $(A\angle\theta)$ into rectangular $(x + jy)$:

$$x = A \cos \theta \qquad y = A \sin \theta$$

Some cautionary notes are in order here. First, I highly recommend storing all computed values in your calculator's memory rather than re-entering them manually, because you will find even slight rounding errors tend to become exaggerated with trigonometric functions. Second, when computing the phase angle (θ) from real and imaginary quantities (x and jy) be careful to verify the angle against your qualitative expectations. For example, $5 + j5 = 7.071\angle 45^o$ and $-5 - j5 = 7.071\angle 225^o$, but you'll find arctan $\frac{-5}{-5}$ yields the same result (45^o) as arctan $\frac{5}{5}$ because $\frac{-5}{-5} = \frac{5}{5}$. To put it simply, the arc-tangent function does not "know" whether the phasor exists in the first or in the third quadrant of the complex plane.

Here are some rectangular and polar equivalents, useful for practice as you master these concepts:

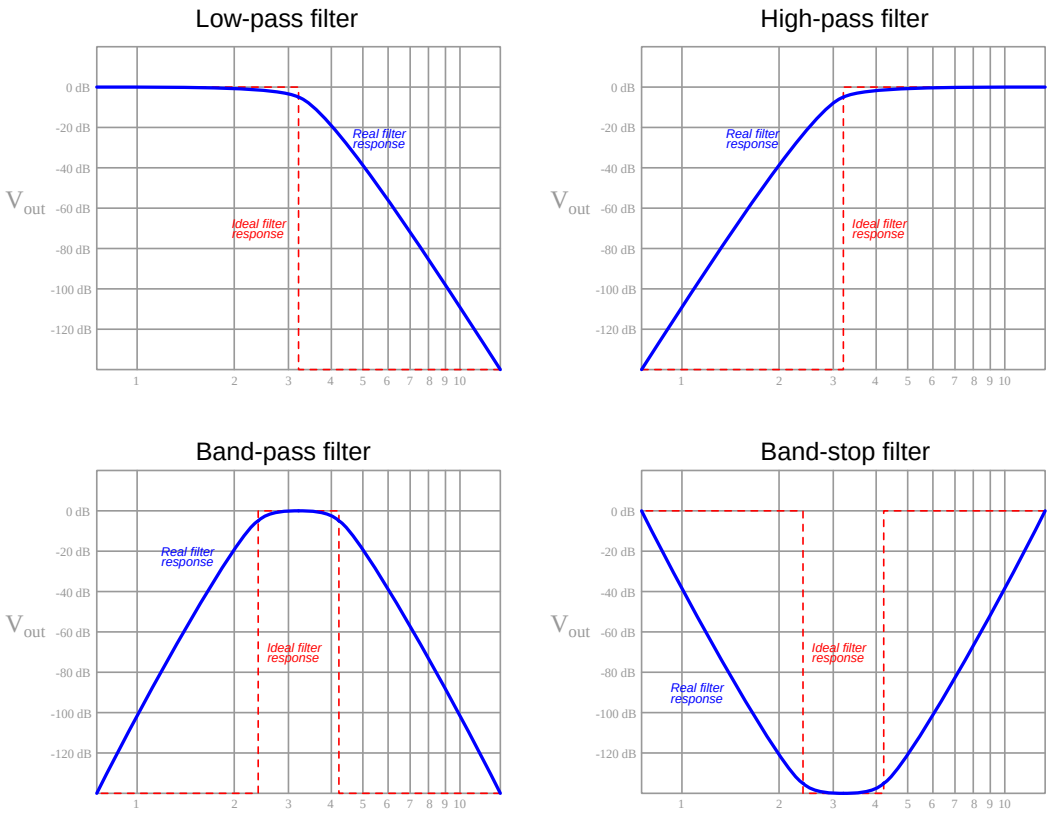
$$20 - j11 = 22.83\angle -28.81^o \qquad 11.49 + j9.642 = 15\angle 40^o$$

$$-10 + j2 = 10.20\angle 168.7^o \qquad -11.82 - j2.084 = 12\angle -170^o$$

3.2 Basic filter networks

A *filter* network is a collection of electrical (and sometimes electronic) components designed to separate different ranges of frequencies from each other in a complex signal consisting of multiple frequencies. An every-day example of signal filtering is in the *bass* and *treble* controls of an audio amplifier system, giving the listener control over the intensity of low-frequency (bass) and high-frequency (treble) tones in the amplified music. Specifically, the bass adjustment regulates the degree to which low-frequency signals get passed along to the loudspeaker, while the treble adjustment regulates the degree to which high-frequency signals get passed along to the loudspeaker.

Filter circuits may be classified as one of four types, based on the range of frequencies intended to pass or block. In the following Bode plots we see the ideal "brick wall[3]" response of each filter type (dashed red) overlaid on more realistic filter response curves (bold blue):

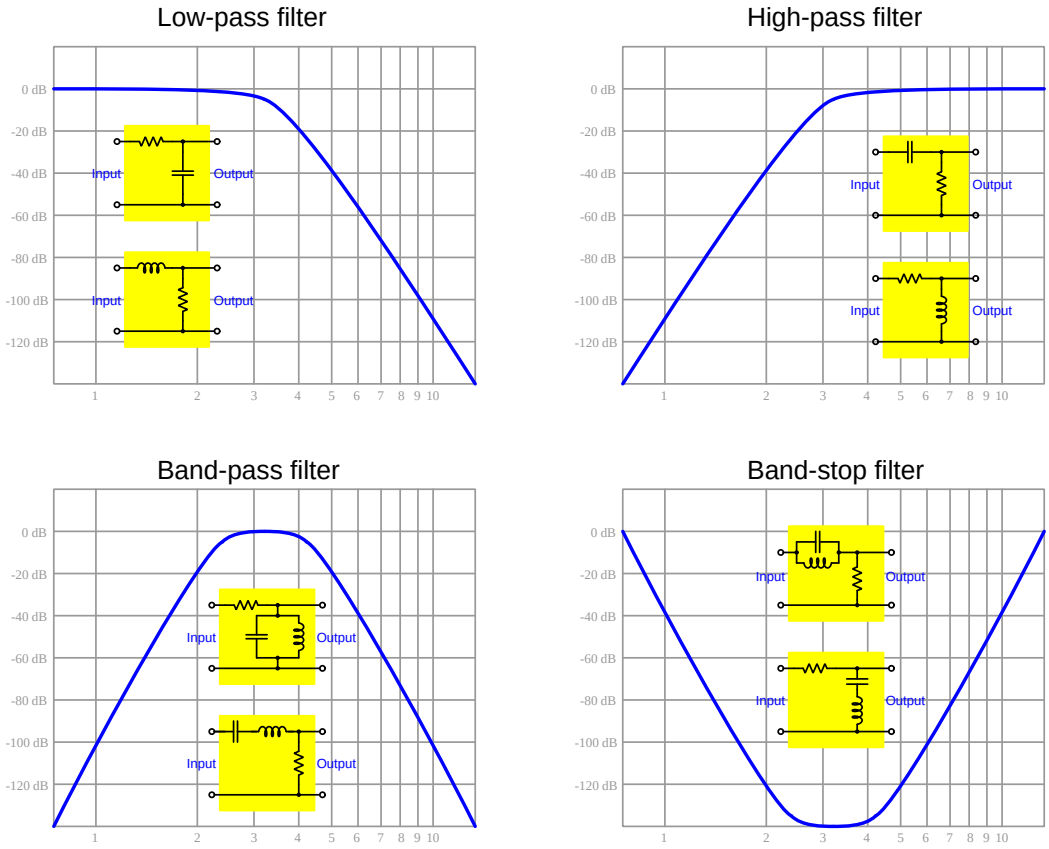


Cutoff frequency is defined as that frequency value at which the signal is reduced to $\frac{\sqrt{2}}{2}$ of the input signal's strength, based on the output signal magnitude of a simple RC or LR filter network when $X = R$. Any frequency resulting in an output signal stronger than this is within the filter's

[3]This rather colorful description of ideal filter response evokes the image of an impenetrable wall completely stopping the passage of certain signal frequencies.

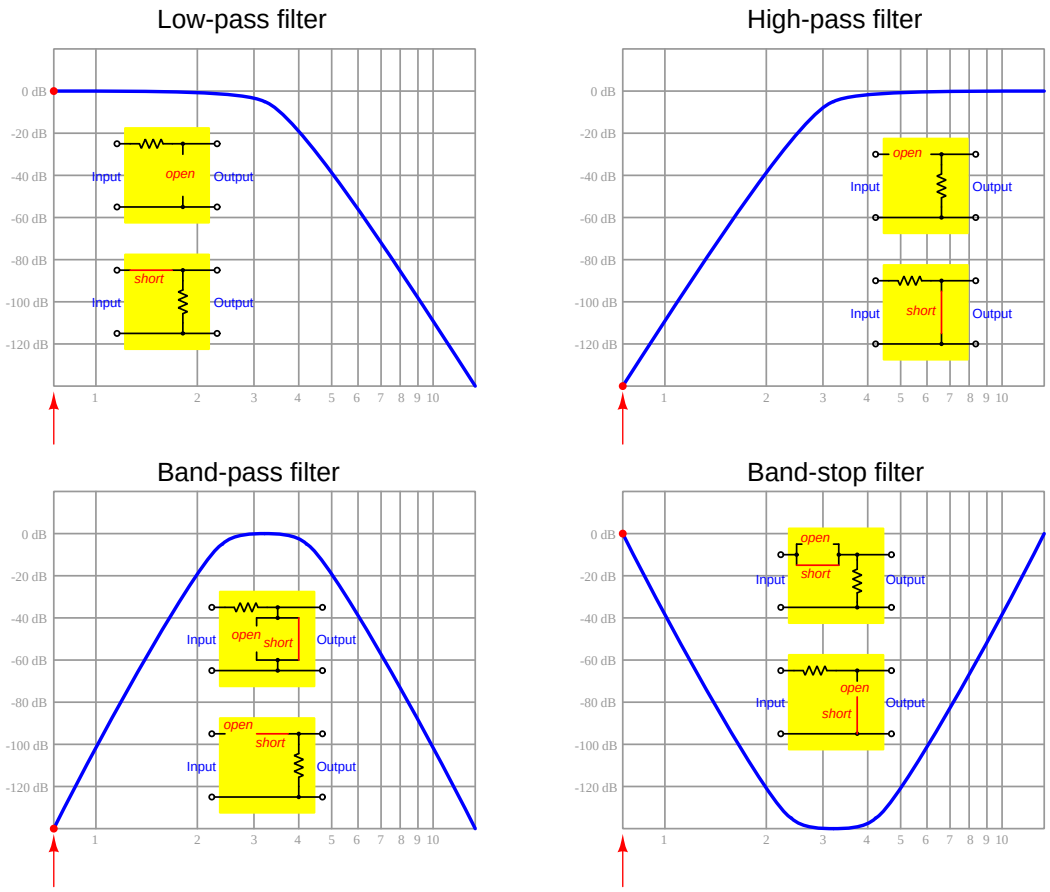
passband, with all other frequencies lying in the filter's *stopband*. *Roll-off* is the rate at which a filter's cutoff changes with frequency, essentially the "steepness" of its Bode plot. Filters with high roll-off are more selective, but typically that increase in selectivity is realized only by unequal passage of signals at slightly different frequencies within the filter's "passband" (i.e. a passband response that is not flat). An ideal filter has a perfectly flat passband and infinite roll-off.

Elementary realizations of each filter type using passive components (resistors, capacitors, and/or inductors) appear below as networks sketched amidst the Bode plots:



Filter networks such as these behave as frequency-dependent voltage dividers, the voltage-division ratio depending on the impedance of its reactive components, with inductive reactance being directly proportional to frequency ($X_L = 2\pi fL$) and capacitive reactance being inversely proportional to frequency ($X_C = \frac{1}{2\pi fC}$).

A good way to approach the analysis of any passive filter network is to imagine how it will change when subjected to extremely high and extremely low signal frequencies, an application of the *limiting cases* method of problem-solving. For example, in the following illustration we see the four basic types of filters subjected to an input signal frequency of zero Hertz(i.e. DC), causing each of the capacitors to act as an "open" and each of the inductors to act as a "short":



The position of the red dot along the Bode plot shows how much output voltage there would be when the input signal has a frequency of zero. Performing this same open/short "limiting cases" analysis at some extremely high frequency where all capacitors act as shorts and all inductors as opens allows us to conceptually grasp each filter's behavior at the other end of the frequency spectrum. This is left as an exercise to the reader, and it is a profitable one when you are new to the topic of filter networks!

The "cutoff" frequency (f_c) of simple RC and LR filter networks is conventionally defined as that frequency at which the reactive component's reactance (either X_C or X_L) equals the resistor's resistance. If we set resistance (R) equal to reactance ($2\pi fL$ for an inductor, and $\frac{1}{2\pi fC}$ for a capacitor) and then solve each equation for f_c, we will arrive at the formulae used to predict cutoff frequency for simple RC and LR filter networks:

$$R = 2\pi f_c L \qquad R = \frac{1}{2\pi f_c C}$$

$$f_c = \frac{R}{2\pi L} \qquad f_c = \frac{1}{2\pi RC}$$

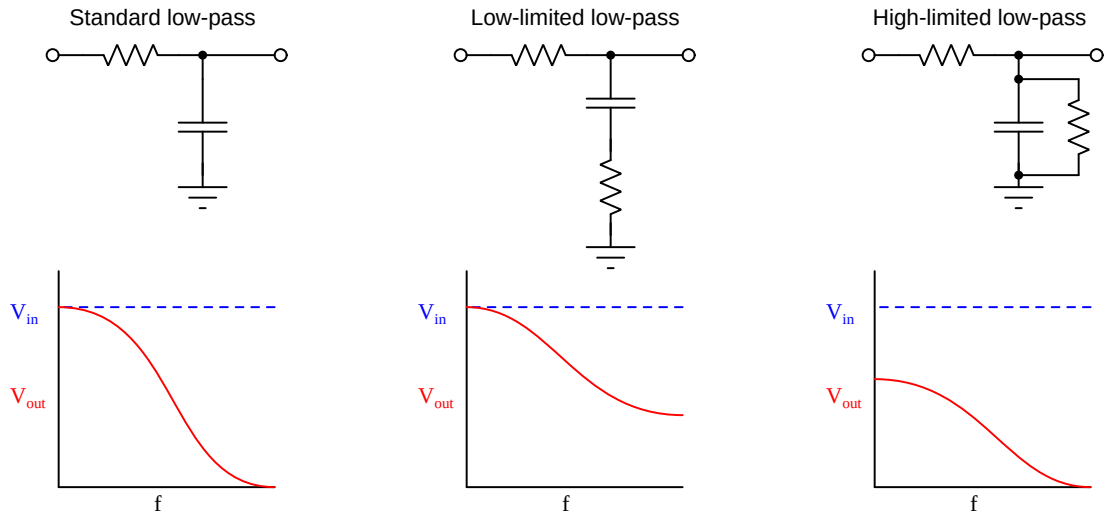
For simple LC band-pass and band-stop filter networks, the "center" frequency representing the peak of the pass-band and the valley of the stop-band is calculated simply by the resonant LC frequency formula. This is the frequency value at which a series LC network has zero total impedance (together acting as a short) and a parallel LC network has infinite impedance (together acting as an open):

$$f = \frac{1}{2\pi\sqrt{LC}}$$

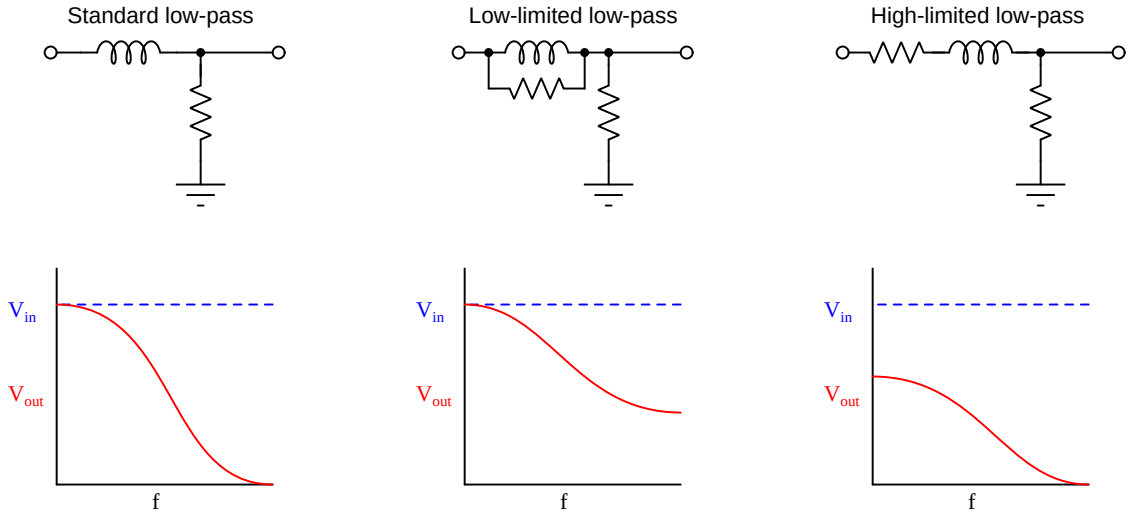
All of these formulae, of course, assume perfect components with no parasitic properties, a source having negligible Thévenin impedance, and no load connected to the filter network's output terminals.

3.3 Output-limited filter networks

Installing an extra resistor into simple RC and LR low-pass and high-pass networks results in the output signal being limited in various ways. First, we have RC filter networks with limited outputs:



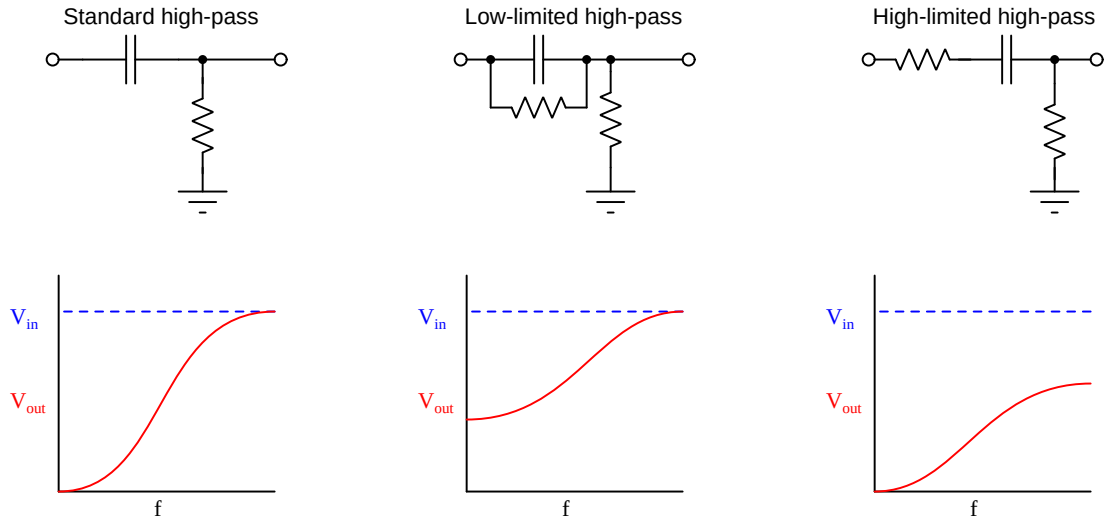
Next, LR filter networks with limited outputs:



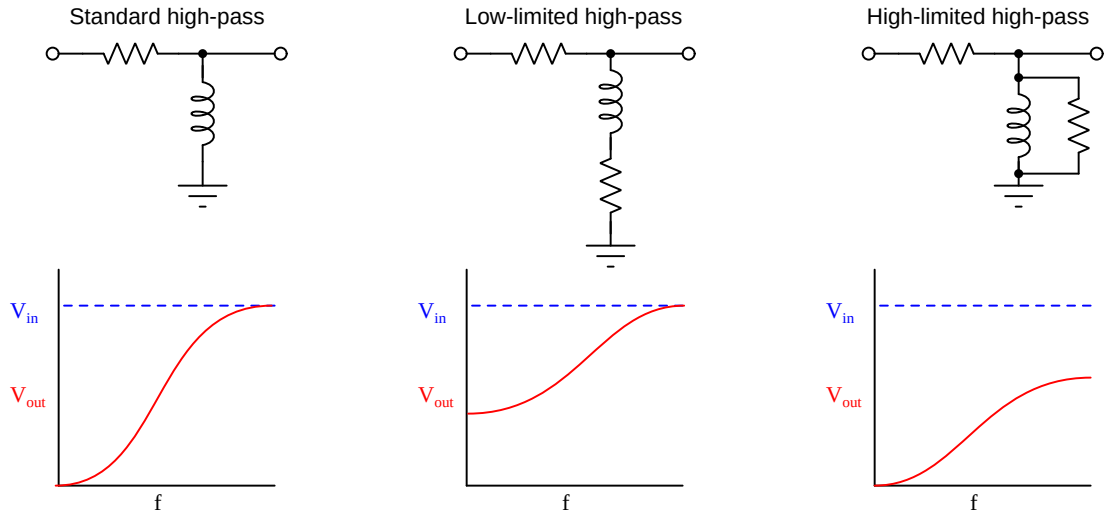
Limiting cases once again enables us to see how and why each of these filter network designs performs as it does. Simply imagine each of these networks at extremely low and extremely high frequencies, determine if the reactive component in each acts more like a short or like an open,

and then based on those determinations figure out how much signal voltage will be output by the network.

High-pass RC filter networks may also be constructed in similar manner by strategically adding a second resistor:

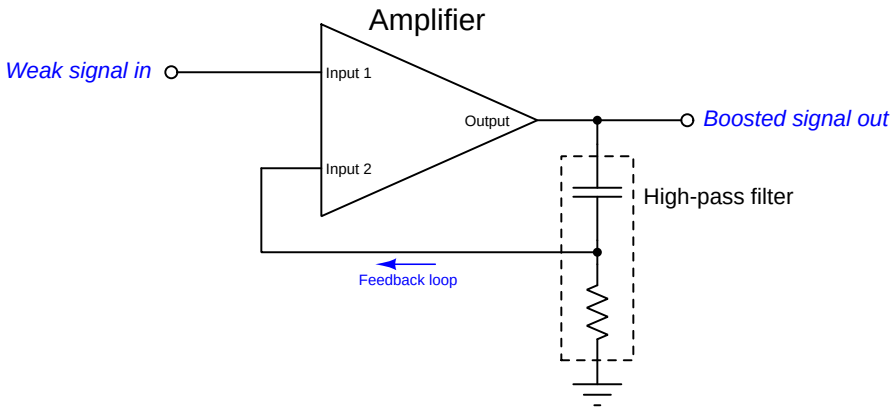


And, of course, inductor-based versions of these same output-limited high-pass circuits exist as well:

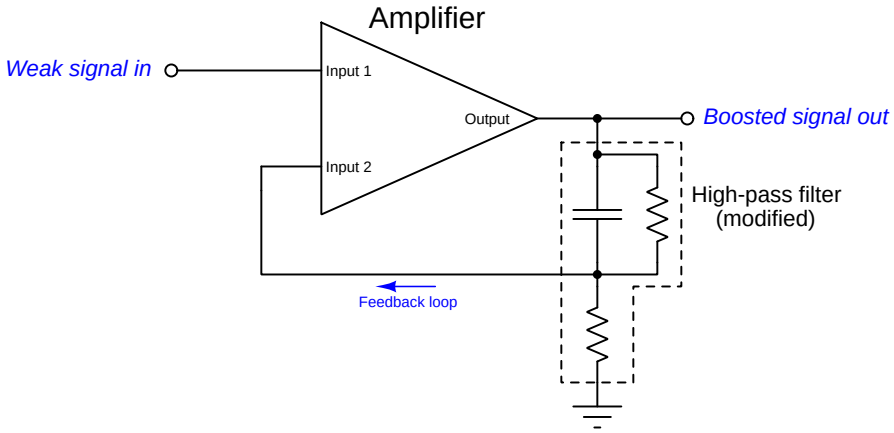


Output-limited filter networks are frequently used in electronic systems employing *negative feedback*, where a signal-boosting system (called an *amplifier*) re-directs part of the boosted output signal back around to the amplifier's input as a means of improving certain performance criteria such as stability. Often a filter network is placed within this feedback "loop" to favor some frequencies over others, but if some of those un-favored frequencies are *completely* cut off by the filter the system may become unstable for those excluded frequencies.

Here is a visual example of an amplifier system using negative feedback with standard high-pass filtering in the feedback loop:

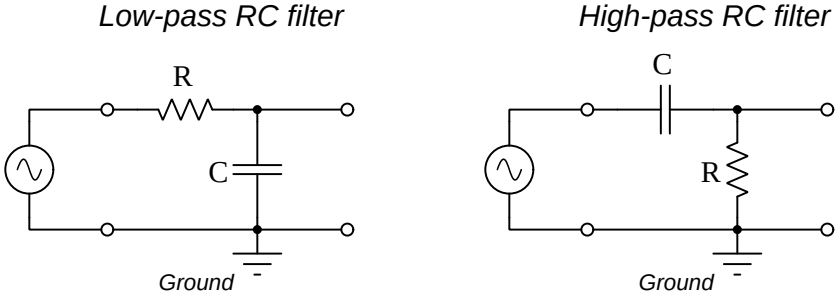


Unfortunately, this feedback system will cause trouble for the amplifier because it completely blocks all DC (zero-frequency) signals from being fed back to Input 2. This will mean that the amplifier cannot naturally stabilize itself for DC signals using negative feedback as it can for AC signals. The solution is to modify the filter network to have a low-limited characteristic so that at least some of the DC signals are able to reach Input 2:

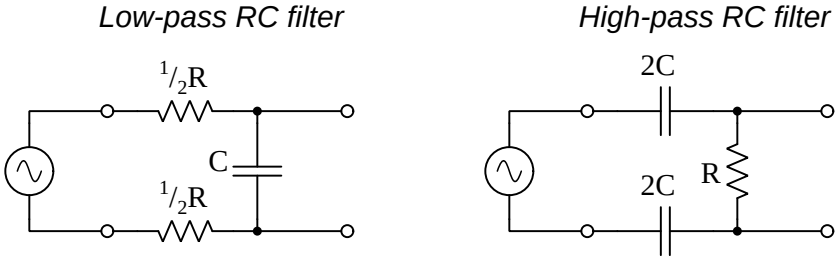


3.4 Balanced versus unbalanced filter networks

So far all of the basic filter networks shown have electrically-common terminals shared between input and output, this pair of terminals typically connected to *ground*. Voltage signals referenced to a common point in a circuit called "ground" are known as *ground-referenced*, *single-ended*, or *unbalanced* voltage signals:



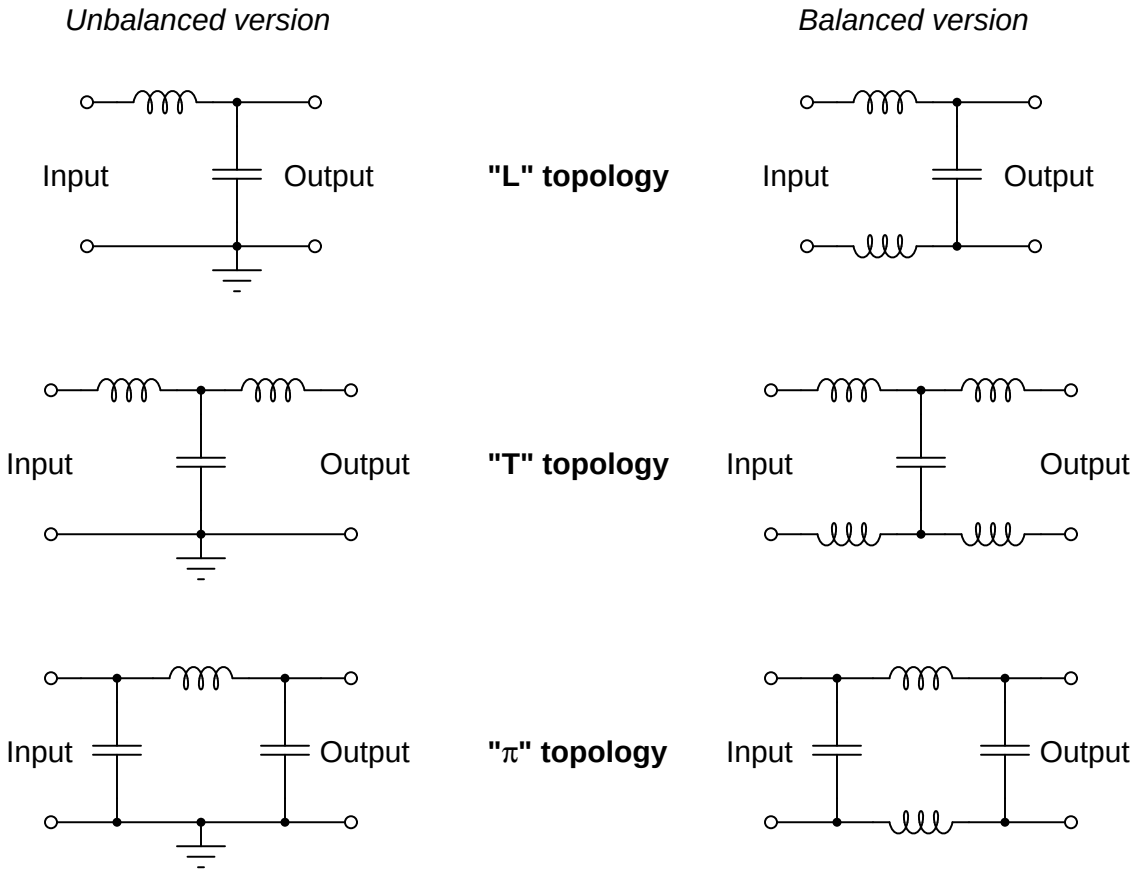
However, not all voltage signals are referenced to ground. Some are measured between two non-grounded points, and are called *differential* or *balanced* voltage signals. Filter networks designed to be used in such applications must impose the same amount of impedance in series with each line, and so the series-connected component must be replaced by two equal-valued components as shown here:



In order to maintain the same cutoff frequency as the unbalanced version, each balanced filter network must have its series components sized such that two in series produced the same amount of series impedance as one in the unbalanced filter network. For the RC low-pass filter this means having two resistors of half the value (each) as the single resistor in the unbalanced filter, because two identical resistors connected in series with each other add to make double their resistance in total. For the RC high-pass filter this means having two capacitors twice the value (each) as the single capacitor in the unbalanced filter, because two identical capacitors connected in series with each other diminish to make half their capacitance in total.

3.5 Filter network topology

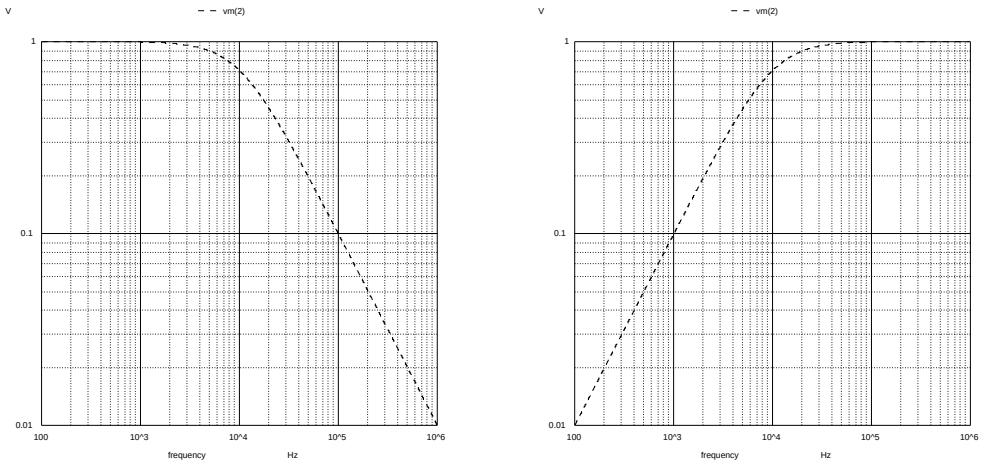
Generally speaking, the manner in which a particular filter network's components interconnect is referred to as its *topology* – literally, the *shape* of the filter network as it appears on a schematic diagram. Balanced and unbalanced variants of otherwise identically-performing filter networks is one example of a topological difference, but there are actually a wide range of filter topologies for any given filtering characteristic. Consider the following low-pass LC filter variations, shown in both unbalanced (left) and balanced (right) forms:



3.6 Improving roll-off

Most filter networks exhibit a relatively flat "passband" and a sloped "stopband" where the signal progressively attenuates at frequencies further away from the "cutoff" frequency. The rate at which the output signal's strength "rolls off" toward zero is called the *roll-off* for any given filter network. For simple filter networks consisting of a single resistor and a single reactive component (capacitor or inductor), this rate of roll-off is −20 dB per decade, which means for every ten-fold change in frequency the output voltage becomes ten times weaker (i.e. −20 dB)[4].

We may see this −20 dB/decade roll-off in the following Bode plots created using SPICE[5], showing a low-pass (left) and a high-pass (right) filter network each with a 10 kHz cutoff frequency.



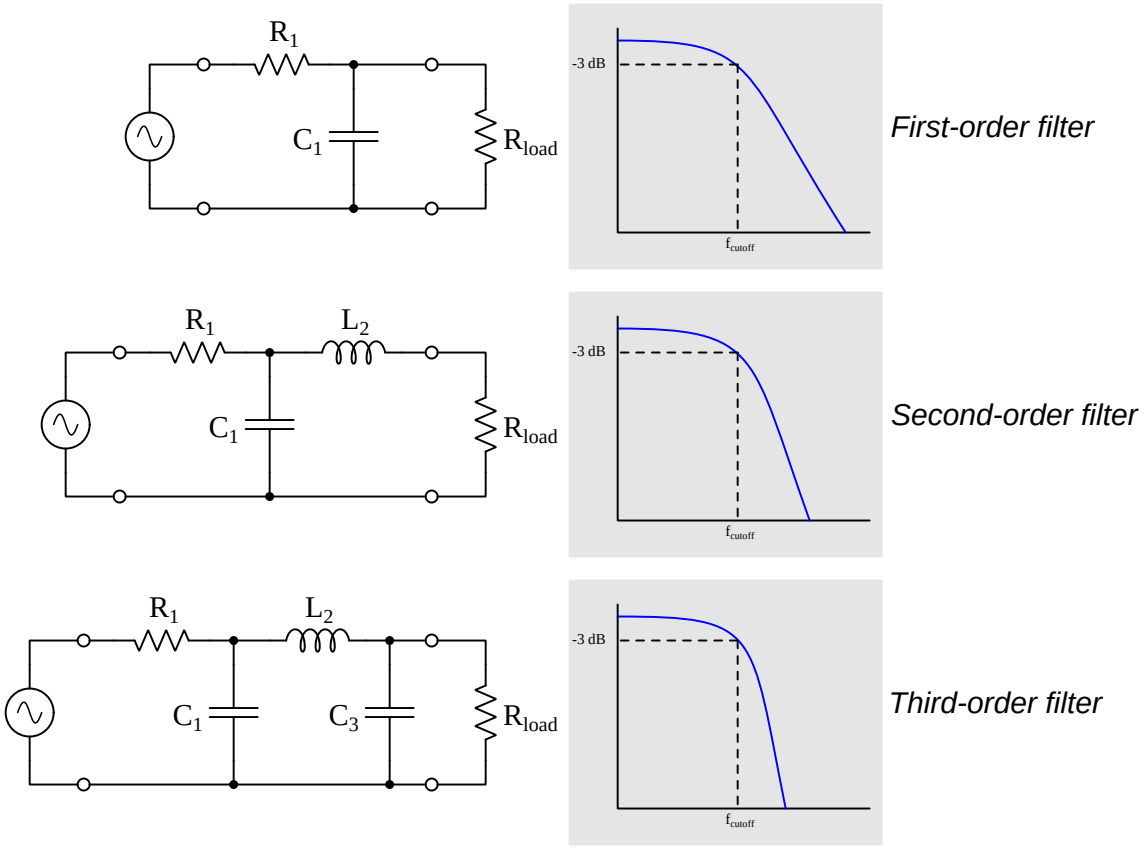
At the 10 kHz point we see the output voltage of each filter diminished from its maximum of 1 Volt to approximately 0.7 Volts ($\frac{\sqrt{2}}{2}$ Volts to be exact) which is the conventional definition of "cutoff" for a filter network. Each filter's roll-off profile appears as a straight line thanks to both the vertical and horizontal axes of the Bode plot being scaled logarithmically, commonly referred to as a "log-log" plot style. These filter simulations have been configured to make it easy to see the −20 dB per decade rolloff slope, the output voltage clearly diminishing by a factor of $\frac{1}{10}$ for every 10-fold change in frequency.

Obviously, the sloped "roll-off" response of these filter networks falls short of the ideal "brick-wall" filter response where *no* signals with frequencies in the stop-band make it through the network. This is simply impossible using passive components such as resistors, capacitors, and/or inductors. However, it is possible to *approach* ideal behavior by building more complex networks having more reactive components, the values of those additional components carefully chosen to yield the best filtering characteristics for our application.

[4] A ten-fold attenuation of *power* is −10 dB. A ten-fold attenuation of *voltage*, however, is −20 dB.

[5] In each case the filter network consisted of a 635 nF capacitor and 25 Ohms of resistance, for a cutoff frequency of 10.0255 kHz. To generate a plot with both horizontal and vertical axes plotted logarithmically, the NGSPICE command `plot loglog vm(2)` was used to plot voltage at output node number 2.

For example, compare the following low-pass filter networks, each successive network having one more reactive component than the previous:

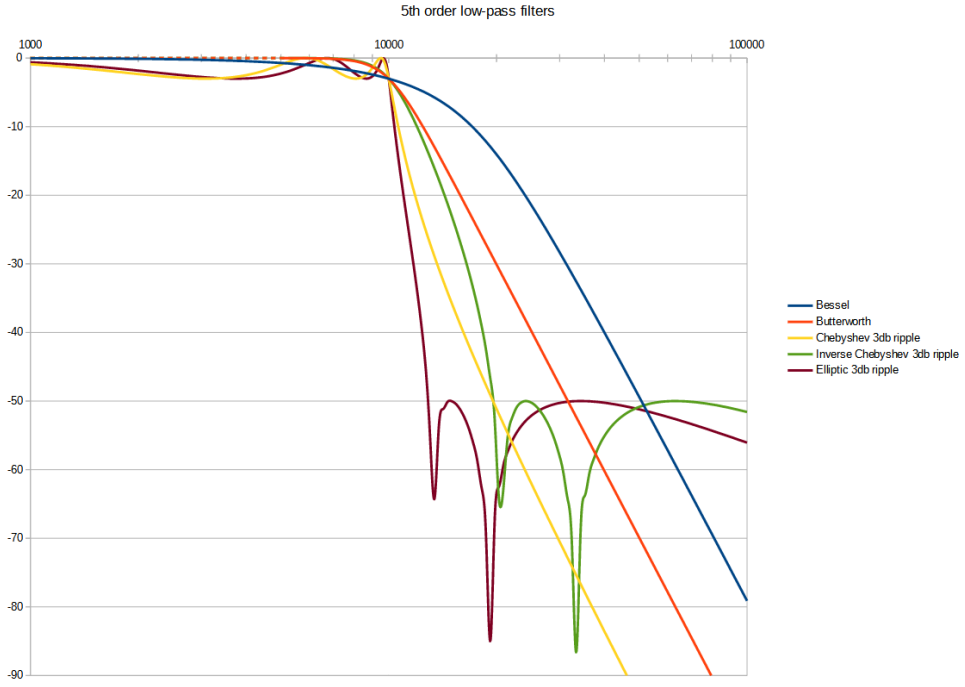


The subscript numbers on each of these components denote which order of filtration it contributes to. Here, the term "order" refers to the mathematical order of polynomial function[6] describing the behavior of each filter network. Just as polynomial algebraic functions of higher order exhibit sharper curvatures when graphed, filters with transfer functions of higher order may achieve steeper roll-offs. A simpler explanation is that the "order" of a filter network is usually[7] just the number of reactive components within it. Note how the third-order filter shown above contains two capacitors and one inductor $(2 + 1 = 3)$, and how the second-order filter contains one capacitor and one inductor $(1 + 1 = 2)$.

[6] A filter whose transfer function only contains an s term is first-order, while a filter with a transfer function containing an s^2 term is second-order, and so on.

[7] This is not always true. For example, if you need a capacitor with a non-standard value and need to combine two or more of them in some series/parallel arrangement to achieve that special capacitance value, that sub-network of capacitors would still just count as one capacitance from the perspective of filter order. Another example is in filter designs using series and/or parallel LC sub-networks in which an inductor and capacitor together contribute to just one of the filter's orders. In general, though, the order number and the number of reactive components are directly related: building a higher-order filter usually requires more capacitors/inductors to be added.

Beyond increasing the number of reactive components in a filter network to achieve steeper roll-off, there is also a broad range of options for component values within any given circuit. Some combinations of component values, for example, yield a "flat" passband response and modest roll-off rate. Other combinations may be chosen, though, that achieve a steeper roll-off rate at the expense of a unstable passband or stop-band. The mathematics involved with filter design are quite complex, but over many decades several "recipes" for component selection have been standardized according to their mathematical forms. Five of the most popular include the *Bessel*, *Butterworth*, *Inverse Chebyshev*, *Chebyshev*, and *Elliptic*, listed in order of least-aggressive to most-aggressive roll-off rate for a given order. The following graph compares these responses, each one showing the Bode plot for a 5th-order low-pass filter having a cutoff frequency of 10 kHz:



As is evident by this comparative plot, the Bessel filter exhibits a gradual attenuation from the very start, the output signal gradually drooping in strength to -3 dB at 10 kHz and becoming progressively weaker beyond that. The Butterworth filter maintains full output signal strength until somewhere around 8 kHz, rolling off more rapidly than the Bessel after that. The two types of Chebyshev filters achieve even steeper roll-off rates, but only by compromising smooth response either in the passband (Chebyshev) or in the stop-band (Inverse Chebyshev), which we commonly refer to as *ripple*. The Elliptic filter achieves the most aggressive roll-off rate of them all, but "ripples" significantly in both the passband and the stopband.

It is worth noting that the substantial differences in frequency response is often a matter of component values alone. For example, three of the five filter types shown in the previous comparative plot (Bessel, Butterworth, and Chebyshev) have identical schematic diagram forms – three parallel capacitors and two series inductors each – differing only in their capacitor and inductor values. This is to say, one cannot easily identify a filter's response type by mere examination of its schematic diagram!

As we have seen, the pursuit of steeper roll-off rates may require compromise in some of the other traits of a filter, such as the consistency[8] of its passband and/or of its stopband. What has not been discussed so far, however, are some of the other traits such as these:

- **Phase** – this refers to the phase shift between output and input at any given frequency, usually plotted as a frequency-domain function (i.e. phase shift on the vertical axis versus signal frequency on the horizontal)
- **Step response** – this refers to the time-domain plot of a filter's output signal when its input is the rising edge of a perfect square wave
- **Group delay** – this is related to, though not identical to, the time delay experienced by a signal propagating through a filter network

For most applications the primary characteristic for any filter network is its ability to exclude certain signal frequencies while passing others, but these other characteristics may be quite important as well. For example, if a filtered signal is compared, mixed, or otherwise processed along with either the original (unfiltered) signal or a signal filtered by a different network from the same original signal, phase shift and group delay effects may significantly affect the outcome. Likewise, if a filter network routinely experiences transient (pulse) signals at its input, a "rippling" step response could be problematic.

With all these criteria to consider, and the mathematics of filter behavior being so complicated, practical filter design is usually a task requiring special-purpose filter-design computer software.

[8] This trait is sometimes referred to as *monotonicity*, a "monotonic" function being one where the sign of the slope never reverses. A low-pass filter with a monotonic response, for example, will always output *less* as frequency increases, and never go in the other direction. Filters with "ripple" in their Bode plots are non-monotonic in that their outputs rise and fall as frequency varies in the same direction over certain ranges.

3.7 Filter transfer functions

A mathematical tool used to describe the behavior of all kinds of systems, but especially filter networks, is something called a *transfer function*. This is conceptually similar to *gain* – being the ratio of a system's output to its input – but what makes transfer functions unique is that they are always mathematical functions of a variable named s rather than being constant values as gains often are. In order to understand what a transfer function represents, we must first familiarize ourselves with s. After that, we will apply this concept to all the basic forms of filter networks (low-pass, high-pass, band-pass, and band-stop) so you may learn by example.

To begin, we will review the concept of *reactance*, which is the degree to which a component opposes the flow of alternating current (AC) by way of storing and releasing energy. Reactance is a mathematical function of frequency (f), which means its value varies as frequency varies. A common way to denote the dependence of one variable on another is to show the dependent variable followed by the influencing variable in parentheses. Thus, if we want to express the fact that reactance (X) is a function of frequency (f), we could simply write $X(f)$ and say "reactance is a function of frequency". This dependence is also shown by the well-known inductive and capacitive reactance formulae, by virtue of the simple fact that reactance is written on the left-hand side of the "equals" symbol and frequency is found on the right-hand side of each:

$$X_L = 2\pi f L \qquad\qquad X_C = \frac{1}{2\pi f C}$$

If we specify signal frequency in units of *radians per second* rather than *cycles per second*, we label it as ω (i.e. "angular velocity" or "natural frequency") and are able to more simply express reactance as a function of ω, because there are 2π radians in every full cycle (i.e. $\omega = 2\pi f$):

$$X_L = \omega L \qquad\qquad X_C = \frac{1}{\omega C}$$

With the introduction of the imaginary operator (i or j, both equal to $\sqrt{-1}$) these reactance functions now describe the *complex impedance* of an inductor and a capacitor, respectively, which means we use Z to express the result rather than X:

$$Z_L = j\omega L \qquad\qquad Z_C = \frac{1}{j\omega C}$$

Impedance is a more complete description of a component's opposition to the passage of current than reactance, and it may be algebraically combined with resistance (which is always a *real* quantity) to analyze the steady-state behavior of AC networks. Some readers may be more familiar with the "polar" representations of inductive and capacitive impedance, where the imaginary (j) aspect shows as a phase angle either pointed straight up (90^o) or straight down (270^o) on a phasor diagram:

$$Z_L = (2\pi f L) \angle 90^o \qquad\qquad Z_C = \left(\frac{1}{2\pi f C}\right) \angle 270^o$$

Unfortunately, though, these "shorthand" notations for phase angle don't lend themselves to algebraic manipulation, and so $Z_L = j\omega L$ and $Z_C = \frac{1}{j\omega C}$ are the more mathematically useful forms.

One more extension is to represent not only the frequency of the signal (ω) and the imaginary operator (j) but also a new factor describing the *growth/decay rate*[9] of the electrical signal symbolized by the variable σ:

$$Z_L = (\sigma + j\omega)L \qquad\qquad Z_C = \frac{1}{(\sigma + j\omega)C}$$

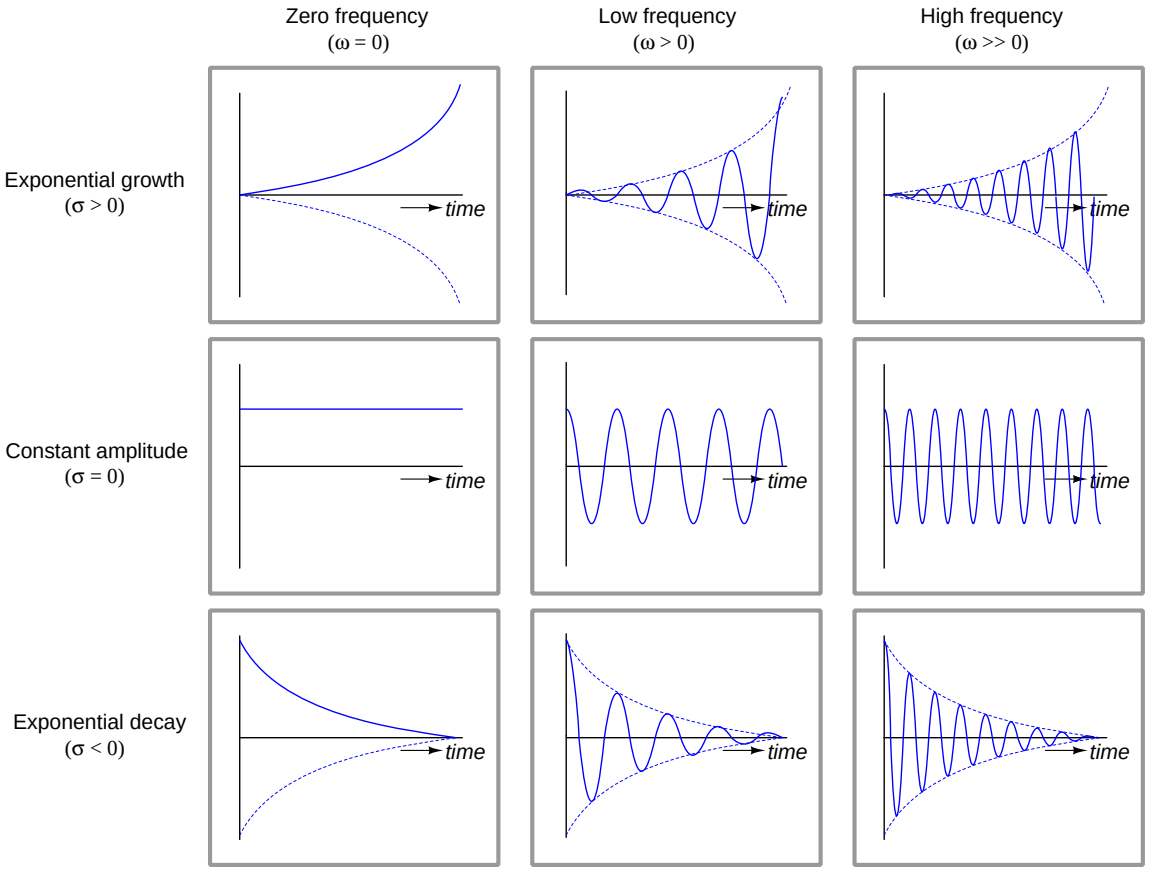
This growth/decay rate is useful not only for describing DC signals that rise and fall exponentially, but also AC signals whose amplitude envelopes similarly rise and fall exponentially. For the sake of convenience, the real quantity σ and the imaginary quantity $j\omega$ are combined into a single complex variable named s, and with this new variable we may completely describe a component's opposition to growing/delaying and/or alternating currents in a very simple form:

$$Z_L = sL \qquad\qquad Z_C = \frac{1}{sC}$$

The utility of s as the independent variable is that we may predict the impedance of any inductor or capacitor for not just steady-state sinusoidal signals but also for decaying or growing DC signals, as well as for decaying or growing sinusoidal signals. This gives us a more complete mathematical description of any network containing inductors and/or capacitors. When applied to filter networks, it means we may describe the network's filtering behavior over a wide range of possible signal conditions.

[9] Also known as the *Neper frequency*, since it is expressed in units of inverse time (reciprocal of seconds) just like ω and f. For those familiar with the concept of a *time constant* (τ), you will be pleased to learn that $\sigma = \frac{1}{\tau}$.

Complex frequency is, well, *complicated* to understand and so deserves some elaboration. Below we see an illustration showing various electrical signals (e.g. voltages) described by different combinations of σ and ω values:



Frequency directly follows the value of ω, with $\omega = 0$ representing a DC condition. A steady-state condition of either DC or AC happens when $\sigma = 0$. When σ is positive, the signal's amplitude grows exponentially; when σ is negative, the signal's amplitude decays in inverse-exponential fashion.

The value of s by itself merely describes the characteristics of a signal over time. However, if we know how various components respond to s we have a way to describe how entire networks of those components will respond to applied signals. Resistors, of course, always exhibit a constant opposition to current regardless of growth/decay rate or frequency, and so we will find that s does not appear as an independent variable for Z_R as it does for Z_L and Z_C:

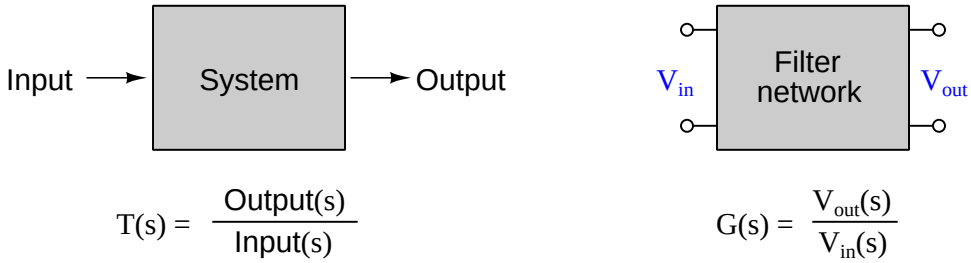
$$Z_R = R \qquad Z_L = sL \qquad Z_C = \frac{1}{sC}$$

All the properties of series and parallel networks become useful for combining individual component impedances to arrive at total network impedance, and also to write equations based on viewing the circuit as a voltage or current divider.

With this we (finally) arrive at the subject of this section: a *transfer function* is simply the ratio of any linear system's output compared to its input, expressed as a function of s. Readers familiar with electronic amplifiers may recognize a similar concept known as *gain*, but the difference between gain and transfer is that gain is typically a fixed quantity for the circuit in question, in the case of amplifiers usually determined by resistor values. When the gain in question varies as a function of signal frequency and/or growth/decay rate, though, and we are able to express it as a mathematical function with s as the independent variable, we have a powerful tool for analyzing the behavior of that system over a wide range of signal conditions.

Filter networks are excellent candidates for characterization by transfer function, because we know the impedance of reactive components varies with frequency (ω) as well as with growth/decay rate (σ). In order to derive the transfer function for any given filter network, we must describe the impedance of its components in terms of s and then write a "gain" equation (or, more often, a "voltage divider" equation) describing how the output signal compares to the input signal.

Transfer functions are often expressed as capital-letter functions of s, such as $H(s)$, $G(s)$, or $T(s)$. We will regard each filter network as a voltage divider when deriving its transfer function:



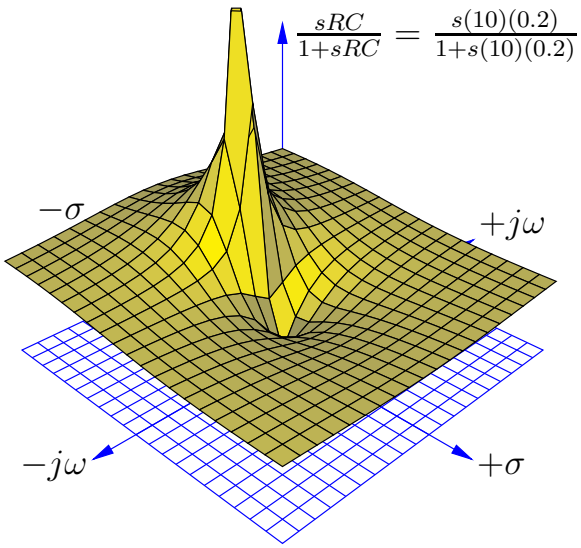
In following subsections we will see examples of transfer functions derived for several different filter networks. For each one the general strategy is to express the impedance of each component as a function of s, and then write an equation for the network's output and input voltages viewing the network as a *voltage divider*. Good advice for readers is to peruse these examples to recognize the techniques applied in each case, and then *try to do them on your own*. Remember that you have not mastered any concept until you can correctly and reliably apply it on your own without assistance!

You will find that most of the "work" in writing a transfer function consists of properly reducing rational (i.e. fraction-based) expressions to their simplest form.

3.7.1 Transfer function poles and zeros

Two terms commonly use to describe characteristics of transfer functions are *poles* and *zeros*. These describe values of s resulting in either infinite gain or zero gain, respectively. Any value of s resulting in no output from the system is called a *zero* of that transfer function, mathematically being any condition resulting in a zero-value numerator[10]. Conversely, any value of s resulting in the denominator of the transfer function being zero is called a *pole*. A "zero" means a particular input signal condition where nothing exits the filter network; a "pole" means a particular condition where the network is able to output any amount of signal despite having no input signal. At first you may think that a "pole" condition is impossible for an electrical filter network, since how can a passive network generate an output signal with no input signal? If we consider the ability of reactive components to store and release energy, however, we see how such a thing could happen: imagine the capacitor(s) and/or inductor(s) of a filter network being "pre-charged" with energy, and then short-circuiting the input terminals of the filter network to ensure zero input signal: whatever response we get at the output signal from that stored energy will represent a "pole" condition. Generally speaking, the more reactive components are present within a filter network, the more unique "pole" conditions exist for its transfer function.

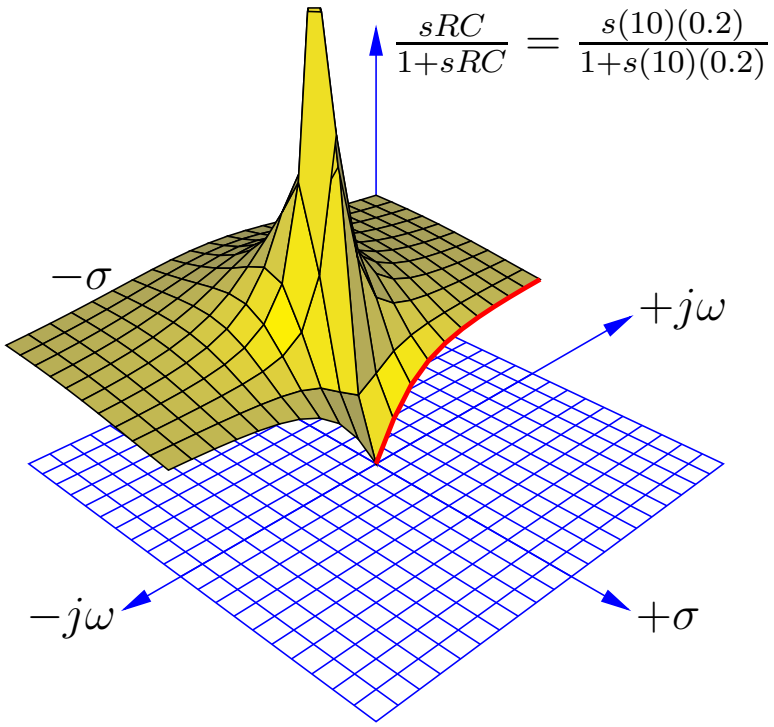
Poles and zeros are easier to understand when plotted on three-dimensional graphs: the real and imaginary portions of the s variable occupying the horizontal axes, and the magnitude of the transfer function fraction displayed as height. Here is a *pole-zero plot* for the transfer function $\frac{s2}{1+s2}$ which happens to be the transfer function for a high-pass RC filter with a 10-Ohm resistor and a 0.2-Farad capacitor:



This transfer function's pole appears as a "spike" while its zero appears as a "pit", both features appearing to stretch the surface of the three-dimensional function as though it were a rubber mat held up by a pole at one location and pinned down by an anchor at another.

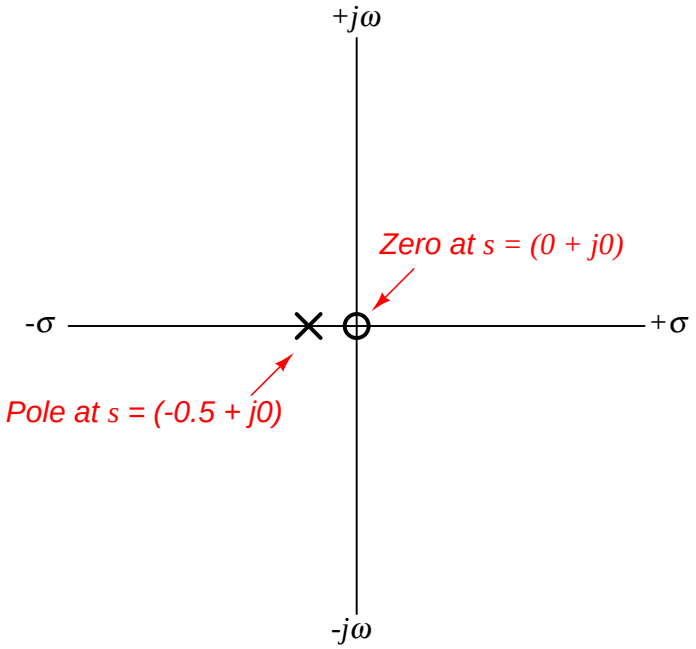
[10] For any polynomial function, any value for the independent variable resulting in zero is called a *root*. The higher-degree the polynomial, the more roots it will have. For transfer functions, then, numerator roots represent zeros while denominator roots represent poles.

One of the valuable insights provided by a three-dimensional pole-zero plot is the system's response to an input signal of constant magnitude and varying frequency. This is commonly referred to as the *frequency response* of the system, otherwise known as a *Bode plot*. We may trace the Bode plot for this system by revealing a cross-sectional slice of the three-dimensional surface along the vertical plane where $\sigma = 0$ (i.e. showing how the system responds to sinusoidal waves of varying frequency that neither grows nor decays over time):



Only one-half of the pole-zero surface has been plotted here, in order to better reveal the cross-section along the $j\omega$ axis. The bold, red curve traces the edge of the transfer function surface as it begins at zero frequency (DC) to increasingly positive values of $j\omega$. The red trace is therefore the Bode plot for this high-pass filter, starting at 0% at a frequency of zero and approaching 100% (i.e. $V_{out} = V_{in}$) as frequency increases.

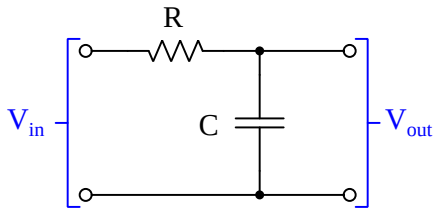
As insightful as three-dimensional pole-zero plots are, they are laborious to plot by hand, and even with the aid of a computer may require significant[11] time to set up. For this reason, pole-zero plots have traditionally been drawn in a two-dimensional rather than three-dimensional format, from a "bird's eye" view looking down at the s-plane. Since this view hides any features of height, poles and zeros are instead located on the s-plane by $\times$ and $\bigcirc$ symbols, respectively. An example of a traditional pole-zero plot for our high-pass filter appears here:



Admittedly, this type of pole-zero plot is much less interesting to view than a three-dimensional surface plotted by computer, but nevertheless contains useful information about the system. The single pole lying on the real (σ) axis tells us the system will not self-oscillate (i.e. $\omega = 0$ at the pole), and that it is inherently stable: when subjected to a pulse, its natural tendency is to decay to a stable value over time (i.e. $\sigma < 0$). The single zero lying at the origin (i.e. $\sigma = 0$ and $\omega = 0$) tells us the system outputs nothing when the input signal is unvarying.

[11] My first pole-zero plot using the `ePiX` C++ mathematical visualization library took several hours to get it just right. Subsequent plots went a lot faster, of course, but they still require substantial amounts of time to adjust for a useful and aesthetically pleasing appearance.

3.7.2 Transfer function for an RC low-pass filter



$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_C(s)}{Z_R(s) + Z_C(s)}$$

$$G(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}}$$

$$G(s) = \frac{\frac{1}{sC}}{\frac{sRC}{sC} + \frac{1}{sC}}$$

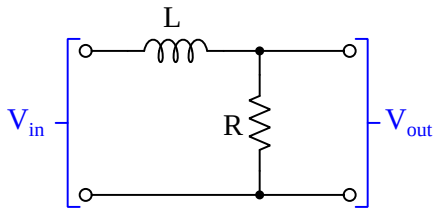
$$G(s) = \frac{\frac{1}{sC}}{\frac{sRC+1}{sC}}$$

$$G(s) = \frac{1}{sRC + 1}$$

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{1}{0+1}$ which means the filter network passes 100% of the input signal to the output terminals. At the other extreme, the transfer function value falls to zero as s approaches infinity. This is exactly as we would expect for a low-pass filter network: passing low-frequency signals and attenuating high-frequency signals.

This filter network has a "zero" at infinite frequency and/or infinite growth/decay rate, where any rapidly-changing input signal is "shorted" by the capacitor's low reactance value. It has a "pole" where s is equal to $-\frac{1}{RC}$, which means a condition with a short-circuited input where the pre-charged capacitor is discharging itself through the resistance, creating an inverse-exponential decay from the original voltage to zero over time. This $-\frac{1}{RC}$ expression should be familiar to anyone who has studied RC time-delay circuits, where decaying quantities follow the form $e^{-\frac{1}{RC}}$.

3.7.3 Transfer function for an LR low-pass filter



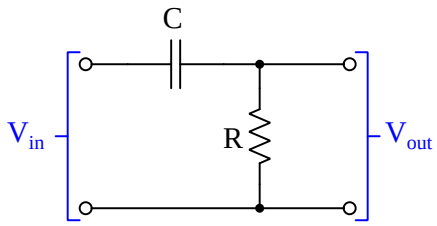
$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_R(s)}{Z_R(s) + Z_L(s)}$$

$$G(s) = \frac{R}{R + sL}$$

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{R}{R+0}$ which means the filter network passes 100% of the input signal to the output terminals. At the other extreme, the transfer function value falls to zero as s approaches infinity. This is exactly as we would expect for a low-pass filter network: passing low-frequency signals and attenuating high-frequency signals.

This filter network has a "zero" at infinite frequency and/or infinite growth/decay rate, where any rapidly-changing input signal is "blocked" by the inductor's high reactance value. It has a "pole" where s is equal to $-\frac{R}{L}$, which means a condition with a short-circuited input where the pre-charged inductor is discharging itself through the resistance, creating an inverse-exponential decay from the original voltage to zero over time. This $-\frac{R}{L}$ expression should be familiar to anyone who has studied LR time-delay circuits, where decaying quantities follow the form $e^{-\frac{1}{\frac{L}{R}}}$.

3.7.4 Transfer function for an RC high-pass filter



$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_R(s)}{Z_R(s) + Z_C(s)}$$

$$G(s) = \frac{R}{R + \frac{1}{sC}}$$

$$G(s) = \frac{R}{\frac{sRC}{sC} + \frac{1}{sC}}$$

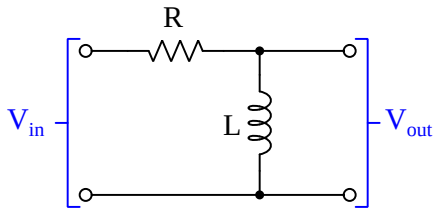
$$G(s) = \frac{R}{\frac{sRC+1}{sC}}$$

$$G(s) = \frac{sRC}{sRC + 1}$$

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{0}{0+1}$ which means the filter network passes 0% of the input signal to the output terminals. At the other extreme, the transfer function rises to 100% as s approaches infinity. This is exactly as we would expect for a high-pass filter network: passing high-frequency signals and attenuating low-frequency signals.

This filter network has a "zero" at zero frequency and zero growth/decay rate, where any steady input signal is "blocked" by the capacitor's high reactance value. It has a "pole" where s is equal to $-\frac{1}{RC}$, which means a condition with a short-circuited input where the pre-charged capacitor is discharging itself through the resistance, creating an inverse-exponential decay from the original voltage to zero over time. This $-\frac{1}{RC}$ expression should be familiar to anyone who has studied RC time-delay circuits, where decaying quantities follow the form $e^{-\frac{1}{RC}}$.

3.7.5 Transfer function for an LR high-pass filter



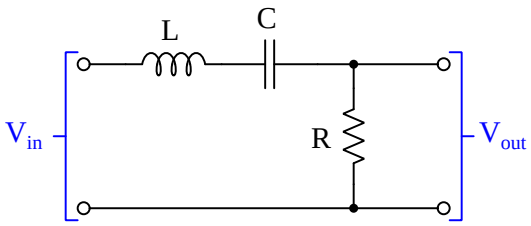
$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_L(s)}{Z_R(s) + Z_L(s)}$$

$$G(s) = \frac{sL}{R + sL}$$

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{0}{R+0}$ which means the filter network passes 0% of the input signal to the output terminals. At the other extreme, the transfer function rises to 100% as s approaches infinity. This is exactly as we would expect for a high-pass filter network: passing high-frequency signals and attenuating low-frequency signals.

This filter network has a "zero" at zero frequency and zero growth/decay rate, where any steady input signal is "shorted" by the inductor's low reactance value. It has a "pole" where s is equal to $-\frac{R}{L}$, which means a condition with a short-circuited input where the pre-charged inductor is discharging itself through the resistance, creating an inverse-exponential decay from the original voltage to zero over time. This $-\frac{R}{L}$ expression should be familiar to anyone who has studied LR time-delay circuits, where decaying quantities follow the form $e^{-\frac{1}{\frac{L}{R}}}$.

3.7.6 Transfer function for an RLC series band-pass filter



$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_R(s)}{Z_R(s) + Z_L(s) + Z_C(s)}$$

$$G(s) = \frac{R}{R + sL + \frac{1}{sC}}$$

$$G(s) = \frac{R}{\frac{sRC}{sC} + \frac{s^2LC}{sC} + \frac{1}{sC}}$$

$$G(s) = \frac{R}{\frac{sRC+s^2LC+1}{sC}}$$

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

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{0}{0+0+1}$ which means the filter network passes 0% of the input signal to the output terminals. At the other extreme, the transfer function also approaches 0% as s approaches infinity, because the s^2 denominator term grows to a much larger value than the s term in the numerator. This is exactly as we would expect for a band-pass filter network: attenuating high-frequency signals as well as attenuating low-frequency signals.

The one signal frequency where this filter passes 100% of the input signal is where the s^2LC term is equal to -1 so as to cancel out the "1" constant, leaving the transfer function equal to $\frac{sRC}{sRC}$. Solving for that value of s:

$$s^2LC = -1$$

$$s^2 = \frac{-1}{LC}$$

$$s = \sqrt{\frac{-1}{LC}}$$

$$s = \frac{j}{\sqrt{LC}}$$

Since we know that $s = \sigma + j\omega$, and we're primarily interested in an AC signal frequency (ω) rather than a DC decay rate (σ), we can set σ equal to zero and just solve for ω:

$$j\omega = \frac{j}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

This result should look familiar, as we know the resonant frequency of a simple LC network is equal to:

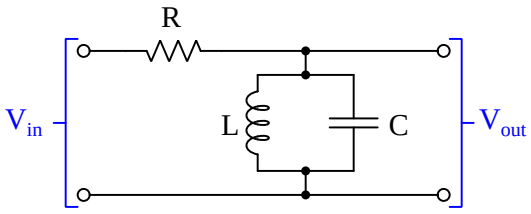
$$f_r = \frac{1}{2\pi\sqrt{LC}}$$

Knowing that ω is just another way to express $2\pi f$, we can cast this resonant frequency formula in terms of angular velocity (radians per second) instead of regular frequency (cycles per second):

$$2\pi f_r = \frac{1}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

3.7.7 Transfer function for an RLC shunt band-pass filter



$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_L(s)||Z_C(s)}{Z_R(s) + (Z_L(s)||Z_C(s))}$$

First we need to derive an expression for the parallel combination of $Z_L(s)$ and $Z_C(s)$.

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{1}{Z_L(s)} + \frac{1}{Z_C(s)}}$$

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{1}{sL} + sC}$$

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{1}{sL} + \frac{s^2LC}{sL}}$$

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{s^2LC+1}{sL}}$$

$$Z_L(s)||Z_C(s) = \frac{sL}{s^2LC + 1}$$

With this expression in hand, we may now proceed as we did with the other filter examples, setting up the transfer function as a voltage-divider ratio:

$$G(s) = \frac{\frac{sL}{s^2LC+1}}{R + \frac{sL}{s^2LC+1}}$$

$$G(s) = \frac{\frac{sL}{s^2LC+1}}{\frac{R(s^2LC+1)}{s^2LC+1} + \frac{sL}{s^2LC+1}}$$

$$G(s) = \frac{\frac{sL}{s^2LC+1}}{\frac{s^2RLC+sL+R}{s^2LC+1}}$$

$$G(s) = \frac{sL}{s^2RLC + sL + R}$$

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{0}{0+0+R}$ which means the filter network passes 0% of the input signal to the output terminals. At the other extreme, the transfer function also approaches 0% as s approaches infinity, because the s^2 denominator term grows to a much larger value than the s term in the numerator. This is exactly as we would expect for a band-pass filter network: attenuating high-frequency signals as well as attenuating low-frequency signals.

The one signal frequency where this filter passes 100% of the input signal is where the s^2RLC term is equal to $-R$ so as to cancel out the "R" constant, leaving the transfer function equal to $\frac{sL}{sL}$. Solving for that value of s:

$$s^2RLC = -R$$

$$s^2LC = -1$$

$$s^2 = \frac{-1}{LC}$$

$$s = \sqrt{\frac{-1}{LC}}$$

$$s = \frac{j}{\sqrt{LC}}$$

Since we know that $s = \sigma + j\omega$, and we're primarily interested in an AC signal frequency (ω) rather than a DC decay rate (σ), we can set σ equal to zero and just solve for ω:

$$j\omega = \frac{j}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

This result should look familiar, as we know the resonant frequency of a simple LC network is equal to:

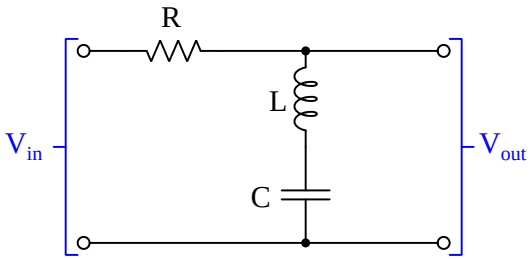
$$f_r = \frac{1}{2\pi\sqrt{LC}}$$

Knowing that ω is just another way to express $2\pi f$, we can cast this resonant frequency formula in terms of angular velocity (radians per second) instead of regular frequency (cycles per second):

$$2\pi f_r = \frac{1}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

3.7.8 Transfer function for an RLC series band-stop filter



$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_L(s) + Z_C(s)}{Z_R(s) + Z_L(s) + Z_C(s)}$$

$$G(s) = \frac{sL + \frac{1}{sC}}{R + sL + \frac{1}{sC}}$$

$$G(s) = \frac{\frac{s^2LC}{sC} + \frac{1}{sC}}{\frac{sRC}{sC} + \frac{s^2LC}{sC} + \frac{1}{sC}}$$

$$G(s) = \frac{\frac{s^2LC+1}{sC}}{\frac{s^2LC+sRC+1}{sC}}$$

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

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{0+1}{0+0+1}$ which means the filter network passes 100% of the input signal to the output terminals. At the other extreme, the transfer function also approaches 100% as s approaches infinity, because the s^2LC terms in both numerator and denominator overshadow both the s and constant terms in the denominator. This is exactly as we would expect for a band-stop filter network: passing high-frequency signals as well as passing low-frequency signals.

The one signal frequency where this filter passes none of the input signal is where the s^2LC term is equal to -1 so as to make the numerator zero, leaving the transfer function equal to $\frac{1}{sRC}$. Solving for that value of s:

$$s^2LC = -1$$

$$s^2 = \frac{-1}{LC}$$

$$s = \sqrt{\frac{-1}{LC}}$$

$$s = \frac{j}{\sqrt{LC}}$$

Since we know that $s = \sigma + j\omega$, and we're primarily interested in an AC signal frequency (ω) rather than a DC decay rate (σ), we can set σ equal to zero and just solve for ω:

$$j\omega = \frac{j}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

This result should look familiar, as we know the resonant frequency of a simple LC network is equal to:

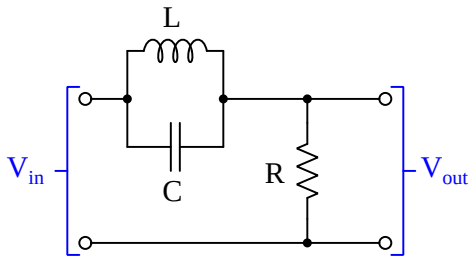
$$f_r = \frac{1}{2\pi\sqrt{LC}}$$

Knowing that ω is just another way to express $2\pi f$, we can cast this resonant frequency formula in terms of angular velocity (radians per second) instead of regular frequency (cycles per second):

$$2\pi f_r = \frac{1}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

3.7.9 Transfer function for an RLC shunt band-stop filter



$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Z_R(s)}{Z_R(s) + (Z_L(s)||Z_C(s))}$$

First we need to derive an expression for the parallel combination of $Z_L(s)$ and $Z_C(s)$.

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{1}{Z_L(s)} + \frac{1}{Z_C(s)}}$$

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{1}{sL} + sC}$$

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{1}{sL} + \frac{s^2LC}{sL}}$$

$$Z_L(s)||Z_C(s) = \frac{1}{\frac{s^2LC+1}{sL}}$$

$$Z_L(s)||Z_C(s) = \frac{sL}{s^2LC+1}$$

With this expression in hand, we may now proceed as we did with the other filter examples, setting up the transfer function as a voltage-divider ratio:

$$G(s) = \frac{R}{R + \frac{sL}{s^2LC+1}}$$

$$G(s) = \frac{R}{\frac{R(s^2LC+1)}{s^2LC+1} + \frac{sL}{s^2LC+1}}$$

$$G(s) = \frac{R}{\frac{s^2RLC+sL+R}{s^2LC+1}}$$

$$G(s) = \frac{R(s^2LC+1)}{s^2RLC+sL+R}$$

$$G(s) = \frac{s^2RLC+R}{s^2RLC+sL+R}$$

When the input voltage signal is absolutely constant (i.e. purely DC, where $s = 0$) the transfer function has a value of $\frac{0+R}{0+0+R}$ which means the filter network passes 100% of the input signal to the output terminals. At the other extreme, the transfer function also approaches 100% as s approaches infinity, because the $s^2LC + R$ terms in both numerator and denominator overshadow the sL term in the denominator. This is exactly as we would expect for a band-stop filter network: passing high-frequency signals as well as passing low-frequency signals.

The one signal frequency where this filter passes none of the input signal is where the s^2RLC term is equal to $-R$ so as to cancel out the "R" constant, leaving the transfer function equal to $\frac{0}{sL}$. Solving for that value of s:

$$s^2RLC = -R$$

$$s^2LC = -1$$

$$s^2 = \frac{-1}{LC}$$

$$s = \sqrt{\frac{-1}{LC}}$$

$$s = \frac{j}{\sqrt{LC}}$$

Since we know that $s = \sigma + j\omega$, and we're primarily interested in an AC signal frequency (ω) rather than a DC decay rate (σ), we can set σ equal to zero and just solve for ω:

$$j\omega = \frac{j}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

This result should look familiar, as we know the resonant frequency of a simple LC network is equal to:

$$f_r = \frac{1}{2\pi\sqrt{LC}}$$

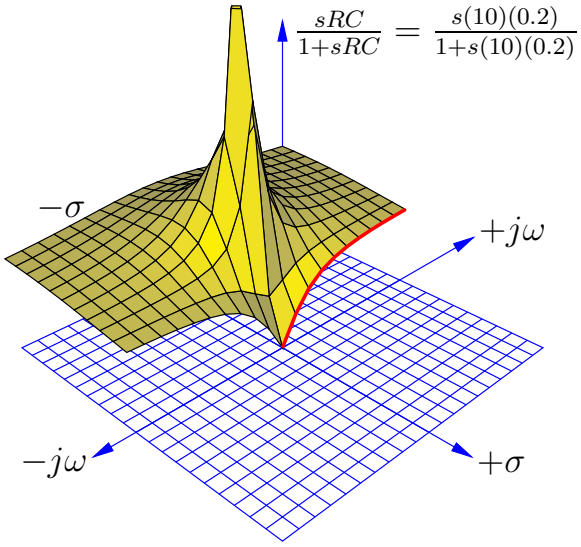
Knowing that ω is just another way to express $2\pi f$, we can cast this resonant frequency formula in terms of angular velocity (radians per second) instead of regular frequency (cycles per second):

$$2\pi f_r = \frac{1}{\sqrt{LC}}$$

$$\omega = \frac{1}{\sqrt{LC}}$$

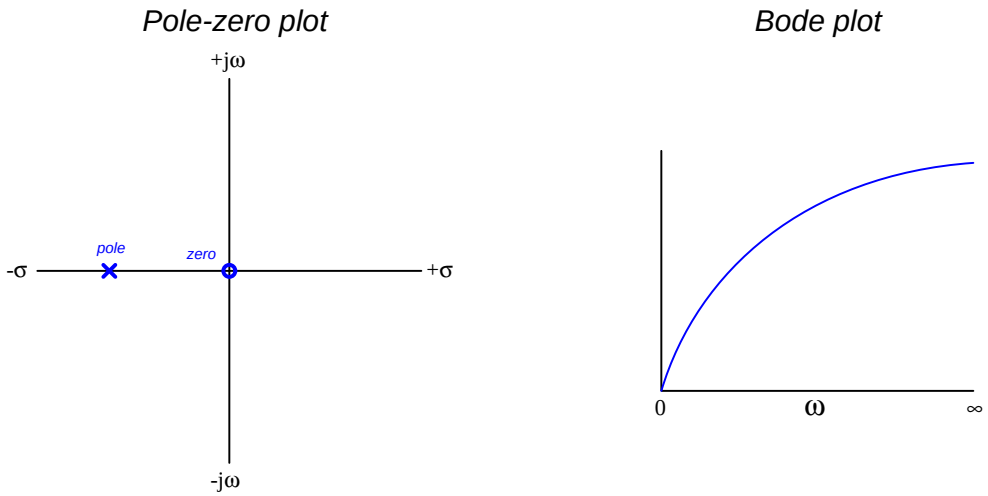
3.8 Filter poles and zeroes

As mentioned previously, one of the valuable insights provided by a pole-zero plot is the system's response to an input signal of constant magnitude and varying frequency – commonly referred to as the *frequency response* of the system usually shown as a *Bode plot*. If we take a cross-sectional slice of the three-dimensional pole-zero plot surface along a vertical plane where $\sigma = 0$, we will see a two-dimensional Bode plot indicating how the system responds to sinusoidal waves of varying frequency and constant peak amplitude):



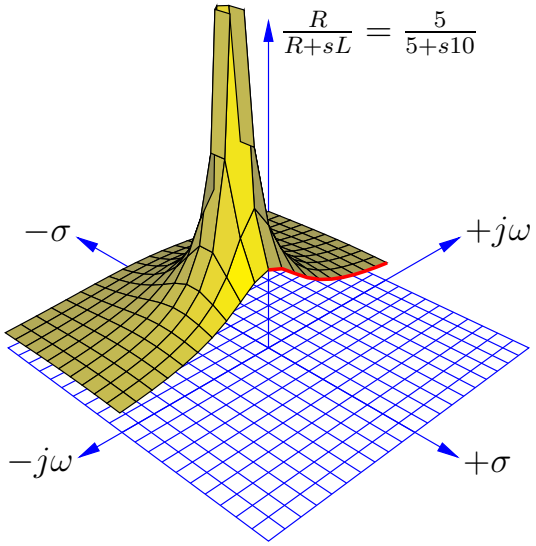
Only one-half of the pole-zero three-dimensional surface has been plotted above, in order to show the cross-section along the $j\omega$ axis where $\sigma = 0$. The bold, red curve traces the edge of the transfer function surface as it begins at zero frequency (DC) to increasingly positive values of $j\omega$, indicating the characteristic frequency response of a single-pole high-pass filter, starting at 0% at a frequency of zero and approaching 100% (i.e. $V_{out} = V_{in}$) as frequency increases.

If we carefully examine a more traditional two-dimensional pole-zero plot for this same filter network where the *s*-plane is parallel to the page, we may picture the shape of the three-dimensional surface created by the single (real) pole and single zero of this high-pass filter's transfer function, and from that we may discern the shape of the Bode plot:

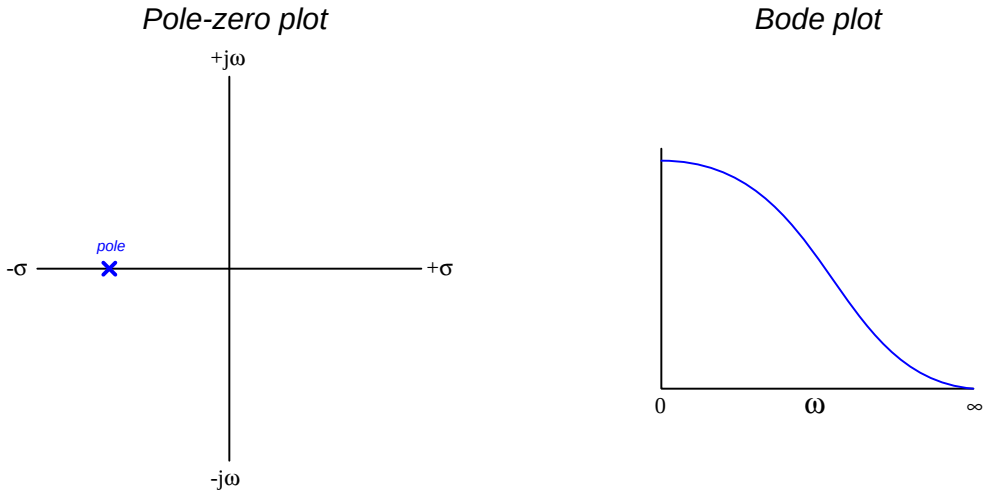


The Bode plot's steep portion on the left-hand side where frequency is low is just the cross-section of the three-dimensional pole-zero surface, the lowest point being where the transfer function's zero pins that surface to the ground (i.e. no signal emerging from the filter at DC, or 0 Hz).

If we do the same with a single-pole low-pass filter network we will once again find that a cross-section of the three-dimensional pole-zero plot yields a Bode plot showing the filter's frequency response for signals having constant amplitude:



Comparing traditional two-dimensional pole-zero plot and Bode plot for this low-pass filter:

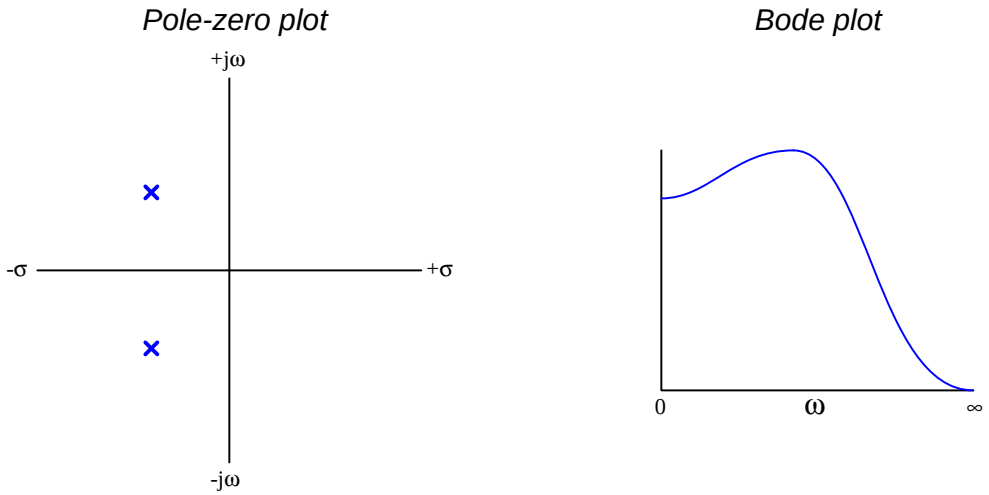


Though no zeroes appear on the pole-zero plot, this filter's transfer function does approach zero as s approaches infinity in any direction, and this explains why the Bode plot decreases toward zero as frequency rises.

The following subsections will illustrate pole-zero and Bode plots for common multi-pole filter networks. In each case the reader is encouraged to mentally visualize the three-dimensional pole-zero

surface as it is stretched by poles and pinned to the ground by zeroes, forming the Bode plot shape along the cross-sectional cut placed at the imaginary axis.

3.8.1 Two-pole low-pass filter plots



$$\text{Transfer function} = \frac{\omega_c^2}{s^2 + \frac{\omega_c}{Q}s + \omega_c^2}$$

Where,

s = Complex frequency ($\sigma + j\omega$), in per-seconds

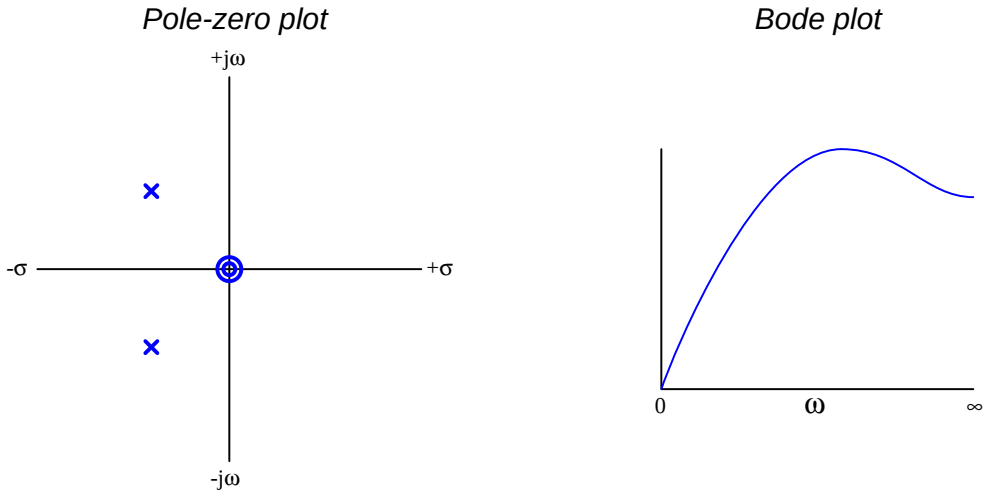
ω_c = Cutoff frequency (a real number), in radians per second

Q = Quality factor of the filter (a real number), unitless

This transfer function's constant numerator equal to the square of the cutoff frequency means there are no finite zeroes. Rather, the function approaches zero as complex frequency goes to infinity in any direction (i.e. as the function's denominator approaches infinity).

Quality factor (Q) helps determine the real value of the poles, which is to say how far from the imaginary axis the poles will lie. A low Q value places the poles farther from the imaginary axis which results in a Bode plot having more a more gradual curvature, and a filter network with gentler rolloff. A high Q value places the poles closer to the imaginary axis which results in a Bode plot having steeper curves and a filter with more aggressive rolloff. Setting Q to infinity causes the fraction $\frac{\omega_c}{Q}$ to go to zero and places the poles exactly along the imaginary axis. Such a filter network would, in theory, oscillate forever at that/those frequency(ies) with no exciting input signal.

3.8.2 Two-pole high-pass filter plots



$$\text{Transfer function} = \frac{s^2}{s^2 + \frac{\omega_c}{Q}s + \omega_c^2}$$

Where,

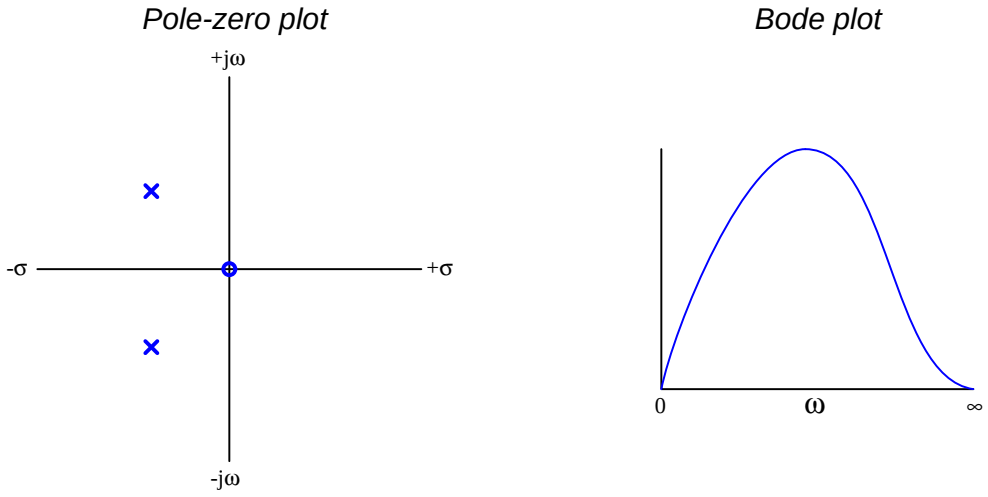
s = Complex frequency ($\sigma + j\omega$), in per-seconds

ω_c = Cutoff frequency (a real number), in radians per second

Q = Quality factor of the filter (a real number), unitless

This transfer function's numerator goes to zero whenever s^2 is zero, and this means *two* overlapping zeroes located at ± 0 radians per second because $\sqrt{0} = \pm 0$.

3.8.3 Two-pole band-pass filter plots



$$\text{Transfer function} = \frac{\frac{\omega_c}{Q} s}{s^2 + \frac{\omega_c}{Q} s + \omega_c^2}$$

Where,

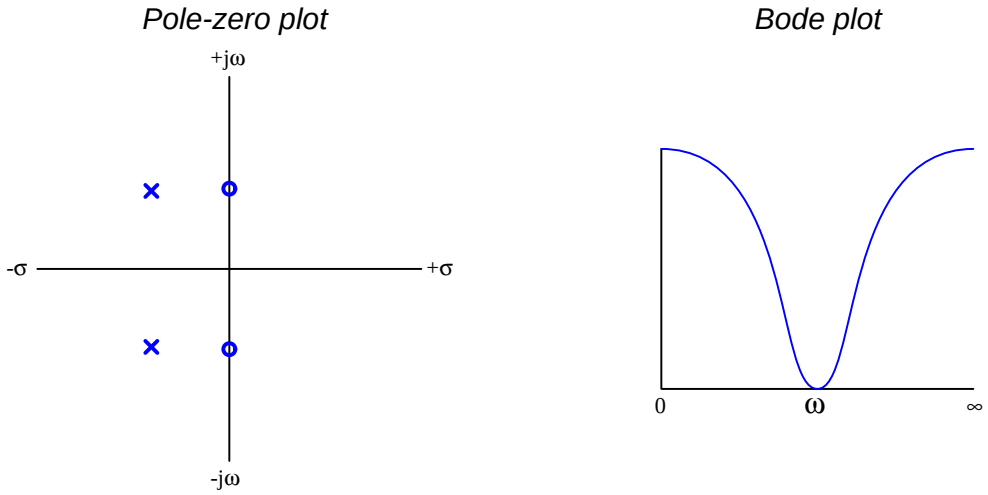
s = Complex frequency ($\sigma + j\omega$), in per-seconds

ω_c = Cutoff frequency (a real number), in radians per second

Q = Quality factor of the filter (a real number), unitless

This transfer function's numerator goes to zero whenever s is zero, but the transfer function as a whole also approaches zero as complex frequency goes to infinity in any direction (i.e. as the function's denominator approaches infinity).

3.8.4 Two-pole band-stop (notch) filter plots



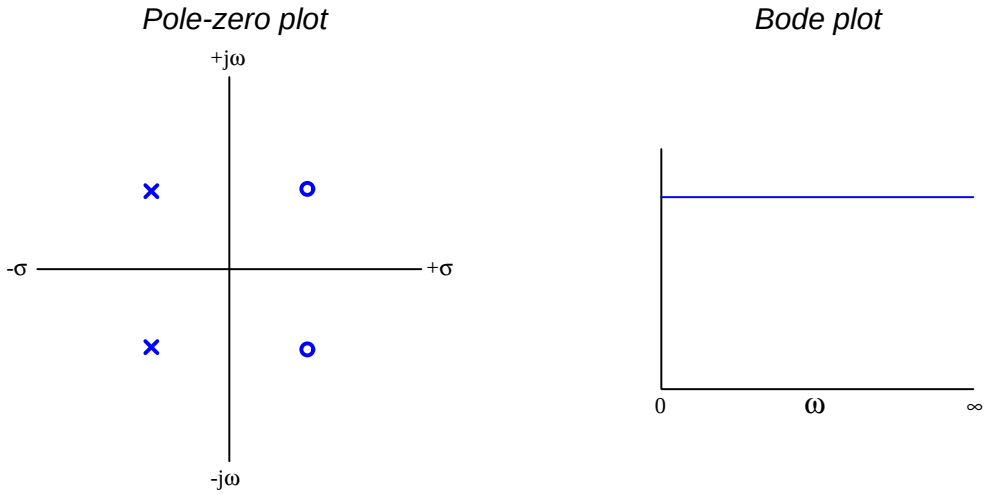
$$\text{Transfer function} = \frac{s^2 + \omega_z^2}{s^2 + \frac{\omega_c}{Q} s + \omega_c^2}$$

Where,

s = Complex frequency ($\sigma + j\omega$), in per-seconds
ω_z = Zero or Notch frequency (a real number), in radians per second
ω_c = Cutoff frequency (a real number), in radians per second
Q = Quality factor of the filter (a real number), unitless

This transfer function's numerator goes to zero whenever $s = \pm j\omega_z$, providing two imaginary zeroes representing frequency values where the notch filter outputs no signal. If the zeroes align horizontally with the poles on the pole-zero plot (i.e. their imaginary parts are equal) then the Bode plot will be symmetrical to the left and right of the notch frequency when viewed on a log-log graph which is the classic notch-type frequency response.

3.8.5 Two-pole all-pass filter plots



$$\text{Transfer function} = \frac{s^2 - \frac{\omega_c}{Q}s + \omega_c^2}{s^2 + \frac{\omega_c}{Q}s + \omega_c^2}$$

Where,

s = Complex frequency ($\sigma + j\omega$), in per-seconds

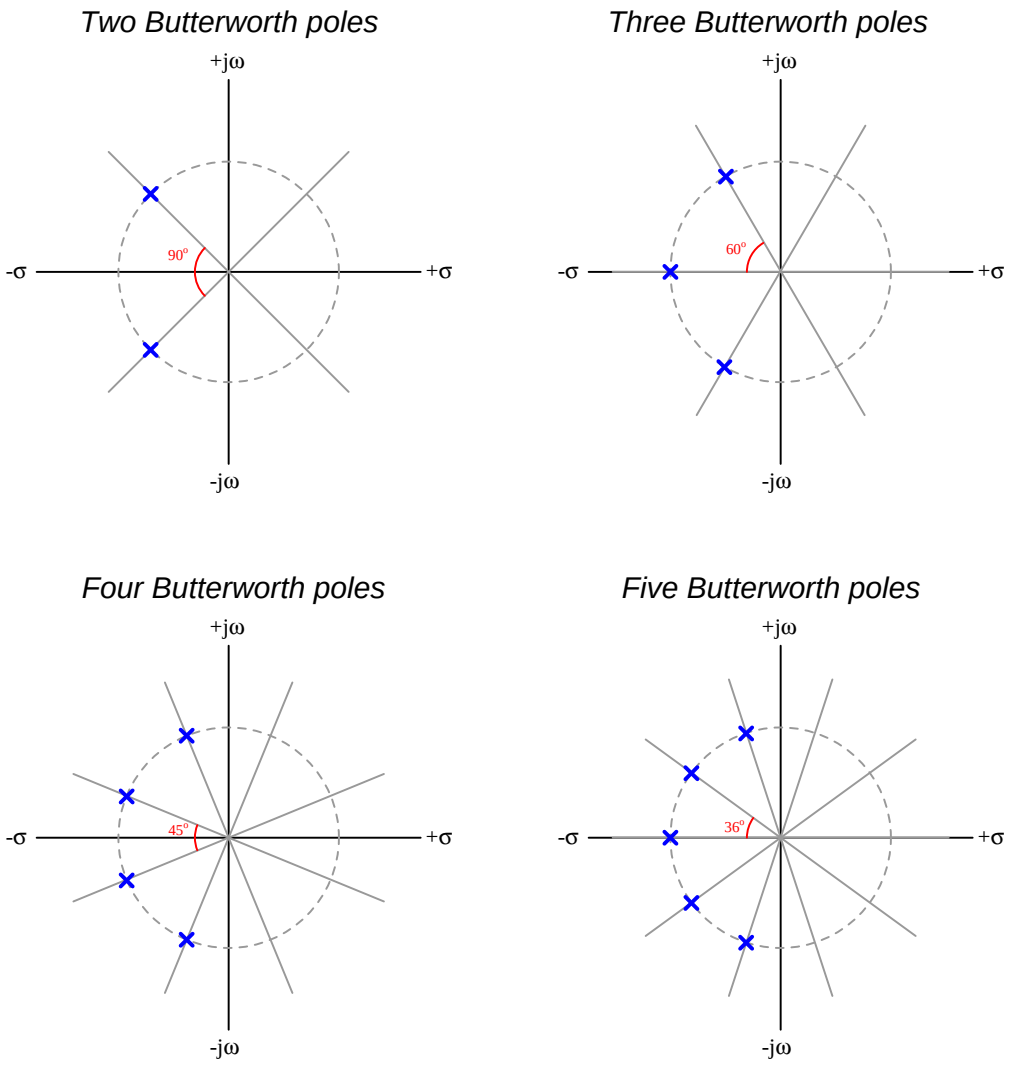
ω_c = Cutoff frequency (a real number), in radians per second

Q = Quality factor of the filter (a real number), unitless

This transfer function's numerator goes to zero at the same imaginary frequency values as the poles, but with oppositely-signed real frequency values which places the zeroes on the right-hand side of the s-plane. With poles and zeroes mirrored opposite each other from the imaginary axis, the Bode plot ends up being a straight and level line. This means the filter passes all frequencies equally well, which may seem pointless for a filter until we realize that *phase shift* between input and output signals and other parameters will *not* be constant for all frequencies. Therefore, all-pass filters are often used in applications where we desire phase shift (but not signal magnitude) to vary as a function of frequency.

3.8.6 Butterworth filter pole locations

Multi-pole filter networks are given a Butterworth characteristic by locating the poles along the circumference of a circle at equal angles of $\frac{1}{2n}$ of one revolution where n is the number of poles:



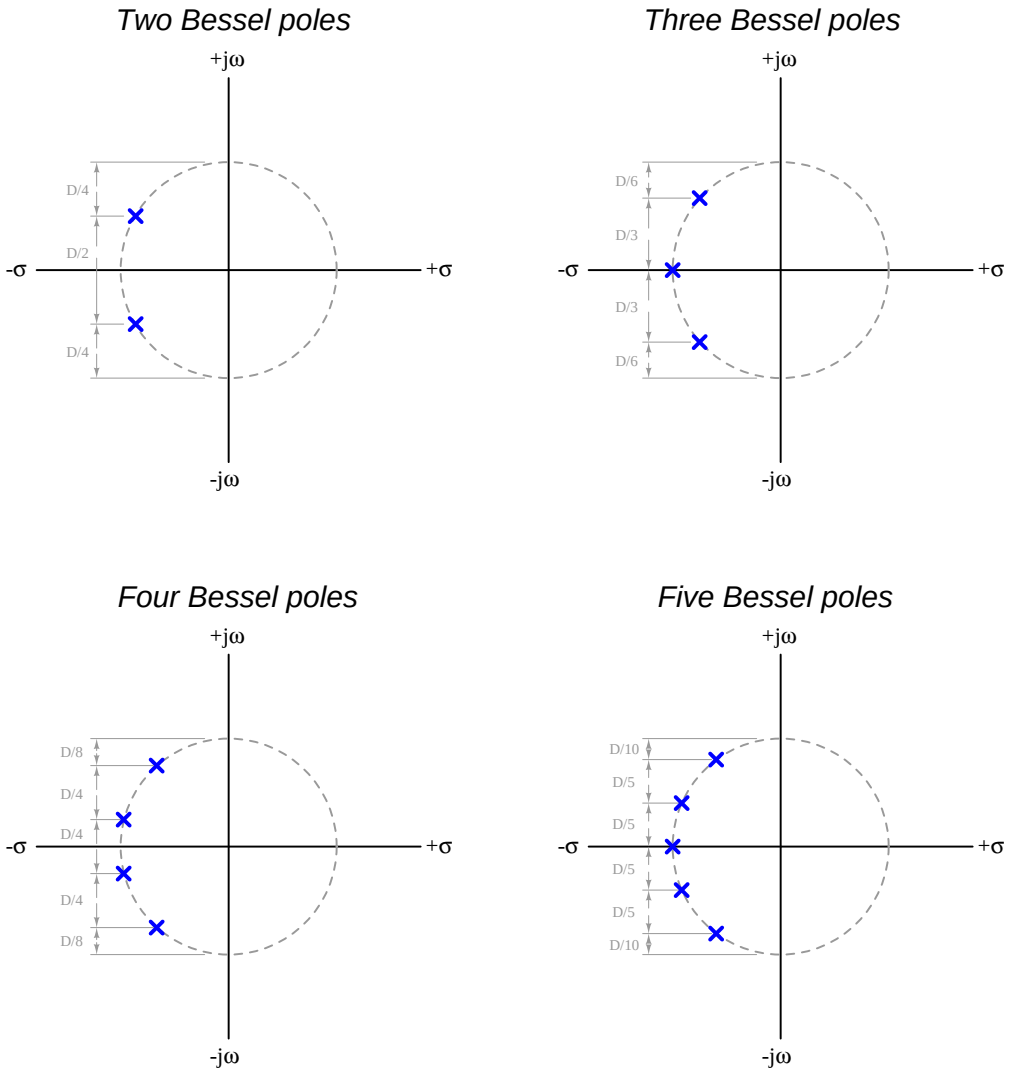
The "maximally-flat" pass-band response typical of Butterworth low-pass and high-pass filter networks is due to this equiangular pole spacing.

Zeroes must be located on the pole-zero diagrams in accordance with the desired filter type, e.g. low-pass, high-pass, etc. and are not shown in these examples[12].

[12] Or, we could say that these zero-less plots all represent *low-pass* filter functions where the only "zeroes" lie at infinite frequency values.

3.8.7 Bessel filter pole locations

Multi-pole filter networks are given a Bessel characteristic by locating the poles along the circumference of a circle symmetrical about the horizontal axis spaced vertically by $\frac{D}{n}$ where D is the circle's diameter and n is the number of poles. The outer-most poles will lie $\frac{D}{2n}$ distance away from the top and bottom of the circle:

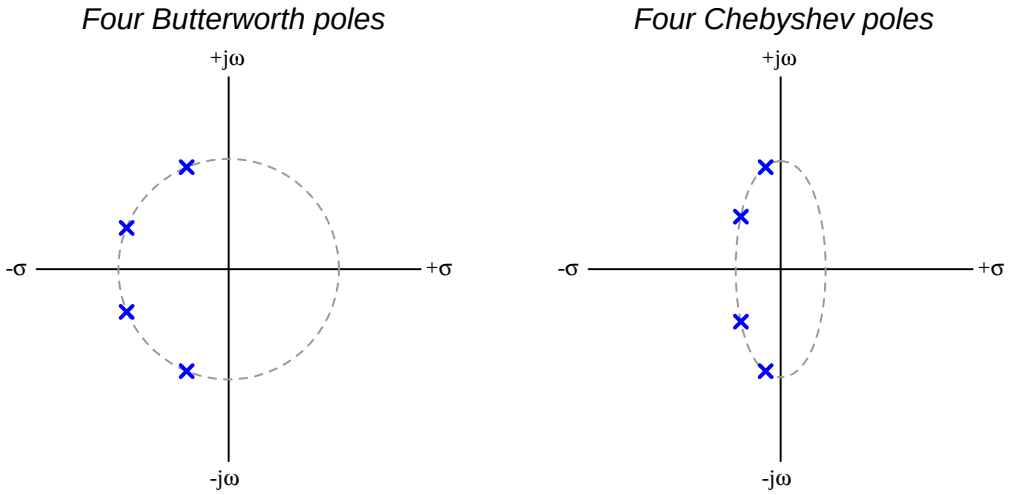


Again, zeroes must be located on the pole-zero diagrams in accordance with the desired filter type, e.g. low-pass, high-pass, etc. and are not shown in these examples[13].

[13] Or, we could say that these zero-less plots all represent *low-pass* filter functions where the only "zeroes" lie at infinite frequency values.

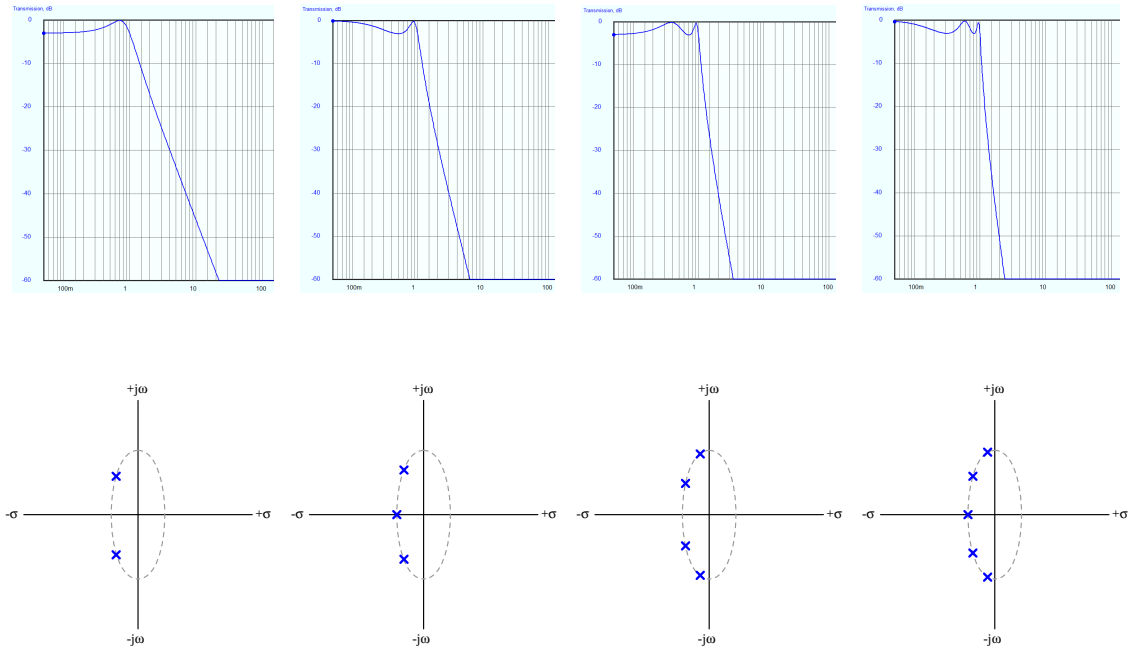
3.8.8 Chebyshev filter pole locations

Multi-pole filters with a Chebyshev characteristic have poles located along the circumference of an ellipse rather than a circle as is the case with Butterworth and Bessel filters:



The Chebyshev pole locations translate to a Bode plot having more "ripple" in the passband. This makes intuitive sense if you imagine poles stretching the three-dimensional pole-zero surface as though it were a rubber sheet and zeroes pinning it down to the *s*-plane like anchors holding a tent to the ground. The equiangular arrangement of poles around the circumference of a circle for a Butterworth characteristic have the effect of making that rubber sheet completely flat as it passes over the imaginary axis, giving Butterworth filter networks their characteristic "maximally-flat" passband response. Chebyshev pole placement along the circumference of an ellipse, by contrast, locate all poles closer to the imaginary axis where they have a more individual and pronounced effect on the shape of that rubber sheet as it passes over the imaginary axis. In other words, the "lumps" seen in the passband portion of the low-pass or high-pass filter's Bode plot stem from the effects of the individual poles

A comparison of Bode plots for low-pass Chebyshev filter networks of different pole numbers makes this clear. From left to right we see the Bode plots for a 2-pole, 3-pole, 4-pole, and 5-pole low-pass Chebyshev filter with a 3 dB ripple passband:

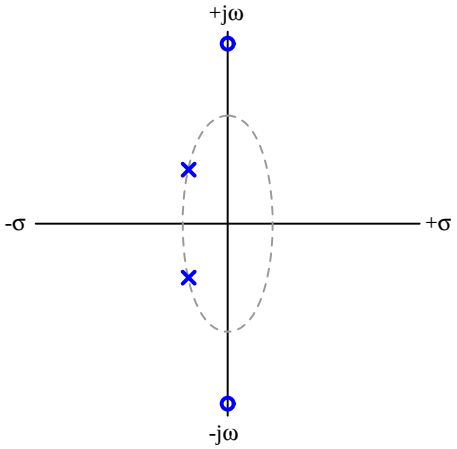


Each "peak" seen in the passband stems from a pole located in the upper half of the s-plane, since Bode plots begin at a frequency value of zero and progress to positive frequency values only. For the 2-pole filter's Bode plot on the far left we see only one peak in the passband because only the upper pole's effect is visible for positive frequencies. For the 3-pole filter's Bode plot (second from left) we see a pronounced peak immediately prior to the rolloff and a gently-sloping peak at zero frequency, representing the two poles having any effect for positive frequencies: one near the cutoff frequency and the other at zero (i.e. on the real axis of the s-plane). The same general pattern applies to the 4-pole and 5-pole Chebyshev filter Bode plots.

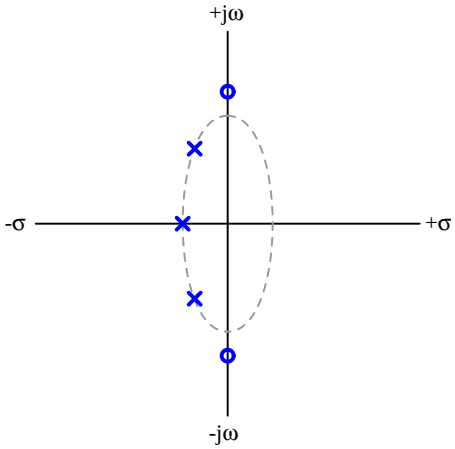
3.8.9 Elliptic filter pole/zero locations

With the exception of the band-stop or notch filter type, Butterworth, Bessel, and Chebyshev filter characteristics may be classified as *all-pole* because their passband, rolloff, and stopband properties are really defined by pole locations only, the zeroes lying either at $s = 0$ or at $s = \infty$. However, we may achieve even more aggressive rolloff rates if we strategically add zeroes to a filter's transfer function, and this is what is done with the so-called *Elliptic* characteristic, also known as *Cauer* in honor of German mathematician and electrical engineer Wilhelm Cauer.

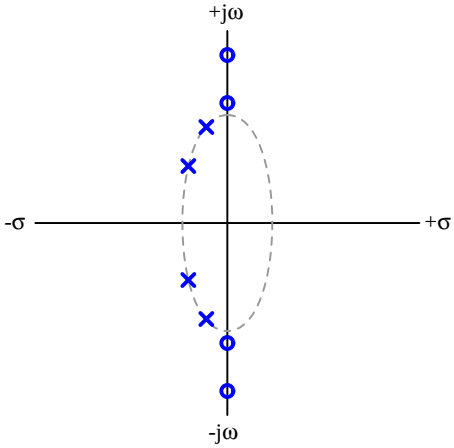
Two Elliptic poles and zeroes



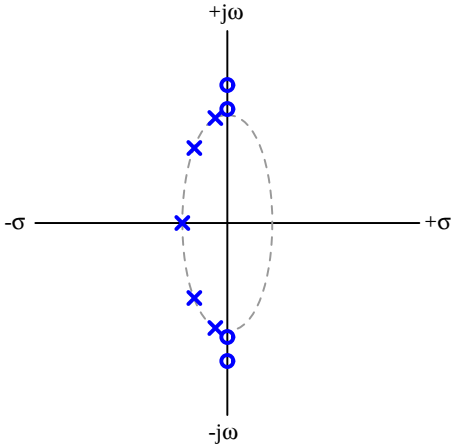
Three Elliptic poles and two zeroes



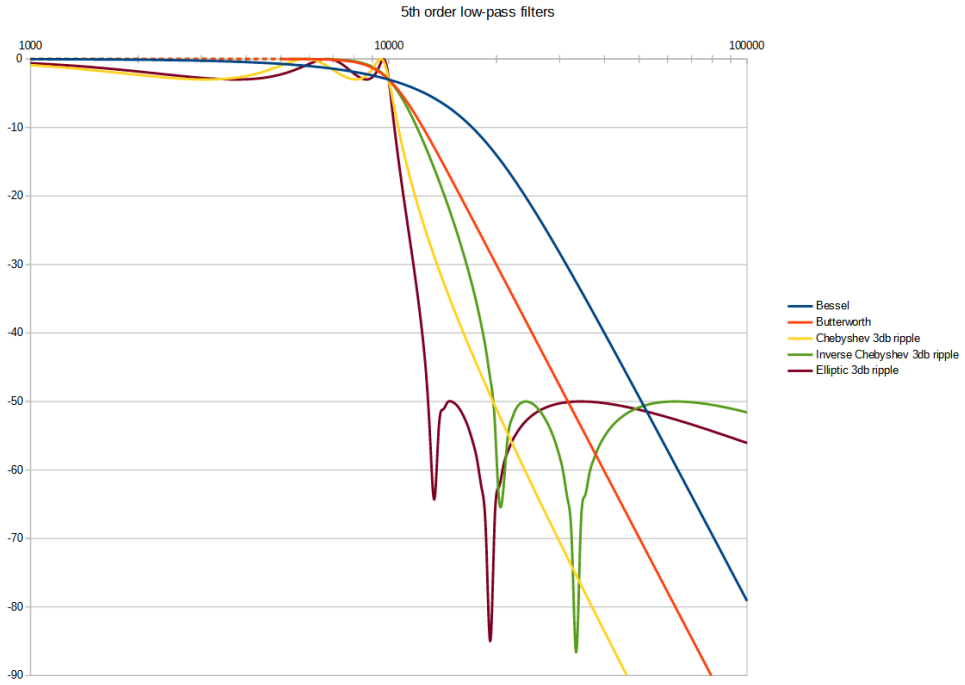
Four Elliptic poles and zeroes



Five Elliptic poles and four zeroes



When we compare the frequency responses of different low-pass filter characteristics, we see the effect of these zeroes in the Elliptic (and also in the Inverse-Chebyshev) Bode plots, revealed as sharp "rippling" in the stopband:

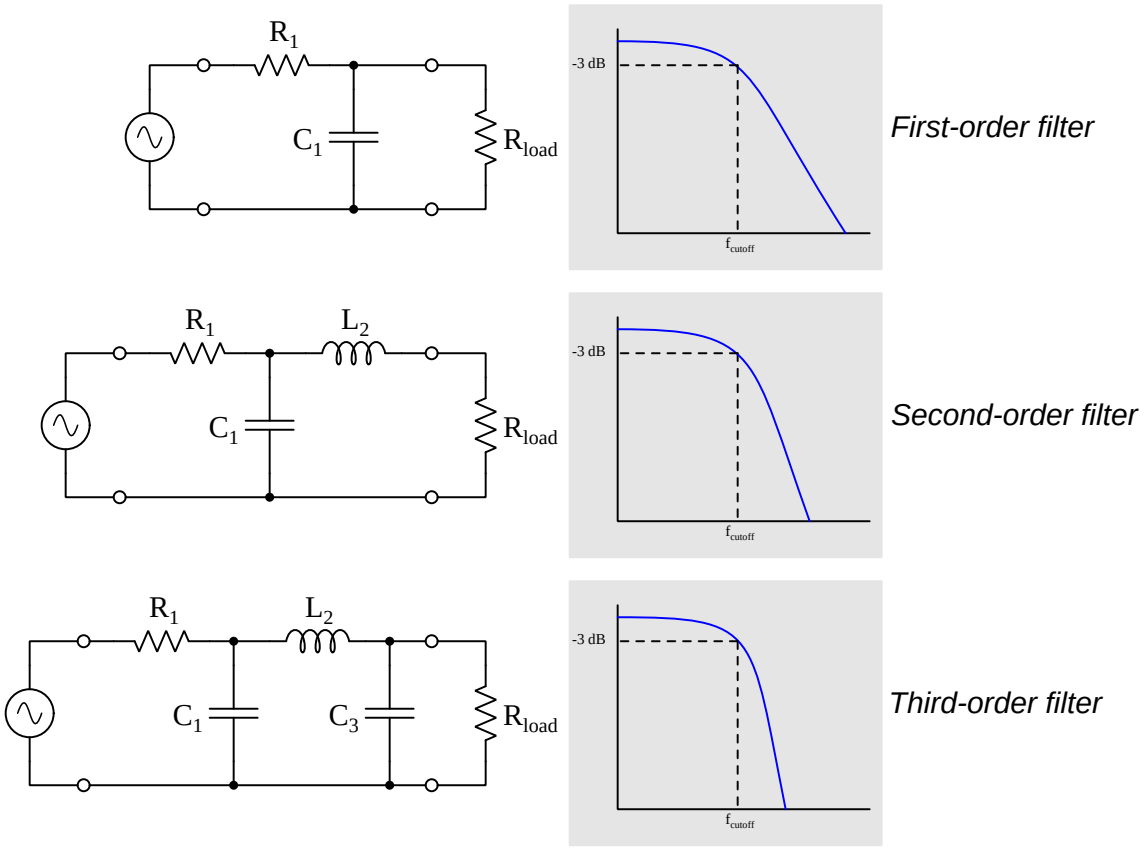


Each of the points along the Bode plot where the graph "dives down" in notch-like fashion is the frequency of one of the filter network's zeroes. Clearly, the presence of the additional zeroes granted by the Elliptic and Inverse Chebyshev filter characteristics is more of an advantage for steep rolloff than it is for consistent attenuation of signals within the stopband.

3.9 Passive filter design

Before the advent of computer software capable of selecting component values necessary for desired filtering characteristics, engineers often relied on standardized tables of component values to accompany standardized filter network topologies. Here we will present similar tables useful for Butterworth-characteristic filters, in order to show how this design strategy works.

The design process generally begins with a low-pass filter topology consisting of parallel capacitors and series inductors, one added for every order of filter desired, with input and load resistances equal to one Ohm ($R_1 = R_{load} = 1\,\Omega$) and a cutoff frequency of one radian per second ($\omega = 1$). Even if we intend to design a high-pass, band-pass, or band-stop filter, and/or with resistance and cutoff values other than 1 (each), the design process begins with this "normalized" low-pass filter which we later modify. Below we see the standard "ladder" progression of added reactive elements assumed in this design:



Below we see a table of capacitor and inductor values for low-pass, Butterworth-characteristic, ladder-topology filter networks up through the 9th order[14]. Each capacitor value is given in the unit of Farads, and each inductor in the unit of Henrys, based on the assumption that the input and load resistances are 1 Ω each and that the cutoff frequency is 1 radian/second:

Order	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9
1	2.000	–	–	–	–	–	–	–	–
2	1.414	1.414	–	–	–	–	–	–	–
3	1.000	2.000	1.000	–	–	–	–	–	–
4	0.765	1.848	1.848	0.765	–	–	–	–	–
5	0.618	1.618	2.000	1.618	0.618	–	–	–	–
6	0.518	1.414	1.932	1.932	1.414	0.518	–	–	–
7	0.445	1.247	1.802	2.000	1.802	1.247	0.445	–	–
8	0.390	1.111	1.663	1.962	1.962	1.663	1.111	0.390	–
9	0.347	1.000	1.532	1.879	2.000	1.879	1.532	1.000	0.347

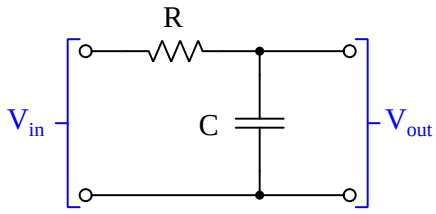
Once we decide on the order number for our filter design, and consult the table for component values, we must *scale* those values to yield practical capacitances and inductances for input and load resistances other than 1 Ω and cutoff frequencies other than 1 radian per second. These "scaling" procedures are fairly easy to perform:

- To scale the table's component values for input and load resistances greater than 1 Ω, we need to increase the impedance of each component accordingly. For example, if we wished to scale our filter design for a 600 Ω load and matching input resistor, every component value would need to be re-sized to yield 600 times as much impedance. The input and load resistances would obviously increase to 600 Ω each, while each capacitor value would *decrease* in value 600-fold and each inductor value would *increase* in value 600-fold. These changes will result in a filter designed for different load and input resistances, but still have the same cutoff frequency as before.

- To scale the table's component values for cutoff frequencies other than 1 rad/sec, we need to maintain the same resistance values as before but adjust the capacitor and inductor values accordingly. Since greater capacitance and greater inductance alike both act to reduce cutoff frequency, increasing cutoff frequency to some reasonable value consists of *decreasing* the values of all capacitors and all inductors by the same factor. For example, if we wished our cutoff frequency to be 2000 radians per second rather than 1 radian per second, all capacitances would have to decrease 2000-fold and all inductances would have to decrease 2000-fold.

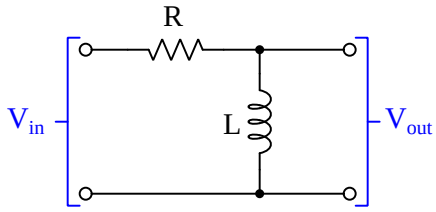
[14] This is by no means an exhaustive table for filter designs. Consulting older filter-design books, it is not uncommon to find tables of normalized values extending out to the 20th order!

For designing a high-pass filter using this table-based method, we must convert the table's figures (which are intended for a low-pass filter network) into figures appropriate for a high-pass filter. This process is easiest to understand on a simple 1st-order filter network such as this:



Cutoff frequency in any simple filter network is defined as the point at which $X = R$. If we assume "normalized" component values of 1 Ω for the resistor and 1 Farad for the capacitor, with a cutoff frequency of 1 radian per second, then the capacitor must exhibit a reactance of 1 Ω as well at this cutoff frequency ($X_C = \frac{1}{\omega C}$).

If we wanted to maintain the resistor's position in this circuit but have the network function as a high-pass filter rather than a low-pass, we could simply replace the capacitor with an inductor:



In order for this high-pass filter to cut off at the same frequency of 1 radian per second as the original low-pass filter, the inductor must exhibit the same 1 Ω reactance at that frequency ($X_L = \omega L$). Knowing that an inductor-capacitor substitution will work to transform the low-pass filter into a high-pass filter if those two components exhibit the same reactance, we may set their reactance values equal to each other and solve for the value of L necessary to replace C:

$$\omega L = \frac{1}{\omega C}$$

$$L = \frac{1}{\omega^2 C}$$

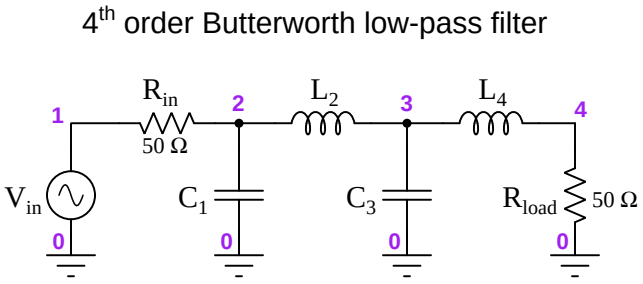
If the filter network we're transforming from low-pass to high-pass has a cutoff frequency of 1 radian per second (i.e. $\omega = 1$), the equivalent inductance equation becomes very simple:

$$L = \frac{1}{C}$$

That is to say, finding inductor values with the same reactance at cutoff as the capacitor values they're replacing when transforming a low-pass filter into a high-pass, it's as simple as reciprocating the original capacitor values. Of course this works for replacing inductors with capacitors as well, for multi-order filters with both L and C elements: just reciprocate the original inductor values to find the replacement capacitor values at a cutoff frequency of 1 radian/second.

3.9.1 Low-pass passive filter design example

Suppose we need a 4th order Butterworth low-pass filter for 50 Ω input and load resistances, and with a 5 MHz cutoff frequency. Our first step is to sketch a schematic diagram for this circuit. Here I will include node numbers in anticipation of using SPICE to simulate its performance:



Then, we will look up normalized component values from the Butterworth parameter table:

Order	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9
4	0.765	1.848	1.848	0.765	–	–	–	–	–

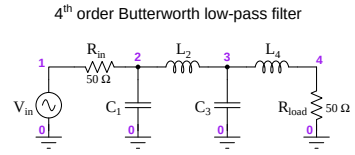
This means for a filter network having a cutoff frequency of 1 radian/second and with input and load resistances of 1 Ω each, C_1 would need to be 0.765 Farads, L_2 would need to be 1.848 Henrys, C_3 would need to be 1.848 Farads, and L_4 would need to be 0.765 Henrys. We know our filter needs to work with 50 Ω at the input and at the load, and so we must scale each of these capacitor and inductor values to have 50 times as much impedance. This means each inductor's value must grow by a factor of 50, and each capacitor's value must shrink by a factor of 50:

- $C_1 = 0.765$ Farads $\div$ 50 = 0.01530 Farads
- $L_2 = 1.848$ Henrys $\times$ 50 = 92.40 Henrys
- $C_3 = 1.848$ Farads $\div$ 50 = 0.03696 Farads
- $L_4 = 0.765$ Henrys $\times$ 50 = 38.25 Henrys

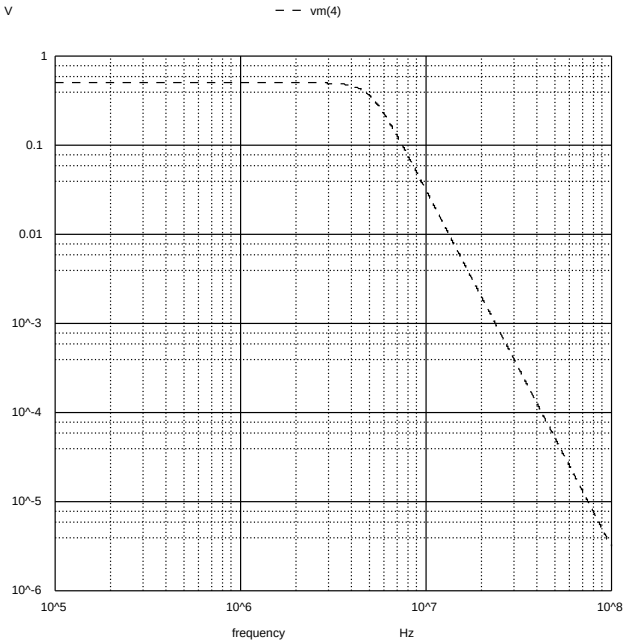
However, we also must scale these values for the desired cutoff frequency of 5 MHz, which is 31.41593 megaradians per second. This is 31.41593×10^6 greater than the normalizing value of 1 radian per second, which means every capacitor and every inductor must be this many times *smaller* than the above figures:

- $C_1 = 0.765$ Farads $\div$ 50 $\div$ (31.41593×10^6) = 0.487 nanoFarads
- $L_2 = 1.848$ Henrys $\times$ 50 $\div$ (31.41593×10^6) = 2.941 microHenrys
- $C_3 = 1.848$ Farads $\div$ 50 $\div$ (31.41593×10^6) = 1.176 nanoFarads
- $L_4 = 0.765$ Henrys $\times$ 50 $\div$ (31.41593×10^6) = 1.218 microHenrys

When simulated using SPICE, we see a very nice low-pass response that cuts off at 5 MHz.

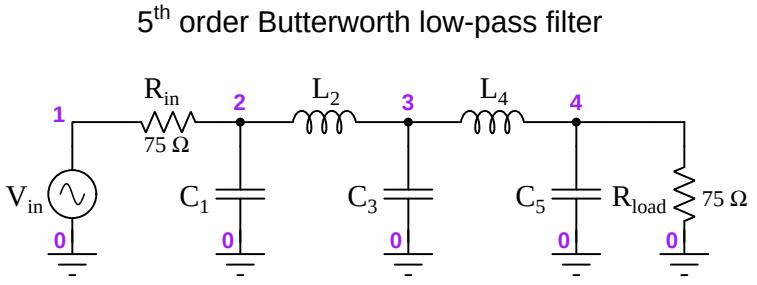



```
* 4th order Butterworth low-pass filter
vin 1 0 ac 1
rin 1 2 50
c1 2 0 0.487e-9
l2 2 3 2.941e-6
c3 3 0 1.176e-9
l4 3 4 1.218e-6
rload 4 0 50
.ac dec 1e3 1e5 1e8
.plot ac loglog vm(4)
.end
```



3.9.2 High-pass passive filter design example

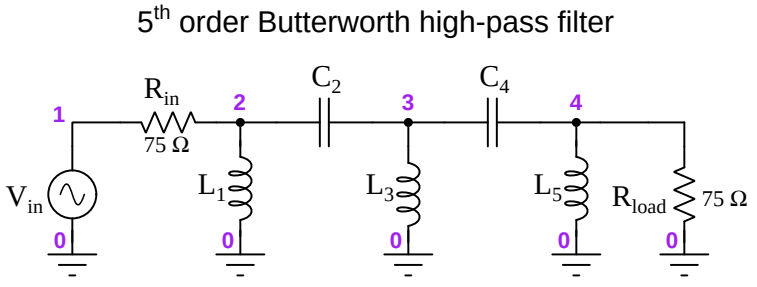
Suppose we need a 5th order Butterworth high-pass filter for 75 Ω input and load resistances, and with a 300 kHz cutoff frequency. Our first step is to sketch a schematic diagram for a *low-pass* filter circuit, because our published table of Butterworth component values assumes low-pass. Here I will include node numbers in anticipation of using SPICE to simulate its performance:



Then, we will look up normalized component values from the Butterworth parameter table:

Order	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9
5	0.618	1.618	2.000	1.618	0.618	–	–	–	–

This means for a low-pass filter network having a cutoff frequency of 1 radian/second and with input and load resistances of 1 Ω each, C_1 would need to be 0.618 Farads, L_2 would need to be 1.618 Henrys, C_3 would need to be 2 Farads, L_4 would need to be 1.618 Henrys, and C_5 would need to be 0.618 Farads. However, what we are trying to design is a *high-pass* filter, not a low-pass filter. To convert from low-pass to high-pass, we must replace each reactive component with its opposite type, and find new component values based on the reciprocals[15] of the old values. In other words, every C_n becomes $\frac{1}{L_n}$, and every L_n becomes $\frac{1}{C_n}$:



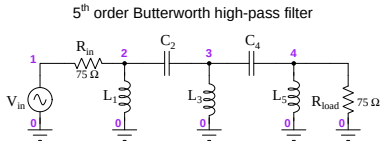
This means for our high-pass filter network with a cutoff frequency of 1 radian/second and input and output resistances of 1 Ω each, L_1 would need to be 1.618 Henrys, C_2 would need to be 0.618 Farads, L_3 would need to be 0.5 Henrys, C_4 would need to be 0.618 Farads, and L_5 would need to be 1.618 Henrys.

[15] Remember that this only works when the cutoff frequency is 1 radian per second, which is the assumed cutoff frequency for all values given in the table.

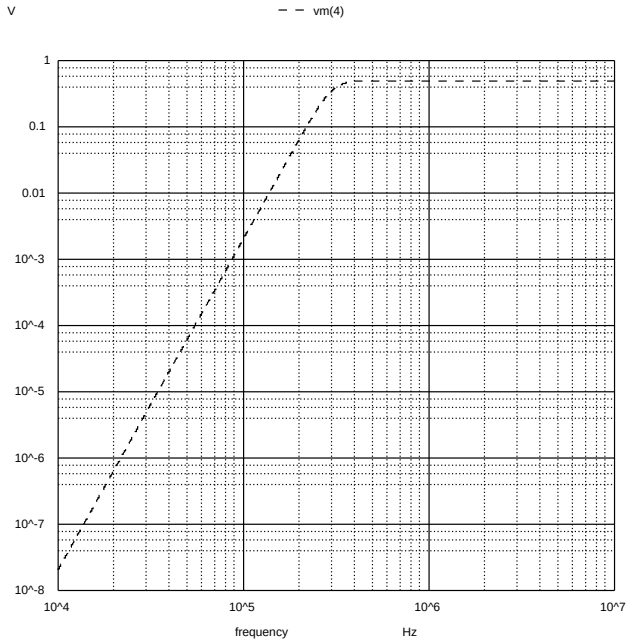
Next comes the impedance scaling and cutoff frequency scaling. We know our filter needs to work with 75 Ω at the input and at the load, and so we must scale each of these capacitor and inductor values to have 75 times as much impedance. This means each inductor's value must grow by a factor of 75, and each capacitor's value must shrink by a factor of 75. We also know our cutoff frequency needs to be 300 kHz, which is 1.884956 megaradians per second (i.e. 1.884956×10^6) times more than it is now). This means decreasing each inductance and each capacitance by a factor of 1.884956×10^6. Performing both scaling operations in one step, for each component:

- $L_1 = 1.618$ Henrys $\times\ 75 \div (1.884956 \times 10^6) = 64.38$ microHenrys
- $C_2 = 0.618$ Farads $\div\ 75 \div (1.884956 \times 10^6) = 4.371$ nanoFarads
- $L_3 = 0.5$ Henrys $\times\ 75 \div (1.884956 \times 10^6) = 19.89$ microHenrys
- $C_4 = 0.618$ Farads $\div\ 75 \div (1.884956 \times 10^6) = 4.371$ nanoFarads
- $L_5 = 1.618$ Henrys $\times\ 75 \div (1.884956 \times 10^6) = 64.38$ microHenrys

When simulated using SPICE, we see a very nice high-pass response that cuts off at 300 kHz.




```
* 5th order Butterworth high-pass filter
vin 1 0 ac 1
rin 1 2 75
l1 2 0 64.38e-6
c2 2 3 4.371e-9
l3 3 0 19.89e-6
c4 3 4 4.371e-9
l5 4 0 64.38e-6
rload 4 0 75
.ac dec 1e3 1e4 1e7
.plot ac loglog vm(4)
.end
```



3.10 Active filters

Filter networks incorporating some form of signal amplification are called *active* filters, in contrast to *passive* filters which only use passive components such as capacitors, inductors, and/or resistors. Advantages enjoyed by active filters include the following:

- The ability to generate a stronger output signal than the input
- Consistent performance even when load impedance changes
- The ability to easily "cascade" filter networks to create higher-order filtering
- With active filters we are able to use capacitors as the only reactive elements, and need not use any inductors
- The ability to easily adjust the quality factor (Q) of the filter

The first advantage of active filters over passive filters is rather obvious: passive filters cannot amplify, and therefore can only attenuate the signal.

The second advantage is related to the fact that in a passive filter the load impedance (usually modeled as a resistor) actually affects certain filtering characteristics such as cutoff frequency, because all electrical power delivered to the load must pass through passive components lying between the input and output terminals. In an active filter, however, the output signal comes from the output terminal of an amplifier which means the amplifier is what sources all power to the load, and in so doing the filtering elements (resistors and capacitors, usually) prior to the amplifier do not experience any impact from the load. In effect, the amplifier "shields" the filtering elements from the load.

The third advantage is a direct extension of the second: if the load impedance does not affect the behavior of the filter network, then it becomes quite easy to connect the output of one filter network to the input of another to create larger filter networks with higher-order responses. By contrast, with passive filter network design we must re-compute all prior component values whenever we add more "orders" to the network precisely because those additional orders constitute a "load" to the preceding order(s). Thus, active filters enable a more "modular" design approach where we may combine filter networks without concern for one filter stage affecting the performance of the other.

The fourth advantage is an advantage only because inductors tend to be poorly-behaved components compared to capacitors. In theory the fundamental behaviors of inductors and capacitors are beautifully symmetrical to one another: the voltage across an inductor proportional to its rate-of-change of current over time ($V = L\frac{dI}{dt}$) and the current through a capacitor proportional to its rate-of-change of voltage over time ($I = C\frac{dV}{dt}$). In practice, though, inductors exhibit far greater levels of parasitic properties than capacitors. Inductors have substantial amounts of parasitic resistance (from wire length and energy dissipation in the core), substantial amounts of parasitic capacitance (from turn-to-turn separation of the insulated wire), as well as substantial nonlinearities due to the B-H curve of the ferromagnetic core material. Lastly, inductors are more difficult to shield than capacitors due to the inherently circuitous shape of magnetic fields, as opposed to electric fields which may be made to terminate at a conducting surface. To be fair, capacitors are not exempt from parasitic effects (e.g. equivalent series resistance (ESR) from dielectric energy dissipation effects, parasitic inductance from lead length, and nonlinearities especially with certain ceramic dielectric materials), but in general the degree of each of these effects is much less than what we find with

typical inductors. Since all of these parasitic effects tend to degrade the performance of a filter network from its ideal response, the ability to exclude inductors from our filter design is good.

The fifth advantage – adjustable quality factor (Q) – comes from the ability of amplifier-based filters to offer some degree of positive feedback which is impossible with passive filters. In fact, with enough positive feedback in an active filter *oscillations* become possible!

With all the advantages offered by active filter designs, one might wonder why passive filters are used at all. Several applications demand passive filter networks rather than active:

- High-reliability systems
- Power source filtering
- High-frequency (RF) systems

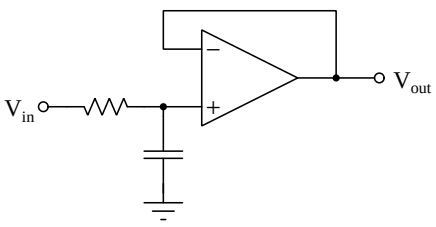
Passive components tend to be more rugged and reliable than active components, all other factors being equal. For example, passive components generally tolerate electrostatic discharge, over-temperature, and transient over-voltage conditions better than most semiconductors. In systems where exceptionally high reliability and long life is essential, and either a passive or active filter network will suffice, a designer may opt for passive based solely on superior reliability.

Filter networks used to condition electrical power from sources to loads must absolutely be passive in nature. The reason for this is that active filters do not transfer power from their inputs to their outputs like passive filters, but rather replicate the (filtered) input signal by drawing from a separate DC power source. It would be pointless to use an active low-pass filter to condition the output of a power source both because the power source's output could not actually drive any power of its own through the active filter (thus making that source moot) and the active filter would require a DC power source of its own to drive any power to a load.

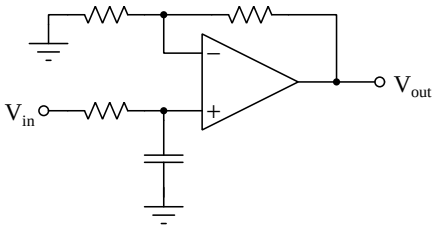
An active filter is able to work only if the bandwidth capability of its amplifier(s) is significantly greater than the bandwidth of the signal being filtered. In radio-frequency (RF) systems where signal frequencies are typically in the megaHertz and gigaHertz range – beyond the capabilities of most operational amplifiers – passive filters are the only practical option.

Several active low-pass filter topologies appear in the following schematic diagrams, the amplifier in each case being an *operational amplifier* (i.e. "opamp"):

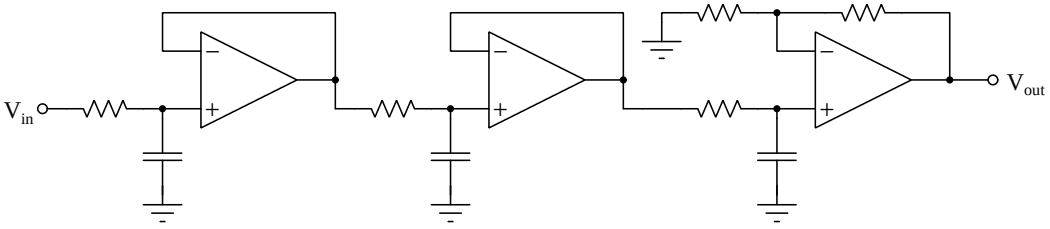
Buffered low-pass filter



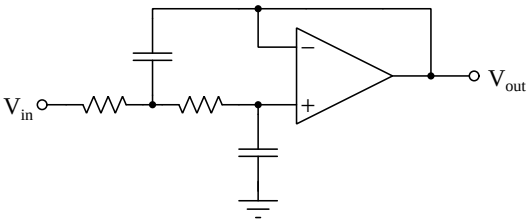
Boosted low-pass filter



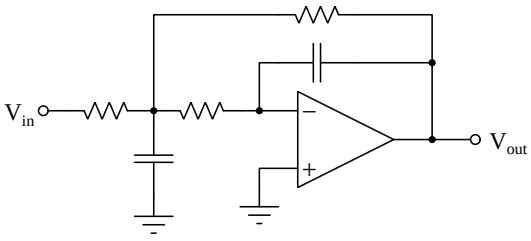
Cascaded low-pass filter



Sallen-Key low-pass filter

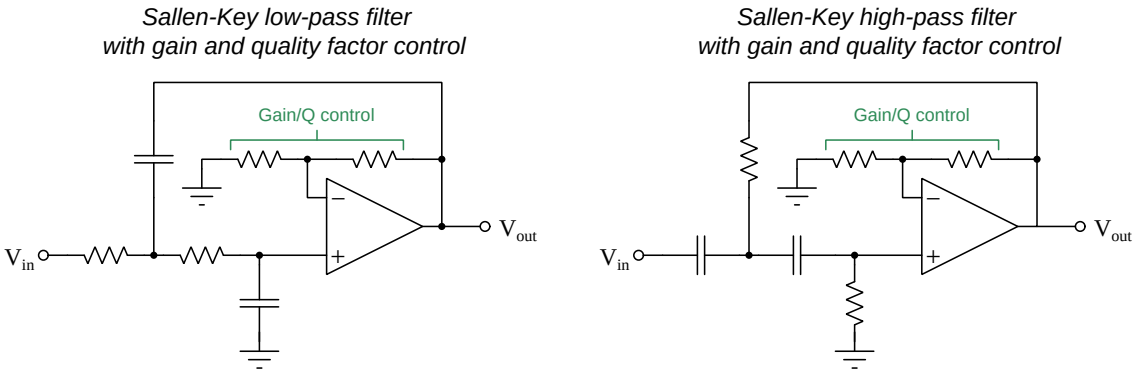


Multiple feedback (MFB) low-pass filter



The upper two active filters shown here are first-order, but the rest are multiple-order (third-order for the cascaded and second-order for the Sallen-Key and MFB designs). In each case the low-pass filter network could be made into a high-pass filter simply by swapping the positions of the filtering resistors and capacitors.

We may realize the advantage of variable quality factor (Q) if we exploit the opamp's ability to provide voltage gains greater than one in feedback-based amplifier topologies such as the Sallen-Key. Note how voltage-divider feedback is used in each of these Sallen-Key active filters to boost the output voltage of the opamp which enhances the feedback to the RC input network:



To understand why amplifier voltage gain affects quality factor in an active filter, it is helpful to remember how we define quality factor (Q) in elementary passive networks. For a passive LC network, the amount of electrical resistance in that network is the primary factor influencing Q: the more resistance, the lower the quality factor, all other factors being equal. If we were to somehow construct an LC network with no resistance whatsoever – including zero ESR in the capacitor, zero magnetic core losses in the inductor, and no other forms of energy dissipation at all – that LC network would exhibit an infinite Q value and thus could sustain oscillations indefinitely. The Second Law of Thermodynamics[16] expressed in the dissipative properties of real circuits conspires to keep Q finite for passive LC networks. However, the ability of active components such as operational amplifiers to infuse energy from an external source into a circuit with passive components makes it possible to overcome the energy losses of those passive components, thereby increasing Q to the point where active filters are able to actually sustain oscillations (with sufficient network gain).

Expressing this concept in terms of the s variable and pole-zero plots, increased gain moves the poles of the network farther to the right, toward the vertical axis. This means smaller negative σ values, meaning slower decay rates for the system's natural signal values. If pushed far enough by adding more amplifier gain, the network's poles may be moved to the right until they lie along the vertical axis itself (i.e. $\sigma = 0$) where neither growth nor decay occurs and self-sustaining oscillations become possible. If even more gain is added to the system, the poles will shift to the right-hand side of the s-plane meaning the system will become unstable and oscillate with ever-increasing amplitude until the amplifier clips and distorts the signal's wave-shape. Again, such behaviors are flatly impossible for passive filter networks, but within the realm of possibility for active filters.

When active filters are cascaded together, their orders simply add. The three first-order filters connected as shown above create a third-order filter. Two Sallen-Key or MFB filters cascaded together would create a fourth-order filter. As with passive filter networks, active filter component selection is a complicated task made easier through the use of filter simulation software.

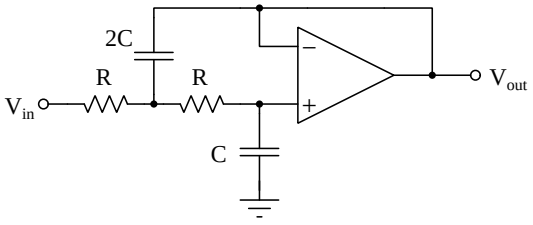
[16] The natural tendency of energy to spread and become more diffuse, usually by conversion of non-thermal energy into *heat*.

3.11 Active filter design

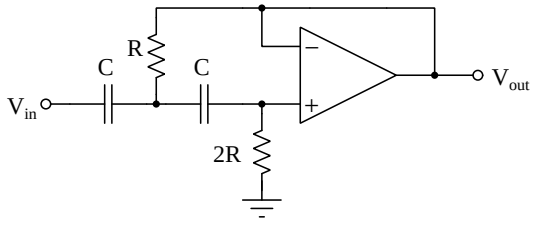
Although active filter design is every bit as mathematically complex as passive filter design – a task worthy of simulation software – there do exist certain "recipes" for selecting component values necessary for desired filtering characteristics. Here we will explore component values for Butterworth-characteristic Sallen-Key topology active filters.

The basic Sallen-Key filter topology will have a Butterworth characteristic when the feedback component's impedance is half that of the grounding component's impedance at any given frequency. For the low-pass version this means a feedback capacitor with twice the capacitance of the grounding capacitor; for the high-pass this means a grounding resistor with twice the resistance of the feedback resistor:

Low-pass with Butterworth response



High-pass with Butterworth response



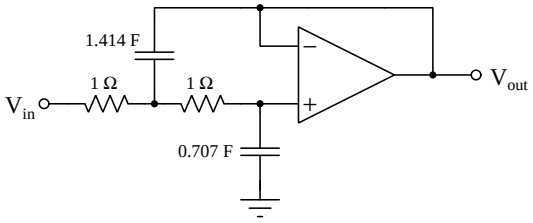
Butterworth response is very desirable for its "maximally flat" passband response in applications such as data acquisition where we wish to attenuate the measured signal as little as possible for frequencies within the passband range. A good example of such a filter application would be an *anti-aliasing* filter placed before the input of an analog-to-digital converter, to prevent frequencies beyond the ADC's sampling limit from arriving full-strength at the ADC's input. A filter achieving a more aggressive roll-off at the expense of "ripple" in the passband such as a Chebyshev or Elliptic response would unfortunately attenuate some of the desired (passband) frequencies more than others, possibly leading to inaccurate signal voltage measurements. Here, the Butterworth response is ideal for passing all desired frequencies at (or nearly at) full-strength while significantly attenuating only the undesired frequencies lying in the stopband.

Cutoff frequency for any[17] low-pass or high-pass Sallen-Key filter is given by the following formulae:

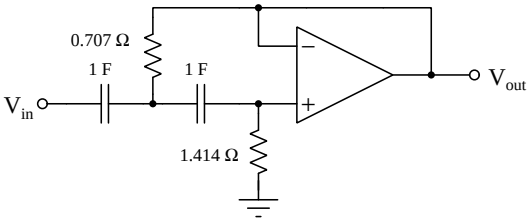
$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \qquad \omega_c = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}}$$

In order to have a Sallen-Key filter exhibit a Butterworth response, the 2:1 impedance ratio of the grounding versus feedback impedances must be maintained. Therefore, Sallen-Key low-pass and high-pass Butterworth filter networks with component values normalized to 1 Ohm at a cutoff frequency of 1 radian per second will have the following values:

Low-pass with Butterworth response and ω_c = 1 radian/sec



High-pass with Butterworth response and ω_c = 1 radian/sec

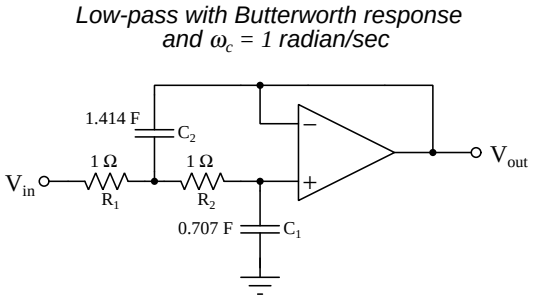


As usual, when designing a Sallen-Key active filter for any particular application, we would need to *scale* the component values for both desired input impedance and for desired cutoff frequency. For the Sallen-Key filter topology, output impedance is largely a function of the opamp's ability to source and sink load current, while input impedance is a rather complicated function of component values and frequency.

[17] This cutoff frequency formula holds true regardless of whether or not the component values are chosen for a Butterworth response. Achieving Butterworth, Bessel, Chebyshev, Elliptic, or other response is a matter of these four components' particular ratios to each other, but in any case the cutoff frequency will be inversely proportional to the square-root of the product of all four values.

3.11.1 Low-pass active filter design example

Suppose we need a 2nd order Butterworth low-pass filter with a 30 kHz cutoff frequency and using 10 kΩ resistors, using the Sallen-Key topology. We know that the basic format for 2nd-order Butterworth-response Sallen-Key low-pass active filters normalized to 1 Ω resistances and 1 radian per second cutoff follow this prototype design:



We will need to scale both the component impedances as well as the cutoff frequency from the default (normalized) values.

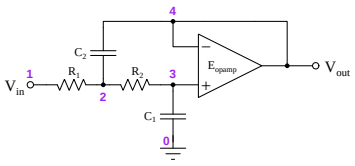
First, scaling the impedances. To go from a default resistor value of 1 Ω to the desired resistor value 10 kΩ is of course a 10,000-fold magnification. This means making each resistor 10,000 times larger in resistance and each capacitor 10,000 smaller in capacitance in order to scale up impedances while not affecting cutoff frequency:

- R_1 = 1 Ohm × 10,000 = 10 kiloOhms
- R_2 = 1 Ohm × 10,000 = 10 kiloOhms
- C_1 = 0.707 Farads ÷ 10,000 = 70.7 microFarads
- C_2 = 1.414 Farads ÷ 10,000 = 141.4 microFarads

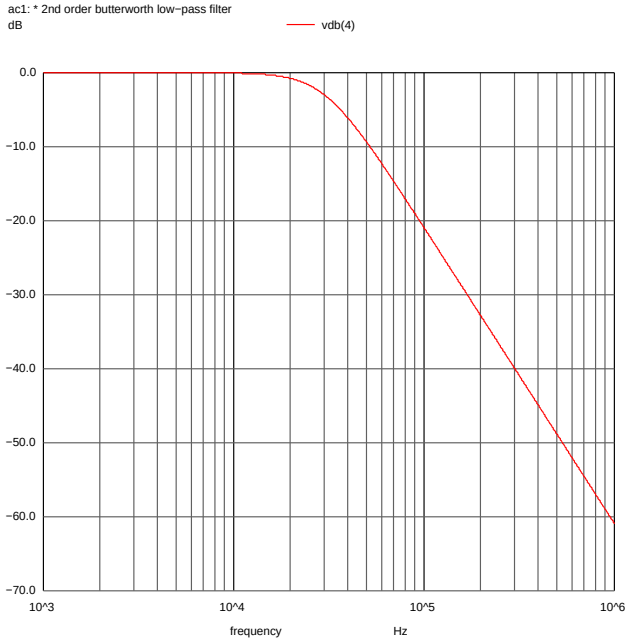
Next, scaling the cutoff frequency. The normalized value of 1 radian per second ($\frac{1}{2\pi}$ Hertz) scaled up to 30 kiloHertz represents a scaling factor of 188,495.6. This means every reactive component in the circuit must be scaled to have the same reactance as it did before at 1 radian/second, but now at 188,495.6 times that frequency. Since this active filter contains no inductors, we need only concern ourselves with scaling the capacitor values down by this factor, while the resistors remain at 10 kΩ each because their impedances do not vary with frequency:

- R_1 = 1 Ohm × 10,000 = 10 kiloOhms
- R_2 = 1 Ohm × 10,000 = 10 kiloOhms
- C_1 = 0.707 Farads ÷ 10,000 ÷ 188,495.6 = 0.375 microFarads
- C_2 = 1.414 Farads ÷ 10,000 ÷ 188,495.6 = 0.750 nanoFarads

When simulated using SPICE[18], we see a very nice low-pass response that cuts off at 30 kHz and exhibits a roll-off of −40 dB per decade:




```
* 2nd order Butterworth low-pass filter
vin 1 0 ac 1
r1 1 2 10e3
r2 2 3 10e3
c1 3 0 0.375e-9
c2 2 4 0.75e-9
e_opamp 4 0 3 4 9e6
.ac dec 1000 1e3 1e6
.plot ac vdb(4)
.end
```



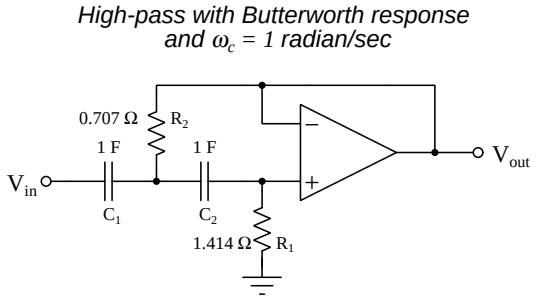
While the calculated resistor and capacitor values certainly deliver the desired filtering performance, the capacitor values are definitely not standard according to the common E12 series of values specified in IEC standard 60063. Non-standard resistances are less of a design and manufacturing problem because it's easy to replace fixed resistors with adjustable potentiometers and be able to calibrate the filter's response thusly. Since variable capacitors are generally not an option but variable resistors are, it might behoove us to re-scale these component values to make the capacitances IEC-60063 standard and let the resistances be non-standard. To re-scale these values, we would need to adjust the capacitances and resistances in opposite directions so as to leave the cutoff frequency fixed.

For example, we could choose an E12-series capacitance of 0.39 μF for C_2 instead of 0.375 μF, and then use two of these standard-value 0.39 μF capacitors connected in parallel to make C_1. This represents a re-scaling factor of 1.03979 from the non-standard C_2 value to the standard C_2 value. If we re-scale all capacitance values upward by this new factor, we will have to re-scale all resistances downward by the same in order to preserve our intended cutoff frequency (i.e. maintain the same $R_1 R_2 C_1 C_2$ product value), making R_1 and R_2 both 9.61731 kΩ rather than 10 kΩ. When we re-simulate the Sallen-Key low-pass filter with these new component values to verify our re-scaling work, we find it still cuts off at 30 kHz.

[18] Note the use of a controlled voltage source (`e_opamp`) with an open-loop gain of nine million to simulate the operational amplifier, as well as decibel plotting of node 4 voltage (`vdb(4)`).

3.11.2 High-pass active filter design example

Suppose we need a 2nd order Butterworth high-pass filter with cutoff frequency of 5 kHz and resistances in the general range of 10 kΩ, with the Sallen-Key topology. We know that the basic format for 2nd-order Butterworth-response Sallen-Key high-pass active filters normalized to 1 F capacitances and 1 radian per second cutoff follow this prototype design:



We will need to scale the component impedances as well as the cutoff frequency from the default (normalized) values.

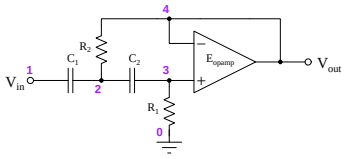
First, scaling the impedance. If we want the resistor values to be closer to 10 kΩ rather than 0.707 Ω and 1.414 Ω, respectively, we may simply multiply each of those normalized values by 10,000. This also requires making each capacitor 10,000 times smaller than the normalized values shown in order to scale up impedances while not affecting cutoff frequency:

- C_1 = 1 Farad ÷ 10,000 = 100 microFarads
- C_2 = 1 Farad ÷ 10,000 = 100 microFarads
- R_1 = 1.414 Ohms × 10,000 = 14.14 kiloOhms
- R_2 = 0.707 Ohms × 10,000 = 7.07 kiloOhms

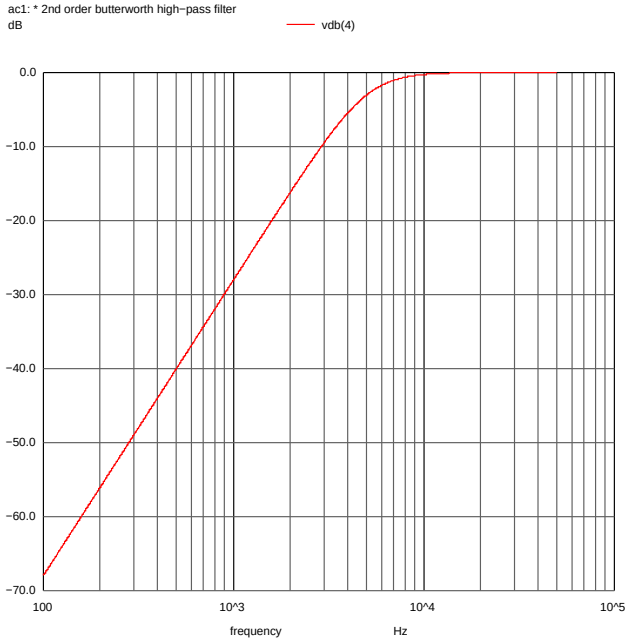
Next, scaling the cutoff frequency. The normalized value of 1 radian per second ($\frac{1}{2\pi}$ Hertz) scaled up to 5 kiloHertz represents a scaling factor of 31,415.93. This means every reactive component in the circuit must be scaled to have the same reactance as it did before at 1 radian/second, but now at 31,415.93 times that frequency. Since this active filter contains no inductors, we need only concern ourselves with scaling the capacitor values down by this factor, while the resistors remain as they were because their impedances do not vary with frequency:

- C_1 = 1 Farad ÷ 10,000 ÷ 31,415.93 = 3.183 nanoFarads
- C_2 = 1 Farad ÷ 10,000 ÷ 31,415.93 = 3.183 nanoFarads
- R_1 = 1.414 Ohms × 10,000 = 14.14 kiloOhms
- R_2 = 0.707 Ohms × 10,000 = 7.07 kiloOhms

When simulated using SPICE[19], we see a very nice high-pass response that cuts off at 5 kHz and exhibits a roll-off of −40 dB per decade:




```
* 2nd order Butterworth high-pass filter
vin 1 0 ac 1
c1 1 2 3.183e-9
c2 2 3 3.183e-9
r1 3 0 14.14e3
r2 2 4 7.07e3
e_opamp 4 0 3 4 9e6
.ac dec 1000 1e2 50e3
.plot ac vdb(4)
.end
```



While the calculated resistor and capacitor values certainly deliver the desired filtering performance, none of the component values are standard according to the common E12 series of values specified in IEC standard 60063. Non-standard resistances are less of a design and manufacturing problem because it's easy to replace fixed resistors with adjustable potentiometers and be able to calibrate the filter's response thusly. Since variable capacitors are generally not an option but variable resistors are, it might behoove us to re-scale these component values to make the capacitances IEC-60063 standard and let the resistances be non-standard. To re-scale these values, we would need to adjust the capacitances and resistances in opposite directions so as to leave the cutoff frequency fixed.

For example, we could choose an E12-series capacitance of 3.3 nF for C_1 and for C_2 instead of 3.183 nF. This represents a re-scaling factor of 1.03673 from each non-standard capacitance value to each standard capacitance value. If we re-scale all capacitance values upward by this new factor, we will have to re-scale all resistances downward by the same in order to preserve our intended cutoff frequency (i.e. maintain the same $R_1 R_2 C_1 C_2$ product value), making $R_1 = 6.8196$ kΩ and $R_2 = 13.639$ kΩ. When we re-simulate the Sallen-Key high-pass filter with these new component values to verify our re-scaling work, we find it still cuts off at 5 kHz.

[19] Note the use of a controlled voltage source (`e_opamp`) with an open-loop gain of nine million to simulate the operational amplifier, as well as decibel plotting of node 4 voltage (`vdb(4)`).

3.12 Active filter transfer functions

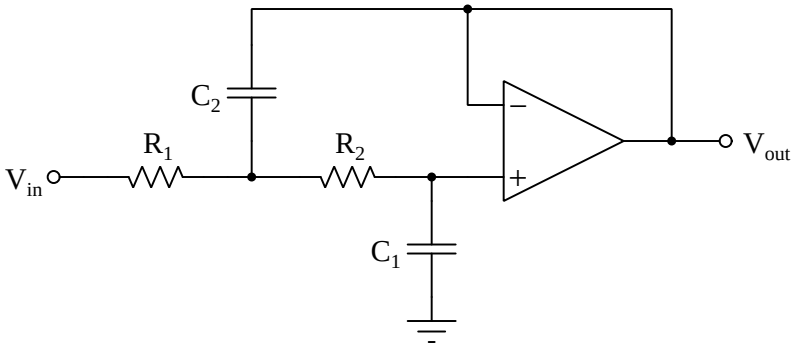
Transfer functions are mathematical expressions of a system's *gain* – i.e. the ratio between output and input – in terms of complex frequency (s). That s variable itself expresses frequency in real and imaginary parts, the real part (σ) representing growth or decay rate in time constants per second (also known as the *Neper frequency*), and the imaginary part ($j\omega$) representing oscillatory rate in radians per second (also known as *angular velocity* or *natural frequency*), such that $s = \sigma + j\omega$.

Transfer functions describing electrical networks usually consist of gain ratios of output voltage to input voltage, e.g. $\frac{V_{out}}{V_{in}}$, and their derivations usually involve formulae containing impedance values of the network's components: R for resistance, $Z_L = sL$ for inductive impedance, and $Z_C = \frac{1}{sC}$ for capacitive impedance. Once a transfer function is derived with s as the independent variable, any value of s resulting in a transfer function value of zero represents a "zero" for the network describing a frequency condition where the output voltage of the filter is zero even with non-zero input voltage, and any value of s resulting in a transfer function value of infinity represents a "pole" for the network describing a frequency condition where the network outputs finite voltage while having zero input voltage. Poles and zeroes help quantify the response of any system to a wide range of stimuli, and for filter networks in particular they help explain Bode plot features such as cutoff frequency(ies), passband ripple, stopband passband, rolloff rates, etc.

In the following subsections we will explore how transfer functions may be derived for various active filter circuits. One feature of these derivations is a firm reliance on fundamental electrical principles such as Ohm's Law, Kirchhoff's Voltage Law, Kirchhoff's Current Law, and of course the s variable and s-based expressions of impedance.

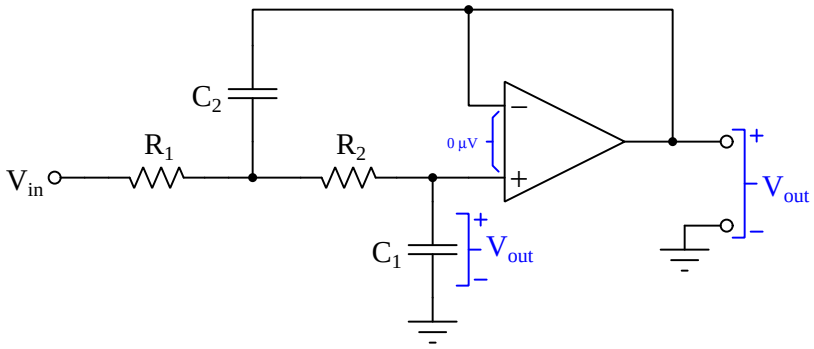
3.12.1 Sallen-Key low-pass filter transfer function derivation

Let's begin with a simple Sallen-Key low-pass filter network, deriving its transfer function as a ratio of output voltage to input voltage:



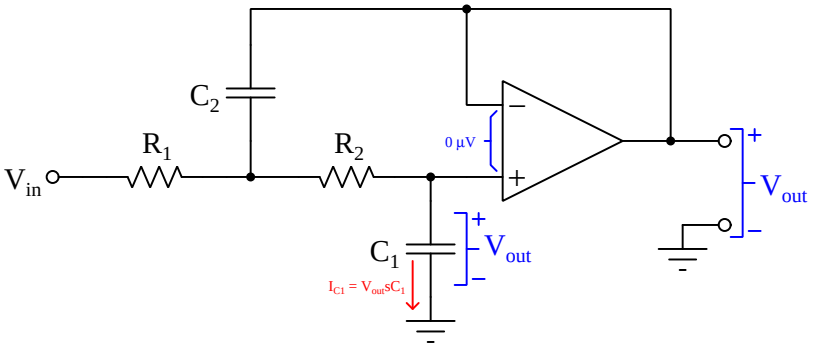
Interestingly, an efficient way to begin analyzing this circuit is to start at its *output*, which we will simply label V_{out} (measured between the output terminal and ground). This is an effective strategy because it results in the formula containing V_{out} as an independent variable from which we may calculate V_{in} as a dependent variable, and once we have a formula containing both of these variables it is a simple matter to manipulate it into a transfer function with $\frac{V_{out}}{V_{in}}$ on one side of the equation and all the s-based terms on the other.

Following the Cardinal Rule of Negative Feedback we know the opamp will regulate its output terminal potential to always maintain its two input terminals in a state of equipotentiality. Therefore, capacitor C_1 must drop the same voltage as V_{out}. In order to establish polarity references for future calculations we will arbitrarily choose V_{out} to be positive with respect to ground:

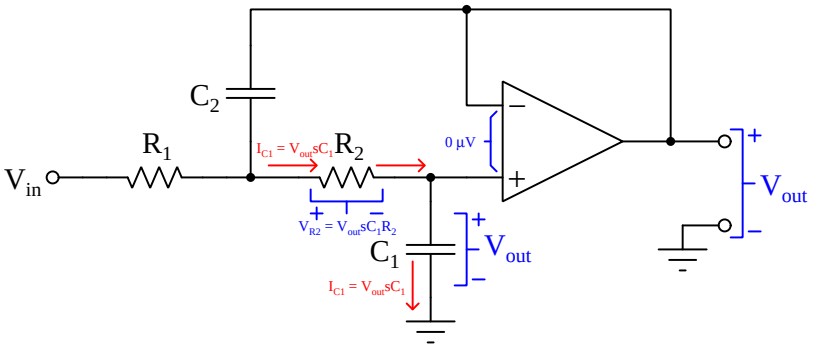


The fact that signal voltages in filter networks are AC and not DC does not make + and − symbols illegitimate. These symbols are still necessary in order to properly relate the phases of voltages and currents to each other in the same network. If it helps, just imagine the + and − symbols representing actual voltage polarity at one particular instant in time.

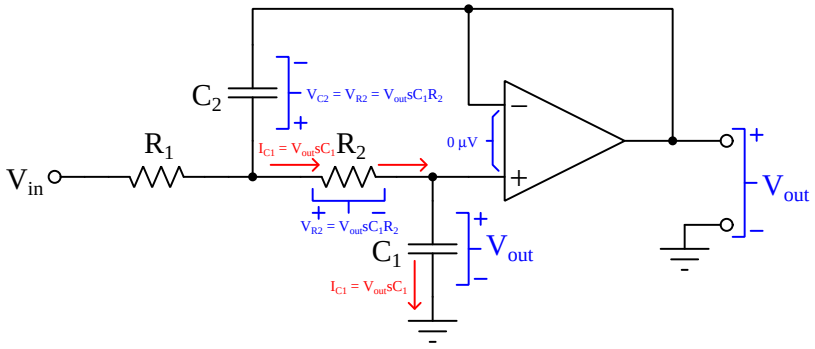
The amount of current through capacitor C_1 may be determined by applying Ohm's Law, dividing the applied voltage (V_{out}) by the capacitor's impedance ($Z_{C1} = \frac{1}{sC_1}$). Therefore, $I_{C1} = V_{out}sC_1$:



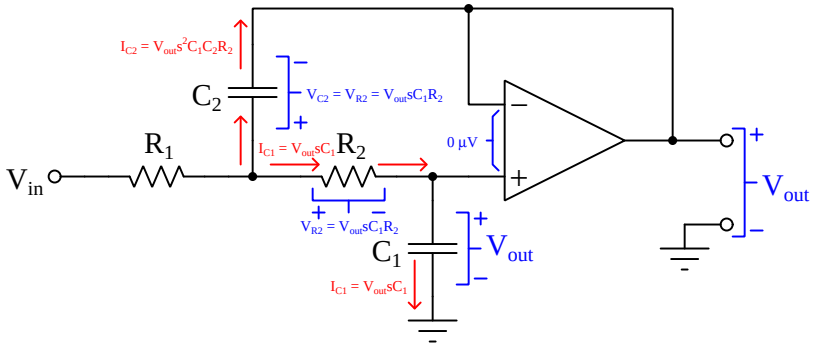
Since opamp input terminals pass negligible current, by Kirchhoff's Current Law we may conclude all of C_1's current must also pass through resistor R_2. This produces a voltage drop across R_2 equal to $V_{out}sC_1R_2$ as annotated below:



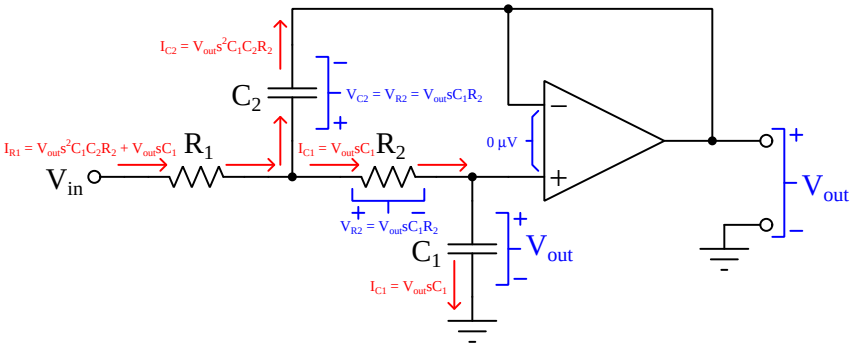
If we now apply Kirchhoff's Voltage Law to R_2, C_2, and the zero voltage between the opamp's input terminals (from the Cardinal Rule of Negative Feedback), we must conclude capacitor C_2 drops this same voltage as well:



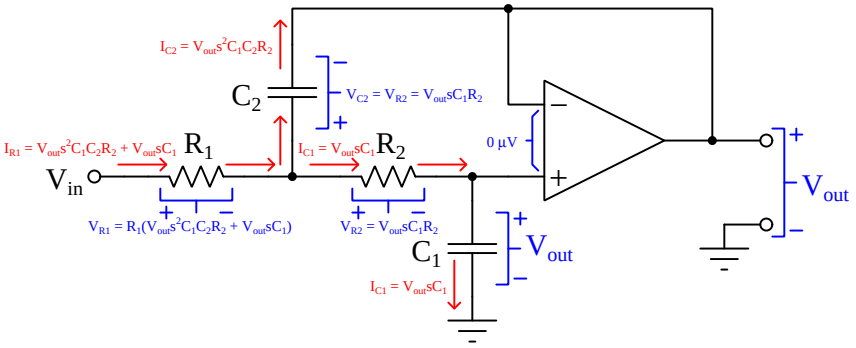
Current through C_2 may also be determined using Ohm's Law just as we did for current through C_1, by dividing the applied voltage ($V_{out}sC_1R_2$) by the impedance of C_2 ($Z_{C2} = \frac{1}{sC_2}$), the result being $I_{C2} = V_{out}s^2C_1C_2R_2$:



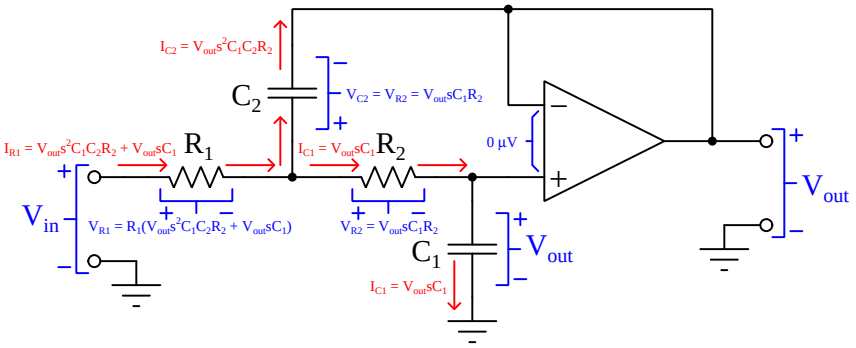
Applying Kirchhoff's Current Law to the node between R_1, R_2, and C_2, we conclude the current through R_1 must be the sum of the currents through C_2 and R_2, which is $I_{R1} = V_{out}s^2C_1C_2R_2 + V_{out}sC_1$:



By Ohm's Law, this drops a voltage across R_1 equal to $V_{R1} = R_1(V_{out}s^2C_1C_2R_2 + V_{out}sC_1)$:



The necessary applied voltage between the input terminal and ground must by Kirchhoff's Voltage Law be the sum $V_{R1} + V_{R2} + V_{C1}$, or $V_{in} = R_1(V_{out}s^2C_1C_2R_2 + V_{out}sC_1) + V_{out}sC_1R_2 + V_{out}$:



From here we may use algebra to arrive at a transfer function defined as $\frac{V_{out}}{V_{in}}$:

$$V_{in} = R_1(V_{out}s^2C_1C_2R_2 + V_{out}sC_1) + V_{out}sC_1R_2 + V_{out}$$

$$V_{in} = V_{out}s^2C_1C_2R_1R_2 + V_{out}sC_1R_1 + V_{out}sC_1R_2 + V_{out}$$

$$V_{in} = V_{out}(s^2C_1C_2R_1R_2 + sC_1R_1 + sC_1R_2 + 1)$$

$$\frac{V_{in}}{V_{out}} = s^2C_1C_2R_1R_2 + sC_1R_1 + sC_1R_2 + 1$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{s^2C_1C_2R_1R_2 + sC_1R_1 + sC_1R_2 + 1}$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{s^2C_1C_2R_1R_2 + sC_1(R_1 + R_2) + 1}$$

With the numerator having a constant value of one, there are no finite "zeroes" for this transfer function – instead, the filter only outputs zero signal when s reaches infinity. However, we may identify "poles" in this transfer function by solving for roots of the denominator, that is values of s resulting in the polynomial attaining a value of zero which would make the transfer function fraction equal $\frac{1}{0}$:

$$s^2C_1C_2R_1R_2 + sC_1(R_1 + R_2) + 1 = 0$$

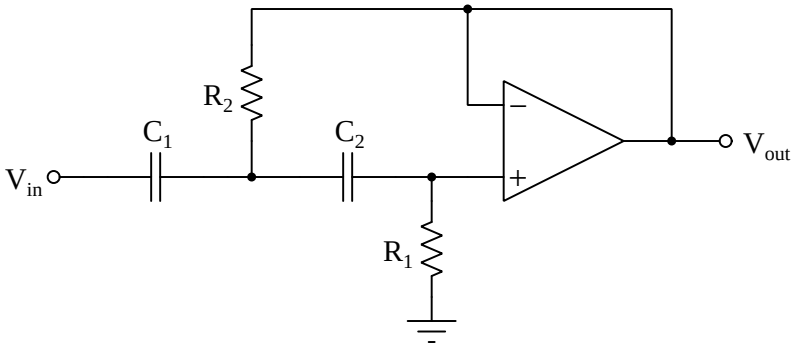
The quadratic formula is helpful here because this expression is a second-degree polynomial:

$$s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$s = \frac{-C_1(R_1 + R_2) \pm \sqrt{(C_1(R_1 + R_2))^2 - (4)(C_1C_2R_1R_2)(1)}}{(2)(C_1C_2R_1R_2)}$$

3.12.2 Sallen-Key high-pass filter transfer function derivation

Now let us turn our attention to the high-pass version of the Sallen-Key filter network topology, approaching this filter much the same as we did the low-pass Sallen-Key filter:



For this derivation we will simply list the steps taken and let annotation of the schematic diagram for each step be an exercise for the reader:

1. Define V_{out} at the opamp's output terminal as positive with respect to ground
2. Label voltage across resistor R_1 also as V_{out} with + on top and − on bottom, knowing the opamp will hold those two voltages equal at all times due to negative feedback
3. Calculate current downward through R_1 by Ohm's Law: $I_{R1} = \frac{V_{out}}{R_1}$
4. The same current must pass through capacitor C_2 (left to right) because the opamp input terminals pass negligible current; $I_{C2} = I_{R1} = \frac{V_{out}}{R_1}$
5. Calculate voltage drop across C_2 as $\frac{V_{out}}{sC_2R_1}$ using Ohm's Law with $Z_{C2} = \frac{1}{sC_2}$, that voltage polarity being + on left and − on right
6. The same voltage drop must also appear across resistor R_2, also owing to the opamp's negative feedback action: $V_{R2} = V_{C2} = \frac{V_{out}}{sC_2R_1}$ with − on top and + on bottom
7. Calculate current through R_2 using Ohm's Law: $I_{R2} = \frac{V_{out}}{sC_2R_1R_2}$ flowing upward out of the node connecting C_1, C_2, and R_2 together
8. Calculate current through capacitor C_1 using Kirchhoff's Current Law, being the sum of R_2's current and C_2's current: $I_{R1} = \frac{V_{out}}{sC_2R_1R_2} + \frac{V_{out}}{R_1}$
9. Calculate voltage across C_1 using Ohm's Law with the impedance of that capacitor being $Z_{C1} = \frac{1}{sC_1}$ such that $V_{C1} = \frac{1}{sC_1}\left(\frac{V_{out}}{sC_2R_1R_2} + \frac{V_{out}}{R_1}\right)$, + on left and − on right, with the Z_{C1} term distributed the voltage expression becomes $V_{C1} = \frac{V_{out}}{s^2C_1C_2R_1R_2} + \frac{V_{out}}{sC_1R_1}$
10. Apply Kirchhoff's Voltage Law to arrive at an expression for V_{in} (+ with respect to ground) that is the sum of V_{C1}, V_{C2}, and V_{R1}: $V_{in} = \frac{V_{out}}{s^2C_1C_2R_1R_2} + \frac{V_{out}}{sC_1R_1} + \frac{V_{out}}{sC_2R_1} + V_{out}$

From here we may use algebra to arrive at a final transfer function defined as $\frac{Vout}{V_{in}}$:

$$V_{in} = \frac{V_{out}}{s^2 C_1 C_2 R_1 R_2} + \frac{V_{out}}{sC_1 R_1} + \frac{V_{out}}{sC_2 R_1} + V_{out}$$

$$V_{in} = V_{out}\left(\frac{1}{s^2 C_1 C_2 R_1 R_2} + \frac{1}{sC_1 R_1} + \frac{1}{sC_2 R_1} + 1\right)$$

$$\frac{V_{in}}{V_{out}} = \frac{1}{s^2 C_1 C_2 R_1 R_2} + \frac{1}{sC_1 R_1} + \frac{1}{sC_2 R_1} + 1$$

$$\frac{V_{in}}{V_{out}} = \frac{1}{s^2 C_1 C_2 R_1 R_2} + \frac{sC_2 R_2}{s^2 C_1 C_2 R_1 R_2} + \frac{sC_1 R_2}{s^2 C_1 C_2 R_1 R_2} + \frac{s^2 C_1 C_2 R_1 R_2}{s^2 C_1 C_2 R_1 R_2}$$

$$\frac{V_{in}}{V_{out}} = \frac{1 + sC_2 R_2 + sC_1 R_2 + s^2 C_1 C_2 R_1 R_2}{s^2 C_1 C_2 R_1 R_2}$$

$$\frac{V_{out}}{V_{in}} = \frac{s^2 C_1 C_2 R_1 R_2}{1 + sC_2 R_2 + sC_1 R_2 + s^2 C_1 C_2 R_1 R_2}$$

$$\frac{V_{out}}{V_{in}} = \frac{s^2 C_1 C_2 R_1 R_2}{s^2 C_1 C_2 R_1 R_2 + sC_1 R_2 + sC_2 R_2 + 1}$$

$$\frac{V_{out}}{V_{in}} = \frac{s^2 C_1 C_2 R_1 R_2}{s^2 C_1 C_2 R_1 R_2 + sR_2(C_1 + C_2) + 1}$$

There is a single "zero" for this transfer function at $s = 0$ because this will make the transfer function have a value of zero. "Poles" in this transfer function may be found by solving for roots of the denominator, that is values of s resulting in the polynomial attaining a value of zero which would make the transfer function fraction equal $\frac{1}{0}$:

$$s^2 C_1 C_2 R_1 R_2 + sR_2(C_1 + C_2) + 1 = 0$$

The quadratic formula is helpful here because this expression is a second-degree polynomial:

$$s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$s = \frac{-R_2(C_1 + C_2) \pm \sqrt{(R_2(C_1 + C_2))^2 - (4)(C_1 C_2 R_1 R_2)(1)}}{(2)(C_1 C_2 R_1 R_2)}$$

Chapter 4

Historical References

This chapter is where you will find references to historical texts and technologies related to the module's topic.

Readers may wonder why historical references might be included in any modern lesson on a subject. Why dwell on old ideas and obsolete technologies? One answer to this question is that the initial discoveries and early applications of scientific principles typically present those principles in forms that are unusually easy to grasp. Anyone who first discovers a new principle must necessarily do so from a perspective of ignorance (i.e. if you truly *discover* something yourself, it means you must have come to that discovery with no prior knowledge of it and no hints from others knowledgeable in it), and in so doing the discoverer lacks any hindsight or advantage that might have otherwise come from a more advanced perspective. Thus, discoverers are forced to think and express themselves in less-advanced terms, and this often makes their explanations more readily accessible to others who, like the discoverer, comes to this idea with no prior knowledge. Furthermore, early discoverers often faced the daunting challenge of explaining their new and complex ideas to a naturally skeptical scientific community, and this pressure incentivized clear and compelling communication. As James Clerk Maxwell eloquently stated in the Preface to his book *A Treatise on Electricity and Magnetism* written in 1873,

> It is of great advantage to the student of any subject to read the original memoirs on that subject, for science is always most completely assimilated when it is in its nascent state . . . `[page xi]`

Furthermore, grasping the historical context of technological discoveries is important for understanding how science intersects with culture and civilization, which is ever important because new discoveries and new applications of existing discoveries will always continue to impact our lives. One will often find themselves impressed by the ingenuity of previous generations, and by the high degree of refinement to which now-obsolete technologies were once raised. There is much to learn and much inspiration to be drawn from the technological past, and to the inquisitive mind these historical references are treasures waiting to be (re)-discovered.

4.1 Wave screens

Charles Proteus Steinmetz was an electrical engineer employed for many years by the General Electric Company in New York. He was widely recognized as a genius in this field, and did much to elevate the mathematical rigor of electrical engineering. In his book *Theory and Calculation of Electric Circuits* first published in 1917 he describes the use of capacitance and inductance to form filtering circuits which he referred to as *wave screens* useful for separating alternating and direct current components of any pulsating electrical signal:

> **78.** By "wave screens" the separation of pulsating currents into their alternating and their continuous component, or the separation of complex alternating currents – and thus voltages – into their constituent harmonics can be accomplished, and inversely, the combination of alternating and continuous currents or voltages into resultant complex alternating or pulsating currents.
>
> The simplest arrangement of such a wave screen for separating, or combining, alternating and continuous currents into pulsating ones, is the combination, in shunt with each other, of a capacity, C, and an inductance, L, as shown in Fig. 75. If, then, a pulsating voltage, e, is impressed upon the system, the pulsating current, i, produced by it divides, as the continuous component can not pass through the condenser, C, and the alternating component is barred by the inductance, L, the more completely, the higher this inductance. Thus the current, i_1, in the apparatus, A, is a true alternating current, while the current, i_0, in the apparatus, C, is a slightly pulsating direct current. [page 156]

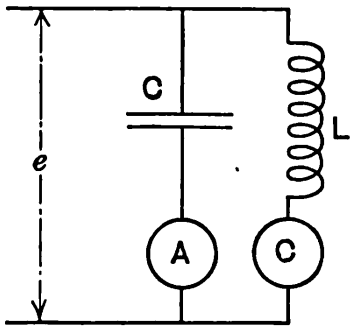


FIG. 75.

In this illustration A and C each represent electrical ammeters registering current through their respective branches of the parallel ("shunt") network.

On the next page, Steinmetz describes the use of series LC resonance to form band-pass filters useful for separating various harmonic[1] frequencies from a complex AC signal:

> Wave screens based on resonance for a definite frequency by series connection of capacity and inductance, can be used to separate the current of this frequency from a complex current or voltage wave, such as those given in Figs. 56 to 63, and thus can be used for the separation of complex waves into their components, by "harmonic analysis."
>
> Thus in Fig. 76, if the successive capacities and inductances are chosen such that
>
> $$2\pi f L_1 = \frac{1}{2\pi f C_1} ,$$
>
> $$6\pi f L_3 = \frac{1}{6\pi f C_3} ,$$
>
> $$10\pi f L_5 = \frac{1}{10\pi f C_5} ,$$
>
> $$2n\pi f L_n = \frac{1}{2\pi f n C_n}$$
>
> where f = frequency of the fundamental wave. [page 180]

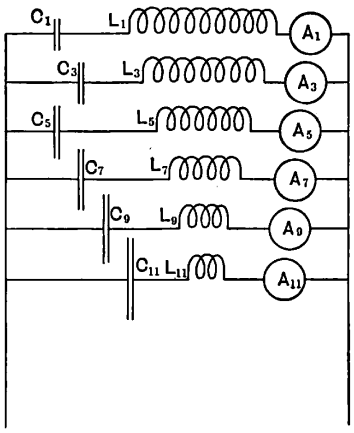


Fig. 76.

Steinmetz's conception of multiple band-pass filter networks tuned to resonate with respective harmonics of a known fundamental frequency, each one connected to its own ammeter to register the strength of each harmonic, is analogous to obsolete vibrating-reed frequency meters with their multiple reeds tuned to different frequencies.

[1] As mathematically proven by Fourier, any periodic wave of any shape whatsoever is mathematically equivalent to the sum of a set of sinusoidal waves having frequency values equal to whole-numbered multiples of the fundamental frequency of the complex wave. For example, a complex-shaped waveform having a frequency of 45 Hz may consist of a 45 Hz "fundamental" sinusoid (the first harmonic) plus other sinusoidal waves of specific amplitudes having frequencies of 90 Hz (the second harmonic), 135 Hz (the third harmonic), 180 Hz (the fourth harmonic), etc.

Chapter 5

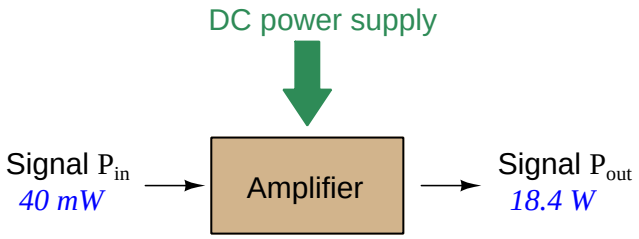
Derivations and Technical References

This chapter is where you will find mathematical derivations too detailed to include in the tutorial, and/or tables and other technical reference material.

5.1 Decibels

One of the mathematical tools popularly used to gauge increases and decreases of electrical power is the *common logarithm*, expressed as a measurement unit called the *decibel*. The basic idea of decibels is to express a ratio of two electrical power quantities in logarithmic terms. Every time you see the unit of "decibel" you can think: *this is an expression of how much greater (or how much smaller) one power is to another*. The only question is which two powers are being compared.

Electronic amplifiers are a type of electrical system where comparisons of power are useful. Students of electronics learn to compare the output power of an amplifier against the input power as a unitless ratio, called a *gain*. Take for example an electronic amplifier with a signal input of 40 milliWatts and a signal output of 18.4 Watts:



$$\text{Gain} = \frac{P_{out}}{P_{in}} = \frac{18.4\ W}{40\ mW} = 460$$

An alternative way to express the gain of this amplifier is to do so using the unit of the *Bel*, defined as the common logarithm of the gain ratio:

$$\log\left(\frac{P_{out}}{P_{in}}\right) = \log\left(\frac{18.4\ \text{W}}{40\ \text{mW}}\right) = 2.66276\ \text{B}$$

When you see an amplifier gain expressed in the unit of "Bel", it's really just a way of saying "The output signal coming from this amplifier is x powers of ten greater than the input signal." An amplifier exhibiting a gain of 1 Bel outputs 10 times as much power as the input signal. An amplifier with a gain of 2 Bels boosts the input signal by a factor of 100. The amplifier shown above, with a gain of 2.66276 Bels, boosts the input signal 460-fold.

At some point in technological history it was decided that the "Bel" (B) was too large and cumbersome, and so it became common to express powers in fractions of a Bel instead: the *deci*Bel (1 dB = $\frac{1}{10}$ of a Bel). Therefore, this is the form of formula you will commonly see for expressing electrical signal power gains or losses:

$$\text{dB} = 10 \log\left(\frac{P_{out}}{P_{in}}\right)$$

The gain of our hypothetical electronic amplifier, therefore, would be more commonly expressed as 26.6276 dB rather than 2.66276 B, although either expression is technically valid[1].

[1] It is interesting to note that although the "Bel" is a metric unit, it is seldom if ever used without the metric prefix

An operation students often struggle with is converting a decibel figure back into a ratio, since the concept of logarithms seems to be universally perplexing. Here I will demonstrate how to algebraically manipulate the decibel formula to solve for the power ratio given a dB figure.

First, we will begin with the decibel formula as given, solving for a value in decibels given a power ratio:

$$\text{dB} = 10 \log(\text{Ratio})$$

If we wish to solve for the ratio, we must "undo" all the mathematical operations surrounding that variable. One way to determine how to do this is to reverse the order of operations we would follow if we knew the ratio and were solving for the dB value. After calculating the ratio, we would then take the logarithm of that value, and then multiply that logarithm by 10: start with the ratio, then take the logarithm, then multiply last. To un-do these operations and solve for the ratio, we must un-do each of these operations in reverse order. First, we must un-do the multiplication (by dividing by 10):

$$\frac{\text{dB}}{10} = \frac{10 \log(\text{Ratio})}{10}$$

$$\frac{\text{dB}}{10} = \log(\text{Ratio})$$

Next, we must un-do the logarithm function by applying its mathematical inverse to both sides of the formula – making each expression a power of 10:

$$10^{\frac{\text{dB}}{10}} = 10^{\log(\text{Ratio})}$$

$$10^{\frac{\text{dB}}{10}} = \text{Ratio}$$

To test our algebra, we can take the previous decibel value for our hypothetical amplifier and see if this new formula yields the original gain ratio:

$$\text{Ratio} = 10^{\frac{26.6276 \text{ dB}}{10}}$$

$$\text{Ratio} = 10^{2.66276 \text{ B}}$$

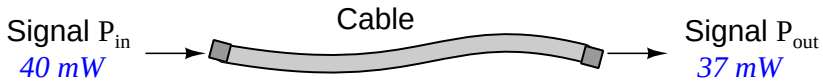
$$\text{Ratio} = 460$$

Sure enough, we arrive at the correct gain ratio of 460, starting with the decibel gain figure of 26.6276 dB.

"deci" ($\frac{1}{10}$). One could express powers in microbels, megabels, or any other metric prefix desired, but it is never done in industry: only the decibel is used.

We may also use decibels to express power *losses* in addition to power *gains*. There are many practical applications of this in signaling systems, both electronic and optical. One such application is *filtering*, where a "filter" circuit screens out certain components of the signal while letting others pass through (e.g. the bass or treble control for an audio system). Another application is *attenuation*, where the entirety of a signal is reduced in magnitude (e.g. the volume control for an audio system).

We will explore yet another application of signal power reduction as a case study for decibels: *cable loss*. Cables designed to convey signals over long distances are not perfect conduits of energy, as some of the signal's energy is inevitably lost along the way. This is true for different types of signals, electrical and optical being two popular examples. In the following illustration we see a signal cable losing power along its length[2], such that the power out is less than the power in:



$$\text{Gain} = \frac{P_{out}}{P_{in}} = \frac{37\ mW}{40\ mW} = 0.925$$

$$10 \log \left(\frac{P_{out}}{P_{in}} \right) = 10 \log \left(\frac{37 \text{ mW}}{40 \text{ mW}} \right) = -0.3386 \text{ dB}$$

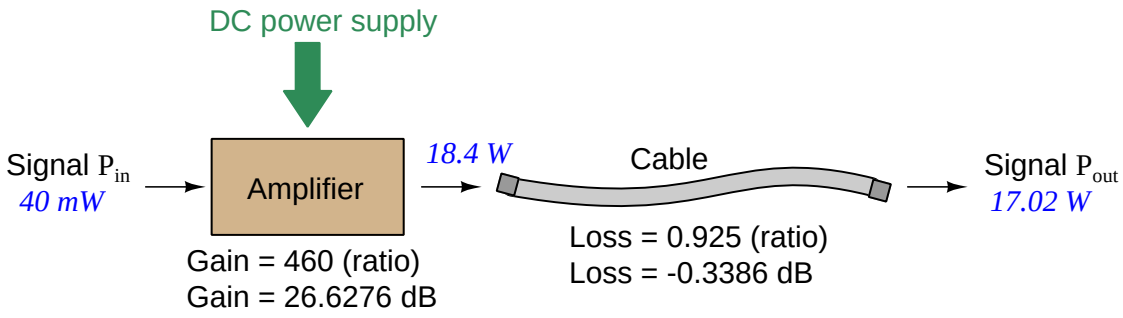
Contrasting this result against the previous result (with the amplifier) we see a very important property of decibel figures: any power *gain* is expressed as a *positive* decibel value, while any power *loss* is expressed as a *negative* decibel value. Any component outputting the exact same power as it takes in will exhibit a "gain" value of 0 dB (equivalent to a gain *ratio* of 1).

Remember that Bels and decibels are nothing more than logarithmic expressions of "greater than" and "less than". Positive values represent powers that are *greater* while negative values represent powers that are *lesser*. Zero Bel or decibel values represent *no change* (neither gain nor loss) in power.

A couple of simple decibel values are useful to remember for approximations, where you need to quickly estimate decibel values from power ratios (or vice-versa). Each addition or subtraction of 10 dB exactly represents a 10-fold multiplication or division of power ratio: e.g. +20 dB represents a power ratio gain of $10 \times 10 = 100$, whereas −30 dB represents a power ratio reduction of $\frac{1}{10} \times \frac{1}{10} \times \frac{1}{10} = \frac{1}{1000}$. Each addition or subtraction of 3 dB approximately represents a 2-fold multiplication or division or power ratio: e.g. +6 dB is approximately equal to a power ratio gain of $2 \times 2 = 4$, whereas −12 dB is approximately equal to a power ratio reduction of $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{16}$. We may combine ± 10 dB and ± 3 dB increments to come up with ratios that are products of 10 and 2: e.g. +26 dB is approximately equal to a power ratio gain of $10 \times 10 \times 2 \times 2 = 400$.

[2]For high-frequency signals such as those used in radio communications, the dominant mode of energy dissipation is *dielectric heating*, where the AC electric field between the cable conductors excites the molecules of the conductor insulation. This energy loss manifests as heat, which explains why there is less signal energy present at the load end of the cable than is input at the source end of the cable. For DC and low-frequency AC circuits the dominant mode of energy dissipation is cable conductor resistance, which is typically very small.

Observe what happens if we combine a "gain" component with a "loss" component and calculate the overall power out versus power in:



The overall gain of this amplifier and cable system expressed as a ratio is equal to the *product* of the individual component gain/loss ratios. That is, the gain ratio of the amplifier *multiplied* by the loss ratio of the cable yields the overall power ratio for the system:

$$\text{Overall gain} = \frac{17.02 \text{ W}}{40 \text{ mW}} = (460)(0.925) = 425.5$$

The overall gain may be alternatively expressed as a decibel figure, in which case it is equal to the *sum* of the individual component decibel values. That is, the decibel gain of the amplifier *added* to the decibel loss of the cable yields the overall decibel figure for the system:

$$\text{Overall gain} = 10 \log \left(\frac{17.02 \text{ W}}{40 \text{ mW}} \right) = 26.6276 \text{ dB} + (-0.3386 \text{ dB}) = 26.2890 \text{ dB}$$

It is often useful to be able to estimate decibel values from power ratios and vice-versa. If we take the gain ratio of this amplifier and cable system (425.5) and round it down to 400, we may easily express this gain ratio as an expanded product of 10 and 2:

$$425.5 \approx 400 = (10) \times (10) \times (2) \times (2)$$

Knowing that every 10-fold multiplication of power ratio is an addition of +10 dB, and that every 2-fold multiplication of power is an addition of +3 dB, we may express the expanded product as a sum of decibel values:

$$(10) \times (10) \times (2) \times (2) = (10 \text{ dB}) + (10 \text{ dB}) + (3 \text{ dB}) + (3 \text{ dB}) = 26 \text{ dB}$$

Therefore, our power ratio of 425.5 is approximately equal to +26 decibels.

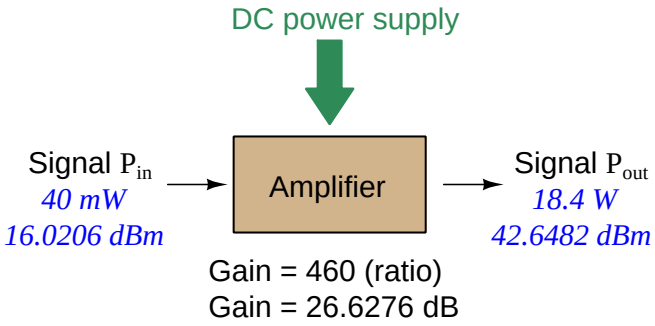
Decibels always represent comparisons of power, but that comparison need not always be P_{out}/P_{in} for a system component. We may also use decibels to express an amount of power compared to some standard reference. If, for example, we wished to express the input power to our hypothetical amplifier (40 milliWatts) using decibels, we could do so by comparing 40 mW against a standard "reference" power of exactly 1 milliWatt. The resulting decibel figure would be written as "dBm" in honor of the 1 *m*illiWatt reference:

$$P_{in} = 10 \log \left(\frac{40 \text{ mW}}{1 \text{ mW}} \right) = 16.0206 \text{ dBm}$$

The unit of "dBm" literally means the amount of dB "greater than" 1 milliWatt. In this case, our input signal of 40 milliWatts is 16.0206 dB greater than a standard reference power of exactly 1 milliWatt. The output power of that amplifier (18.4 Watts) may be expressed in dBm as well:

$$P_{out} = 10 \log \left(\frac{18.4 \text{ W}}{1 \text{ mW}} \right) = 42.6482 \text{ dBm}$$

A signal power of 18.4 Watts is 42.6482 dB greater than a standard reference power of exactly 1 milliWatt, and so it has a decibel value of 42.6482 dBm.



Notice how the output and input powers expressed in dBm relate to the power gain of the amplifier. Taking the input power and simply *adding* the amplifier's gain factor yields the amplifier's output power in dBm:

$$P_{in}(\text{dB}) + P_{gain}(\text{dB}) = P_{out}(\text{dB})$$

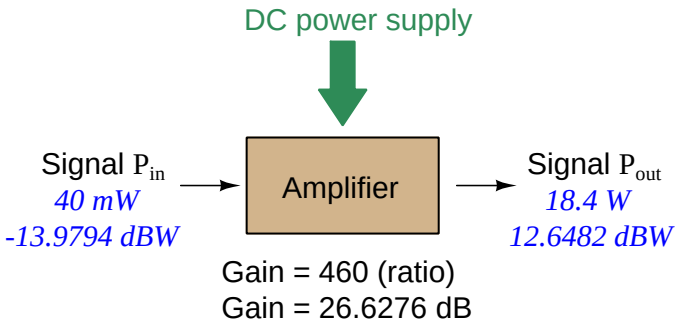
$$16.0206 \text{ dBm} + 26.6276 \text{ dB} = 42.6482 \text{ dBm}$$

An electronic signal that begins 16.0206 dB greater than 1 milliWatt, when boosted by an amplifier gain of 26.6276 dB, will become 42.6482 dB greater than the original reference power of 1 milliWatt.

We may alternatively express all powers in this hypothetical amplifier in reference to a 1-Watt standard power, with the resulting power expressed in units of "dBW" (decibels greater than 1 Watt):

$$P_{in} = 10 \log \left(\frac{40 \text{ mW}}{1 \text{ W}} \right) = -13.9794 \text{ dBW}$$

$$P_{out} = 10 \log \left(\frac{18.4 \text{ W}}{1 \text{ W}} \right) = 12.6482 \text{ dBW}$$



Note how the input power of 40 milliWatts equates to a negative dBW figure because 40 milliWatts is *less* than the 1 Watt reference, and how the output power of 18.4 Watts equates to a positive dBW figure because 18.4 Watts is *more* than the 1 Watt reference. A positive dB figure means "more than" while a negative dB figure means "less than."

Note also how the output and input powers expressed in dBW still relate to the power gain of the amplifier by simple addition, just as they did when previously expressed in units of dBm. Taking the input power in units of dBW and simply *adding* the amplifier's gain factor yields the amplifier's output power in dBW:

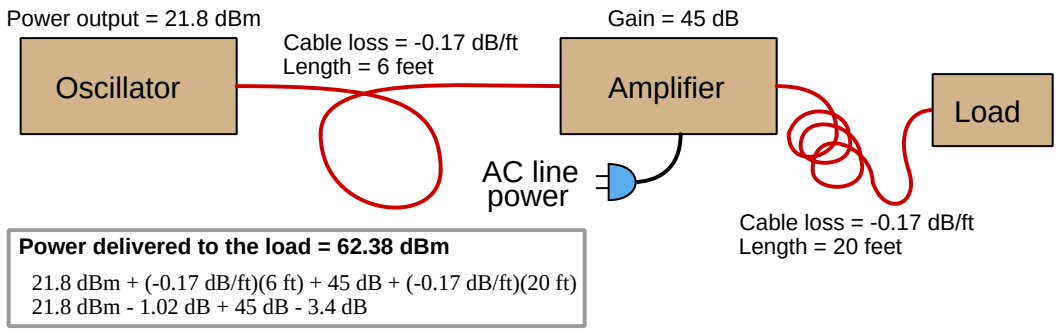
$$P_{in}(\text{dB}) + P_{gain}(\text{dB}) = P_{out}(\text{dB})$$

$$-13.9794 \text{ dBW} + 26.6276 \text{ dB} = 12.6482 \text{ dBW}$$

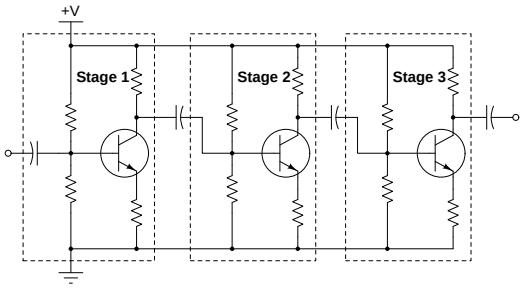
An electronic signal that begins 13.9794 dB less than 1 Watt, when boosted by an amplifier gain of 26.6276 dB, will become 12.6482 dB greater than the original reference power of 1 Watt.

This is one of the major benefits of using decibels to express powers: we may very easily calculate power gains and losses by summing a string of dB figures, each dB figure representing the power gain or power loss of a different system component. Normally, any compounding of *ratios* involves multiplication and/or division of those ratios, but with decibels we may simply add and subtract. One of the interesting mathematical properties of logarithms is that they "transform[3]" one type of problem into a simpler type: in this case, a problem of multiplying ratios into a (simpler) problem of adding decibel figures.

For example, we may express the power dissipated along a cable in terms of decibels per foot; the longer the cable, of course, the more power will be lost this way, all other factors being equal. For example, a radio-frequency signal cable having a loss figure of −0.15 decibels per foot at a signal frequency of 2.4 GHz will suffer −15 dB over 100 feet, and −150 dB over 1000 feet. To illustrate how decibels may be used to calculate power delivered to a load in such a system, accounting for various gains and losses along the way using decibel figures:



A similar application of decibels is found in multi-stage amplifier circuits, where one stage amplifies a signal to be fed into a successive stage to be amplified more. The power gains of these stages, each expressed as a ratio, *multiply* to make the over-all amplifier's power gain (ratio). The power gains of those same stages, each expressed as a decibel figure, *add* to make the over-all amplifier's power gain (dB):



[3]In fact, logarithms are one of the simplest examples of a *transform function*, converting one type of mathematical problem into another type. Other examples of mathematical transform functions used in engineering include the *Fourier transform* (converting a time-domain function into a frequency-domain function) and the *Laplace transform* (converting a differential equation into an algebraic equation).

Another common application of decibels is to express ratios of voltage and/or current rather than power. However, since the unit of the Bel has been defined as an expression of a *power* ratio, we cannot simply substitute V or I for P in any of the formulae we've seen so far.

Suppose an amplifier has a voltage gain of 2 (i.e. V_{out} is twice as large as V_{in}), and we would like to express this gain in decibels. Since decibels are intended to express power gain and not voltage gain, we must figure out how much power gain is equivalent to a voltage gain of two. Obviously, voltage and power are fundamentally different quantities, but if we imagine ourselves connecting a fixed load resistance to the input signal, and then to the output signal, we will realize that load's power dissipation will be more than double when energized by a voltage twice as large. Joule's Law is helpful to determine the exact ratio of power dissipation:

$$P = \frac{V^2}{R}$$

Doubling voltage for any given load resistance results in power *quadrupling* because power is proportional to the square of the voltage applied to a fixed resistance. Using this as the basis for applying decibels to a voltage ratio. Knowing that Joule's Law also declares power is proportional to the square of the current applied to a fixed resistance ($P = I^2R$) means this same mathematical relationship will apply to current gains and reductions as well as voltage gains and reductions:

$$\text{dB} = 10 \log \left(\frac{P_{out}}{P_{in}} \right) = 10 \log \left(\frac{V_{out}}{V_{in}} \right)^2 = 10 \log \left(\frac{I_{out}}{I_{in}} \right)^2$$

An algebraic identity of logarithms is that the logarithm of any quantity raised to a power is equal to that power multiplied by the logarithm of the quantity. Expressed in general terms:

$$\log x^y = y \log x$$

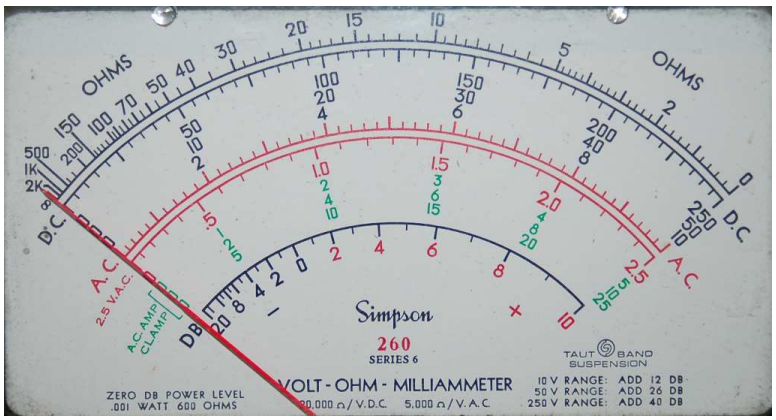
Therefore, we may simplify the decibel formula for voltage gain by removing the "2" power and making it a multiplier:

$$10 \log \left(\frac{V_{out}}{V_{in}} \right)^2 = (2)(10) \log \left(\frac{V_{out}}{V_{in}} \right) = 20 \log \left(\frac{V_{out}}{V_{in}} \right)$$

$$10 \log \left(\frac{I_{out}}{I_{in}} \right)^2 = (2)(10) \log \left(\frac{I_{out}}{I_{in}} \right) = 20 \log \left(\frac{I_{out}}{I_{in}} \right)$$

Thus, we may use decibels to express voltage or current ratios if we simply substitute 20 instead of 10 as the multiplier.

We can see the practicality of using decibels to represent something other than electrical *power* by examining this analog meter face, belonging to a Simpson model 260 VOM (Volt-Ohm-Milliammeter). Note the bottom scale on this meter's face, calibrated in decibels (DB):



Pay attention to the note on decibels written in the lower-left corner of the meter face, where 0 dB is defined as 0.001 Watt dissipated by 600 Ohms. The fact that 0 dB is defined as 1 milliWatt means it should (properly) be labeled dBm rather than dB[4]. A load resistance value is necessary as part of this definition for dB because this meter cannot measure power directly but must infer signal power from measurements of AC *voltage*. Without a specific load resistance, there is no clear relation between voltage and power. 600 Ohms is an old telecommunications standard for audio-frequency AC signals, and continues to be used today for voltage-based decibel measurements of audio-frequency AC signals.

The meter as shown is connected to nothing at all, and so registers 0 Volts AC. This, of course, corresponds to zero power, and it has no corresponding decibel value because the logarithm of zero is mathematically *undefined*[5]. Practically, it means $-\infty$ dB, which is why the needle at the 0 Volt position "falls off" the left-hand end of the dB scale.

Close inspection of the dB scale on this meter face reveals another interesting property of decibels, and that is the *nonlinear* nature of the dB scale. This contrasts starkly against all the voltage and current scales on this meter face which are linear. This nonlinearity is a fundamental property of decibels because it is based on the *logarithm* function which is nonlinear.

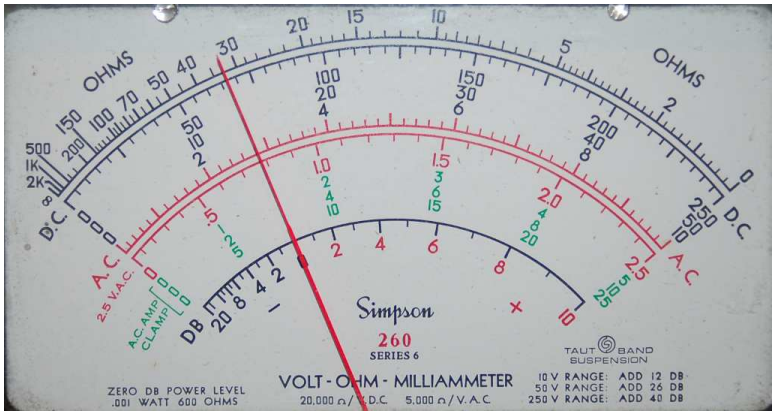
[4] Such mis-labeling is not that uncommon in the profession, the expectation being that the technician or engineer working with the instrument ought to be familiar enough with the concept of decibels to know when dB really means dBm, or dBW, etc.

[5] Your electronic calculator will complain if you attempt to take the logarithm of zero!

Now, we will explore what is necessary to make this meter register 0 dBm (i.e. 1 milliWatt) with an applied AC voltage. 1 milliWatt of power dissipated by 600 Ohms is equivalent to:

$$V = \sqrt{PR} = \sqrt{(0.001)(600)} = 0.7746 \text{ Volts}$$

Setting the VOM to the 2.5 VAC range and applying just enough AC voltage to bring the needle to the 0 dB mark allows us to verify that this is indeed equivalent to just under 0.8 Volts (read on the 2.5 VAC scale):



In the lower-right corner of the meter face we see some notes regarding correction values for decibel measurements when using different AC voltage ranges. The dB scale is read directly when the meter is set on the 2.5 VAC range. When set on the 10 VAC range (i.e. a range *four times* as great), the meter's needle will experience a deflection one-fourth as much as when set to the 2.5 VAC range, and therefore it will point to a lesser (or even negative) value on the dB scale. Converting a voltage ratio of 0.25 into a decibel figure shows us how much less the needle will register on the dB scale when the voltage range is quadrupled:

$$20 \log \left(\frac{2.5}{10} \right) = -12.04 \text{ dB}$$

Therefore, when using the 10 VAC range instead of the 2.5 VAC range, one must add 12 dB to the reading. Likewise, we may prove each of the printed correction factors for the alternative voltage-measurement ranges listed (50 Volt AC range and 250 Volt AC range):

$$20 \log \left(\frac{2.5}{50} \right) = -26.02 \text{ dB}$$

$$20 \log \left(\frac{2.5}{250} \right) = -40.0 \text{ dB}$$

5.2 IEC standard component values

Components such as resistors, inductors, and capacitors are manufactured in several *standard values*, described by IEC standard 60063. Rather than having a single series of standard values, the IEC publishes lists called *E series* based on the number of unique values spanning a single *decade* (i.e. a 10:1 range).

The shortest of these series, called *E3* contains just three values: 10, 22, and 47. The next series is called *E6* with six unique values: 10, 15, 22, 33, 47, and 68. These values represent *significant values* for components, meaning the decimal point may be freely moved to create values spanning multiple decades. For example, "33" simply means one can expect to find components manufactured in values of 33, 3.3, 0.33, and 0.033 as well as 330, 3.3 k, 33 k, etc.

Although this may seem like a strange standard for component manufacturers to follow, there is a compelling logic to it. The terms of each series are closer-spaced at the low end than at the high end, and this allows for *series* and/or *parallel* combinations of components to achieve most any desired value. For example, in the E6 series we only have values with the significant figures 10, 15, 22, 33, 47, and 68, but this doesn't mean we are limited to *total* values with these significant figures. For example, if we needed 80 Ohms of resistance we could connect a 33 Ohm and 47 Ohm resistor together in series. 50 Ohms could be made from two 68 Ohm resistors in parallel (making 34 Ohms) plus a 15 Ohm and 1 Ohm resistor in series.

On the next page is a table showing the four most common E-series specified by IEC standard 60063.

E3	**E6**	**E12**	**E24**
10	10	10	10
			11
		12	12
			13
	15	15	15
			16
		18	18
			20
22	22	22	22
			24
		27	27
			30
	33	33	33
			36
		39	39
			43
47	47	47	47
			51
		56	56
			62
	68	68	68
			75
		82	82
			91

E48, *E96*, and *E192* series are also found in the IEC 60063 standard, used for components with tighter tolerance ratings than typical.

5.3 The meaning of the s variable

Euler's Relation allows us to express rotating sinusoidal wave functions as imaginary exponents of e. For example, $Ae^{j\theta}$ represents a phasor[6] of length A at an angle of θ radians. $Ae^{j\omega t}$ represents a phasor of length A rotating at a velocity of ω radians per second at a particular instant in time t. This happens to be an incredibly useful mathematical "trick" for representing sinusoidal waves in physical systems. For example, if we wished to mathematically express a sinusoidal AC voltage as a function of time with a peak voltage value of 10 Volts and a frequency of 60 hertz (377 radians per second, since $\omega = 2\pi f$), we could do so like this:

$$V(t) = 10e^{j377t}$$

Exponential functions aren't just useful for expressing sinusoidal waves, however. They also work well for expressing rates of *growth* and *decay*, as is the case with RC and L/R time-delay circuits where exponential functions describe the charging and discharging of capacitors and inductors. Here, the exponent is a real number rather than an imaginary number: the expression $e^{-t/\tau}$ approaching zero as time (t) increases. The Greek letter "tau" (τ) represents the *time constant* of the circuit, which for capacitive circuits is the product of R and C, and for inductive circuits is the quotient of L over R. For example, if we wished to mathematically express the decaying voltage across a 33 μF capacitor initially charged to 10 Volts as it dissipates its stored energy through a 27 kΩ resistor (the circuit having a time constant of 0.891 seconds, since $\tau = RC$), we could do so like this:

$$V(t) = 10e^{-(t/0.891)}$$

The sign of the exponential term here is very important: in this example we see it is a negative number. This tells us the function *decays* (approaches zero) over time, since larger positive values of t result in larger negative values of t/τ (recall from algebra that a negative exponent is the equivalent of reciprocating the expression, so that $e^{-x} = \frac{1}{e^x}$). If the exponent were a real positive number, it would represent some quantity *growing* exponentially over time. If the exponent were zero, it would represent a *constant* quantity. We expect a discharging resistor-capacitor circuit to exhibit decaying voltage and current values, and so the negative exponent sign shown here makes sense.

If imaginary exponents of e represent sinusoidal waves, and real exponents of e represent growth or decay, then a *complex* exponent of e (having both real and imaginary parts) must represent a sinusoidal wave growing or decaying in magnitude over time. Engineers use the lower-case Greek letter "omega" (ω) along with the imaginary operator j to represent the imaginary portion, and the lower-case Greek letter "sigma"[7] (σ) to represent the real portion. For example, if we wished to mathematically express a sine wave AC voltage with a frequency of 60 hertz (377 radians per second) and an amplitude beginning at 10 Volts but decaying with a time constant (τ) of 25 milliseconds ($\sigma = 1/\tau = 40$ time constants per second), we could do so like this:

$$V(t) = 10e^{-40t+j377t}$$

[6] A "phasor" is a complex-number expression of a sinusoidal function at some moment in time.

[7] σ is equal to the reciprocal of the signal's time constant τ. In other words, $\sigma = 1/\tau$.

We may factor time from the exponential terms in this expression, since t appears both in the real and imaginary parts:

$$V(t) = 10e^{(-40+j377)t}$$

With t factored out, the remaining terms $-40 + j377$ completely describe the sinusoidal wave's characteristics. The wave's decay rate is described by the real term ($\sigma = -40$ time constants per second), while the wave's phase is described by the imaginary term ($j\omega = 377$ radians per second). Engineers use a single variable s to represent the complex quantity $\sigma + j\omega$, such that any growing or decaying sinusoid may be expressed very succinctly as follows:

$$Ae^{st} = Ae^{(\sigma+j\omega)t} = Ae^{\sigma t}e^{j\omega t}$$

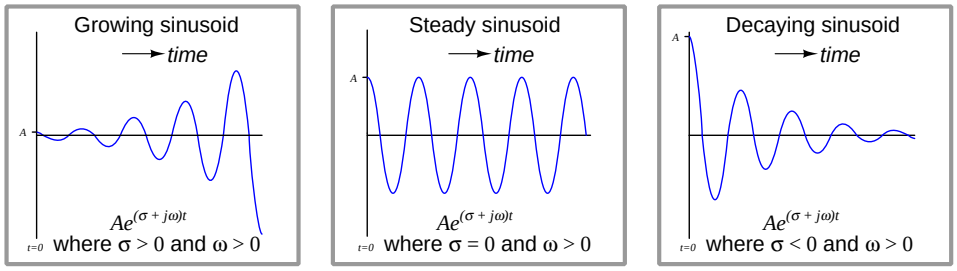
Where,

A = Initial amplitude of the sinusoid (e.g. Volts, Amperes) at time $t = 0$ (arbitrary units)
s = Complex growth/decay rate and frequency (sec^{-1})
$\sigma = \frac{1}{\tau}$ = Real growth/decay rate (time constants per second, or sec^{-1})
$j\omega$ = Imaginary frequency (radians per second, or sec^{-1})
t = Time (seconds)

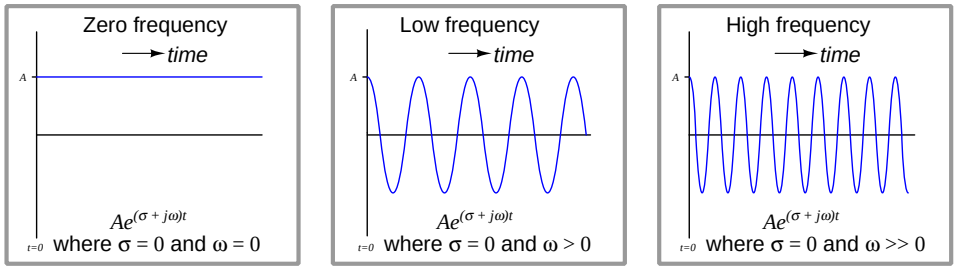
Separating the expression $Ae^{\sigma t}e^{j\omega t}$ into three parts – A, $e^{\sigma t}$, and $e^{j\omega t}$ – we get a complete description of a rotating phasor:

A = Initial amplitude of the phasor ($t = 0$)
$e^{\sigma t}$ = How much the phasor's magnitude has grown ($\sigma > 0$) or decayed ($\sigma < 0$) at time t
$e^{j\omega t}$ = Unit phasor (length = 1) at time t

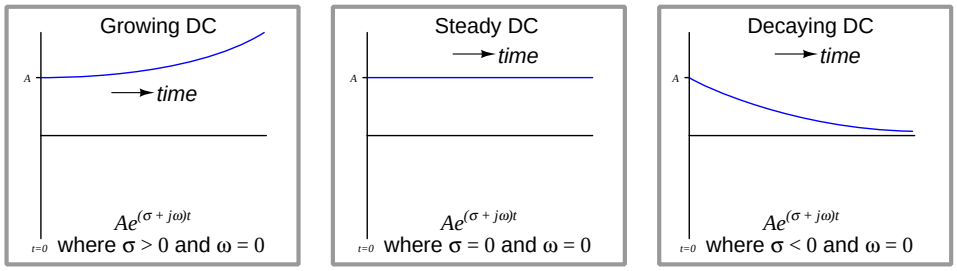
If we set ω at some constant value and experiment with different values of σ, we can see the effect σ has on the shape of the wave over time:



If we set σ at zero and experiment with different values[8] of ω, we can see the effect ω has on the shape of the wave over time:



If we set ω at zero to describe DC signals and experiment with different values of σ, we can see the effect σ has on the magnitude of DC signals over time:



As we will soon see, characterizing a sinusoidal response using the complex variable s allows us to mathematically describe a great many things. Not only may we describe voltage waveforms using s as shown in these simple examples, but we may also describe the response of entire physical systems including electrical filter networks, feedback control systems, servomechanisms, chemical reactions, and even nuclear chain-reactions. In fact, it is possible to map the essential characteristics of *any*

[8] One value of ω not shown in this three-panel graphic comparison is a *negative* frequency. This is actually not as profound as it may seem at first. All a negative value of ω will do is ensure that the phasor will rotate in the opposite direction (clockwise, instead of counter-clockwise as phasor rotation is conventionally defined). The real portion of the sinusoid will be identical, tracing the same cosine-wave plot over time. Only the imaginary portion of the sinusoid will be different, as $j \sin -\theta = -j \sin \theta$.

linear system in terms of how exponentially growing, decaying, or steady sinusoidal waves affect it, and that mapping takes the form of mathematical functions of s.

When engineers or technicians speak of a *resonant* system, they mean a circuit containing inductive and capacitive elements tending to sustain oscillations of a particular frequency (ω). A lossless resonant system (e.g. a superconducting tank circuit, a frictionless pendulum) may be expressed by setting the real portion of s equal to zero ($\sigma = 0$; no growth or decay) and letting the imaginary portion represent the resonant frequency ($j\omega = j2\pi f$). Real-life resonant systems inevitably dissipate some energy, and so a real resonant system's expression will have both an imaginary portion to describe resonant frequency and a negative real portion to describe the oscillation's rate of decay over time.

Systems exhibiting a positive σ value are especially interesting because they represent *instability*: unrestrained oscillatory growth over time. An aircraft auto-pilot control system with excessive sensitivity, for example, is a system where σ could become positive, in which case the aircraft would begin to pitch (or yaw, or roll) in an oscillating manner until the aircraft fell out of the sky. This situation, of course, is highly undesirable.

5.4 Impedance expressed using the s variable

We know that voltage across a capacitor and current "through" a capacitor are related as follows:

$$I = C\frac{dV}{dt}$$

Next, we substitute an expression[9] for voltage in terms of s and then use calculus to differentiate it with respect to time:

$$I = C\frac{d}{dt}\left(e^{st}\right)$$

$$I = sCe^{st}$$

The ratio of $\frac{V}{I}$ (the definition of impedance) will then be:

$$Z_C = \frac{V}{I} = \frac{e^{st}}{sCe^{st}}$$

$$Z_C = \frac{1}{sC}$$

Instead of the common scalar expression for capacitive impedance ($Z_C = \frac{1}{2\pi fC}$) which only tells us the magnitude of the impedance (in Ohms) but not the phase shift, we have a complex expression for capacitive impedance ($Z_C = \frac{1}{sC}$) describing magnitude, phase shift, and its reaction to the growth or decay of the signal.

[9] The expression used here to represent voltage is simply e^{st}. I could have used a more complete expression such as Ae^{st} (where A is the initial amplitude of the signal), but as it so happens this amplitude is irrelevant because there will be an "A" term in both the numerator and denominator of the impedance quotient. Therefore, A cancels out and is of no consequence.

Likewise, we may do the same for inductors. Recall that voltage across an inductor and current through an inductor are related as follows:

$$V = L\frac{dI}{dt}$$

Substituting an expression for current in terms of s and using calculus to differentiate it with respect to time:

$$V = L\frac{d}{dt}\left(e^{st}\right)$$

$$V = sLe^{st}$$

The ratio of $\frac{V}{I}$ (the definition of impedance) will then be:

$$Z_L = \frac{V}{I} = \frac{sLe^{st}}{e^{st}}$$

$$Z_L = sL$$

As with capacitors, we now have a complex expression for inductive impedance describing magnitude, phase shift, and its reaction to signal growth or decay ($Z_L = sL$) instead of merely having a scalar expression for inductive impedance ($Z_L = 2\pi fL$).

Resistors directly oppose current by dropping voltage, with no regard to rates of change. Therefore, there are no derivatives in the relationship between voltage across a resistor and current through a resistor:

$$V = IR$$

If we substitute e^{st} for current into this formula, we will see that voltage must equal Re^{st}. Solving for the ratio of voltage over current to define impedance:

$$Z_R = \frac{V}{I} = \frac{Re^{st}}{e^{st}}$$

$$Z_R = R$$

Not surprisingly, all traces of s cancel out for a pure resistor: its impedance is exactly equal to its DC resistance.

In summary:

Inductive impedance (Z_L)	**Capacitive impedance** (Z_C)	**Resistive impedance** (Z_R)
sL	$1/sC$	R

Now let's explore these definitions of impedance using real numerical values. First, let's consider a 22 μF capacitor exposed to a steady AC signal with a frequency of 500 Hz. Since the signal in this case is steady (neither growing nor decaying in magnitude), the value of σ will be equal to zero. ω is equal to $2\pi f$, and so a frequency of 500 Hz is equal to 3141.6 radians per second. Calculating impedance is as simple as substituting these values for s and computing $1/sC$:

$$Z_C = \frac{1}{sC} = \frac{1}{(\sigma + j\omega)C}$$

$$Z_C = \frac{1}{(0 + j3141.6 \text{ s}^{-1})(22 \times 10^{-6} \text{ F})}$$

$$Z_C = \frac{1}{j0.0691}$$

$$Z_C = \frac{-j}{0.0691}$$

$$Z_C = 0 - j14.469\,\Omega \quad \text{(rectangular notation)}$$

$$Z_C = 14.469\,\Omega \angle -90^o \qquad \text{(polar notation)}$$

Thus, the impedance of this capacitor will be 14.469 Ohms at a phase angle of -90^o. The purely imaginary nature of this impedance (its orthogonal phase shift between voltage and current) tells us there is no net power dissipated by the capacitor. Rather, the capacitor spends its time alternately absorbing and releasing energy to and from the circuit.

Next, we will consider the case of a 150 mH inductor exposed to an exponentially rising DC signal with a time constant (τ) of 5 seconds. 5 seconds per time constant (τ) is equal to 0.2 time constants per second (σ). Since the signal in this case is DC and not AC, the value of ω will be equal to zero. Calculating impedance, once again, is as simple as substituting these values for s and computing sL:

$$Z_L = sL = (\sigma + j\omega)L$$

$$Z_L = (0.2 + j0 \text{ s}^{-1})(150 \times 10^{-3} \text{ H})$$

$$Z_L = 0.03 + j0\,\Omega \quad \text{(rectangular notation)}$$

$$Z_L = 0.03\,\Omega \angle 0^o \qquad \text{(polar notation)}$$

Thus, the impedance of this inductor will be 0.03 Ohms at a phase angle of 0^o. The purely real nature of this impedance (i.e. no phase shift between voltage and current) tells us energy will be continually absorbed by the inductor, and for this reason it will be seen by the rest of the circuit as though it were a resistor dissipating energy for however long the signal continues to exponentially grow.

A phase shift of 0 degrees for a reactive component such as an inductor may come as a surprise to students accustomed to thinking of inductive impedances *always* having 90 degree phase shifts! However, the application of the complex variable s to impedance mathematically demonstrates we can indeed have conditions of no phase shift given just the right circumstances. This makes conceptual sense as well if we consider how inductors store energy: if the current through an inductor increases exponentially over time, never reversing direction, it means the inductor's magnetic field will always be growing and therefore absorbing more energy from the rest of the circuit.

We see something even more interesting happen when we subject a reactive component to a *decaying* DC signal. Take for example a 33,000 μF capacitor exposed to a decaying DC signal with a time constant of 65 milliseconds. 65 milliseconds per time constant (τ) is equal to 15.38 time constants per second (σ). Once again ω will be zero because this is a non-oscillating signal. Calculating capacitive impedance:

$$Z_C = \frac{1}{sC} = \frac{1}{(\sigma + j\omega)C}$$

$$Z_C = \frac{1}{(-15.38 + j0 \text{ s}^{-1})(33000 \times 10^{-6} \text{ F})}$$

$$Z_C = \frac{1}{-0.508}$$

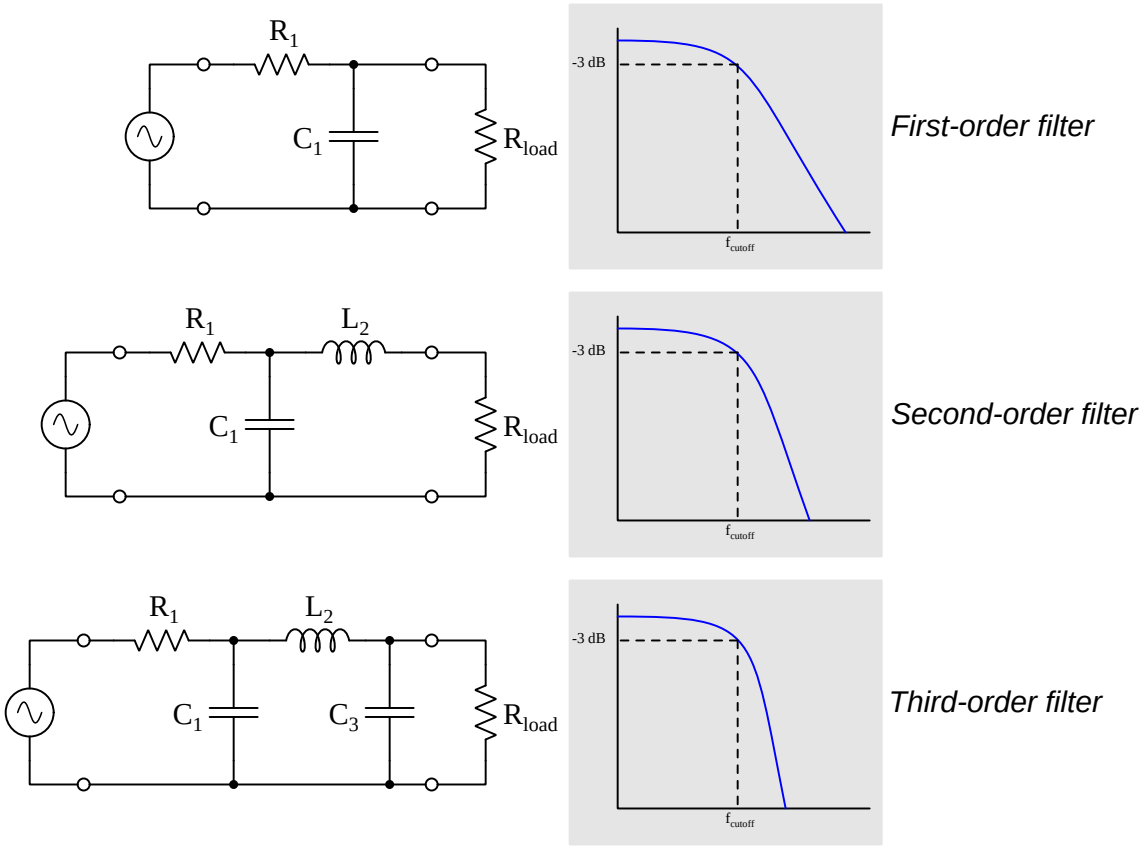
$$Z_C = -1.970 + j0\,\Omega \quad \text{(rectangular notation)}$$

$$Z_C = 1.970\,\Omega \angle 180^o \quad \text{(polar notation)}$$

A *negative* real impedance figure represents a phase shift of 180^o between voltage and current. Once again, this may surprise students of electronics who are accustomed to thinking of capacitive impedances *always* having phase shifts of -90 degrees. What a 180 degree phase shift means is the direction of current with respect to voltage polarity has the capacitor functioning as an energy *source* rather than as a *load*. If we consider what happens to a capacitor when it discharges, the 180 degree phase shift makes sense: current flowing in this direction depletes the capacitor's plates of stored charge, which means the electric field within the capacitor weakens over time as it releases that energy to the rest of the circuit.

5.5 Butterworth passive filter component values

Passive filter networks having a *ladder* topology may exhibit any number of desired orders by progressively adding reactive components in the pattern shown below, each successive stage causing the filter's roll-off rate to be steeper. These examples show low-pass filter networks of 1st-order, 2nd-order, and 3rd-order characteristic:



The choice of component values dictate the filter's cutoff frequency as well as its transfer function characteristic (e.g. Bessel, Butterworth, Chebyshev, Elliptic, etc.). That is to say, we could take the exact same topology and give it different component values causing it to exhibit any of these characteristics.

The following table lists capacitances in Farads and inductances in Henrys necessary to implement the Butterworth polynomial transfer function for a ladder-topology low-pass filter where input and load resistances are both 1 Ω and whose cutoff frequency (ω) is 1 radian per second:

$$R_{load} = 1\Omega \ ; \ \omega_{cutoff} = 1 \text{ rad/sec}$$

Order	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9
1	2.000	–	–	–	–	–	–	–	–
2	1.414	1.414	–	–	–	–	–	–	–
3	1.000	2.000	1.000	–	–	–	–	–	–
4	0.765	1.848	1.848	0.765	–	–	–	–	–
5	0.618	1.618	2.000	1.618	0.618	–	–	–	–
6	0.518	1.414	1.932	1.932	1.414	0.518	–	–	–
7	0.445	1.247	1.802	2.000	1.802	1.247	0.445	–	–
8	0.390	1.111	1.663	1.962	1.962	1.663	1.111	0.390	–
9	0.347	1.000	1.532	1.879	2.000	1.879	1.532	1.000	0.347

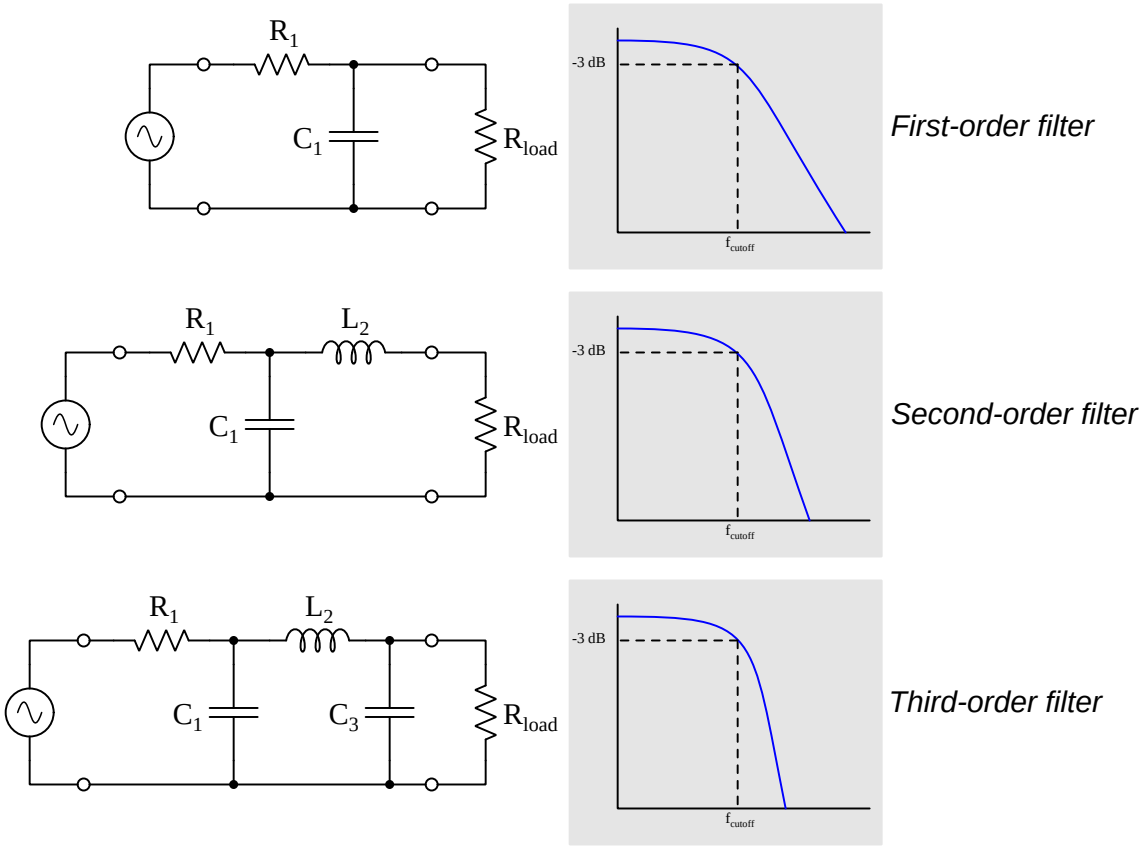
These "normalized" component values may be scaled for different input/load resistance values by multiplying all inductor values by the scaling factor and dividing all capacitance values by the same scaling factor. For example, to scale the filter's values for 600 Ω load and input resistances instead of 1 Ω as assumed by the table, you would multiply each inductance by 600 and divide each capacitance by 600.

These "normalized" component values may also be scaled for different cutoff frequency values by dividing all inductor and capacitor values by the same scaling factor. For example, to scale the filter's values for a cutoff frequency of 4000 radians per second instead of the 1 radian per second frequency assumed by the table, you would divide each inductance by 4000 and divide each capacitance by 4000.

The conversion factor between frequency in Hertz and frequency in radians per second is 2π, given that $\omega = 2\pi f$. For example, a natural frequency of 1000 radians per second is equal to 159.15 Hertz. Also, 60 Hertz is equal to 376.99 radians per second.

5.6 Chebyshev passive filter component values

Passive filter networks having a *ladder* topology may exhibit any number of desired orders by progressively adding reactive components in the pattern shown below, each successive stage causing the filter's roll-off rate to be steeper. These examples show low-pass filter networks of 1st-order, 2nd-order, and 3rd-order characteristic:



The choice of component values dictate the filter's cutoff frequency as well as its transfer function characteristic (e.g. Bessel, Butterworth, Chebyshev, Elliptic, etc.). That is to say, we could take the exact same topology and give it different component values causing it to exhibit any of these characteristics.

The following table lists capacitances in Farads and inductances in Henrys necessary to implement the Chebyshev polynomial transfer function for a ladder-topology low-pass filter where input and load resistances are both 1 Ω, where the cutoff frequency (ω) is 1 radian per second, and where the passband ripple is 1 dB:

$R_{load} = 1\Omega$; $\omega_{cutoff} = 1$ rad/sec ; Passband ripple = 1 dB

Order	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9
1	1.018	–	–	–	–	–	–	–	–
3	2.024	0.994	2.024	–	–	–	–	–	–
5	2.135	1.091	3.001	1.091	2.135	–	–	–	–
7	2.167	1.112	3.094	1.174	3.094	1.112	2.167	–	–
9	2.180	1.119	3.121	1.190	3.175	1.190	3.121	1.119	2.180

For the Chebyshev transfer function, only odd-numbered orders allow for equal input and load resistance values.

These "normalized" component values may be scaled for different input/load resistance values by multiplying all inductor values by the scaling factor and dividing all capacitance values by the same scaling factor. For example, to scale the filter's values for 600 Ω load and input resistances instead of 1 Ω as assumed by the table, you would multiply each inductance by 600 and divide each capacitance by 600.

These "normalized" component values may also be scaled for different cutoff frequency values by dividing all inductor and capacitor values by the same scaling factor. For example, to scale the filter's values for a cutoff frequency of 4000 radians per second instead of the 1 radian per second frequency assumed by the table, you would divide each inductance by 4000 and divide each capacitance by 4000.

The conversion factor between frequency in Hertz and frequency in radians per second is 2π, given that $\omega = 2\pi f$. For example, a natural frequency of 1000 radians per second is equal to 159.15 Hertz. Also, 60 Hertz is equal to 376.99 radians per second.

The following table lists capacitances in Farads and inductances in Henrys necessary to implement the Chebyshev polynomial transfer function for a ladder-topology low-pass filter where input and load resistances are both 1 Ω, where the cutoff frequency (ω) is 1 radian per second, and where the passband ripple is 3 dB:

$R_{load} = 1\Omega$; $\omega_{cutoff} = 1$ rad/sec ; Passband ripple = 3 dB

Order	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9
1	1.995	–	–	–	–	–	–	–	–
3	3.349	0.712	3.349	–	–	–	–	–	–
5	3.481	0.762	4.538	0.762	3.481	–	–	–	–
7	3.519	0.772	4.639	0.804	4.639	0.772	3.519	–	–
9	3.534	0.776	4.669	0.812	4.727	0.812	4.669	0.776	3.534

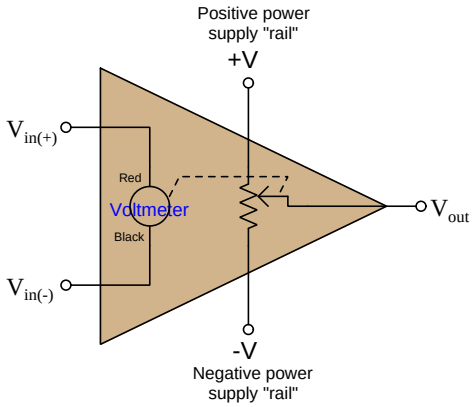
For the Chebyshev transfer function, only odd-numbered orders allow for equal input and load resistance values.

5.7 Operational amplifier imperfections

Comparators and operational amplifiers are incredibly useful and practical as electronic circuit building-blocks, but they are not perfect. For a great many applications these imperfections are insignificant, but for others they may be debilitating. Therefore, it is important to understand the ways in which operational amplifiers deviate from ideal behavior.

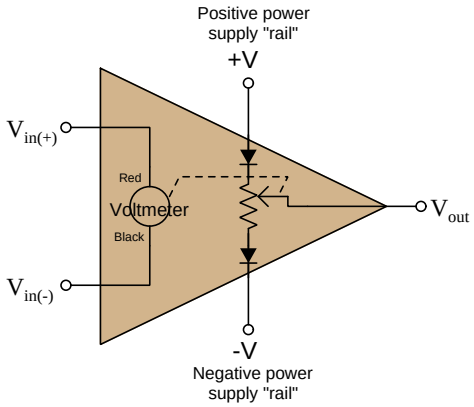
5.7.1 Output voltage limits

The output voltage of any amplifier circuit cannot exceed its DC supply "rail" voltages, because all an amplifier does is *modulate* its supply voltage. A useful model of a differential amplifier is that of a potentiometer with its wiper driven by the pointer of an analog voltmeter driven by the differential voltage applied to the input terminals:



No matter how much voltage is applied between the + and − input terminals, the potentiometer can only "swing" as far as the +V and −V "rails" of the DC power supply. This behavior is often referred to as *rail-to-rail output*, where the V_{out} limits are exactly (or very nearly) equal to the DC rail potentials.

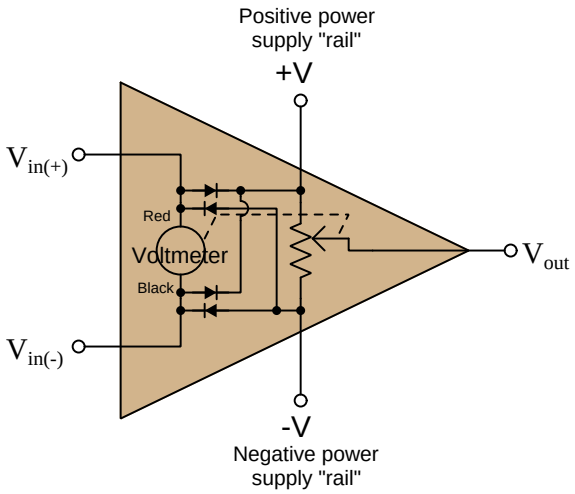
However, many practical amplifier models cannot even swing that far. A more accurate model for many is one where the potentiometer cannot quite reach the +V and −V rails:



With the diodes' voltage drops in effect, V_{out} is only able to "swing" to within approximately 0.7 Volts of each rail. Such internal voltage drops are typical of comparators and operational amplifiers using BJTs in the final output stage. Rail-to-rail capability demands FET output transistors which are capable of operating with mere milliVolts of drain-source voltage drop.

5.7.2 Input voltage limits and latch-up

As a general rule, input voltages for any integrated circuit should remain within the limits of its DC power supply "rail" potentials. Many integrated circuits including comparators and operational amplifiers include "protection" diodes at the input terminals effectively *clamping* input terminal potentials to the power supply rails (to within fractions of a Volt). Expanding upon our previous circuit model, we may show these protection diodes (two per input terminal) connected such that one will begin conducting if the potential at any input exceeds either power supply rail, essentially making that input terminal the new "rail" to power the amplifier:



If the common-mode[10] voltage applied to the inputs of a differential amplifier exceeds the point at which both transistors in the differential pair subcircuit turn off, the amplifier's output may saturate rather than respond properly to the differential input voltage. In some designs this may even lead to a condition where the output remains "latched" in this saturated mode even after the offending input condition ceases – a phenomenon known as *latch-up*[11].

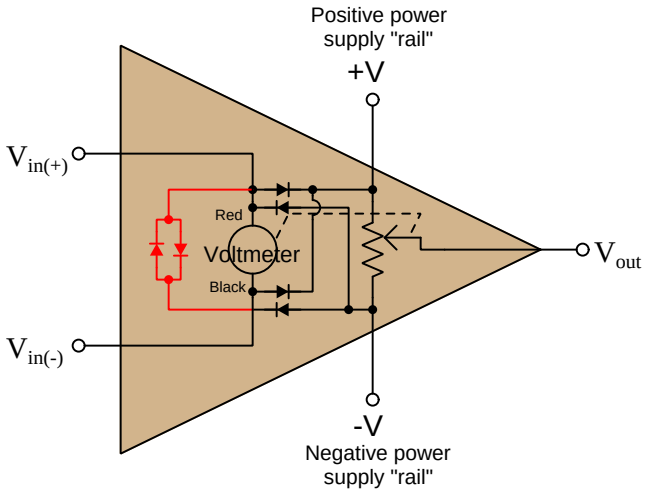
Amplifiers may fail to function properly even with input voltage values falling well within their supplied rail voltages. For example, the popular TL08x series of operational amplifiers specify common-mode voltage limits between +4 Volts above the negative rail voltage and −4 Volts below the positive rail voltage: if we happened to power one of these opamps on ± 12 Volt rails, the permissible input signal voltage range for either of its inputs would be −8 Volts to +8 Volts. In

[10] For a differential amplifier, the *common-mode voltage* is the amount of voltage with respect to ground that is shared among both input terminals, typically calculated as the average of the two inputs' ground-referenced voltages. In other words, if $V_{diff} = V_{in(+)} - V_{in(-)}$ then $V_{CM} = \frac{V_{in(+)} + V_{in(-)}}{2}$. However, for the sake of determining safe limits for amplifier input voltage we may more simply define "common-mode voltage" as being any voltage measured with respect to ground for either of a differential amplifier's two inputs that may be either high or low enough to cause a problem for that amplifier.

[11] Interestingly, the term "latch-up" more properly describes a condition in an integrated circuit where the particularly layering of semiconductor materials forms a PNPN (thyristor) structure accidently triggered by certain abnormal input conditions. Like the "latched" state just described, this triggered-thyristor state can only be reset by cycling power to the circuit.

contrast, other opamp models such as the MCP600x series offer *rail-to-rail inputs* capable of receiving any common-mode voltage up to and including the two DC supply rail voltages.

Some operational amplifiers additionally clamp *differential* input signals using a pair of paralleled protection diodes as shown in red below on the simplified opamp model:

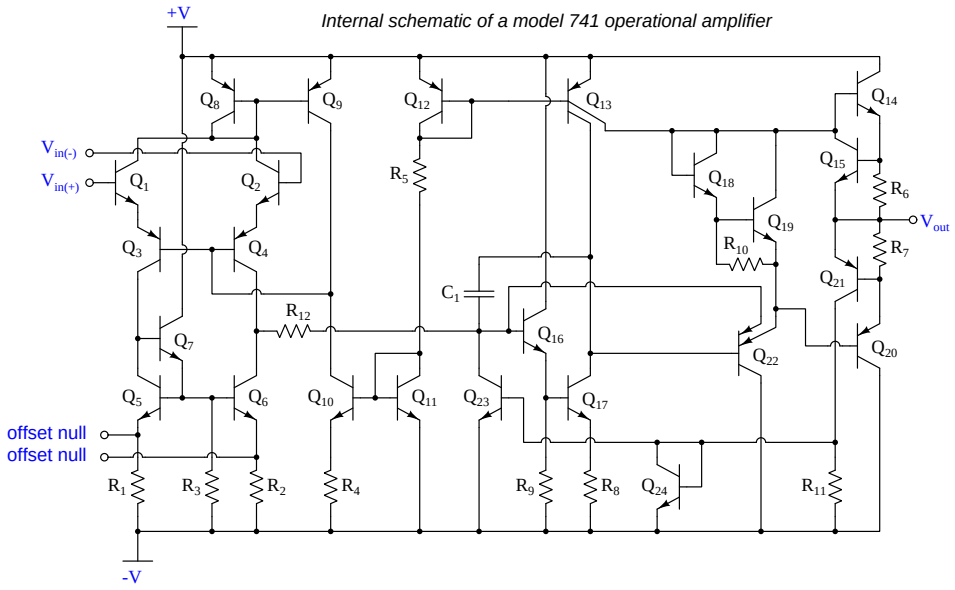


These differential protection diodes attempt to clamp differential voltage to a maximum of 0.7 Volts, which for a typical opamp employing negative feedback is perfectly acceptable because negative feedback should maintain the differential voltage at approximately zero anyway. However, if one attempts to use such a differential-clamped opamp as a *comparator* rather than as an operational amplifier with negative feedback to compare two input potentials differing by more than 0.7 Volts, improper circuit operation will result, with possible damage to the opamp as well!

5.7.3 Input bias current

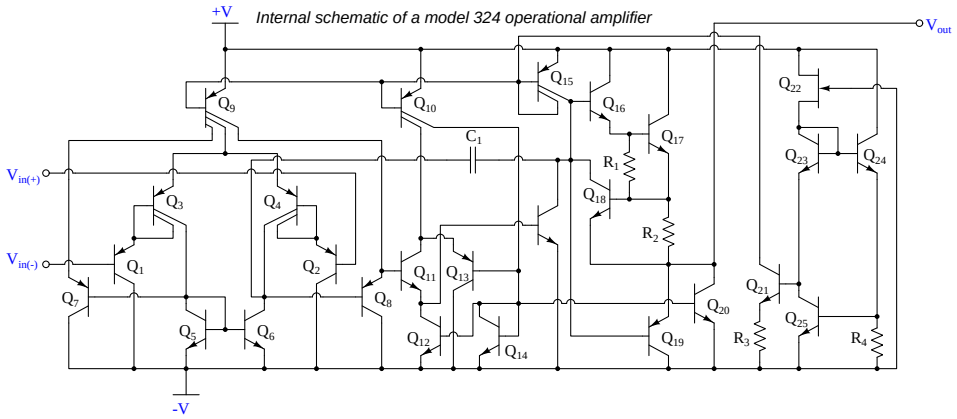
A generally safe assumption for most circuits is that the input terminals of a comparator or operational amplifier carry no current. Like the input of an ideal voltmeter, we assume these inputs have infinite input impedance and therefore pose no load on the sources they sense.

However, this is not entirely true. The input transistor stage of a differential amplifier represents a finite – albeit very large – impedance, and bipolar junction transistors do require *some* amount of base current to function. An inspection of the internal circuitry for the model 741 opamp reveals the source of these currents as well as their directions:



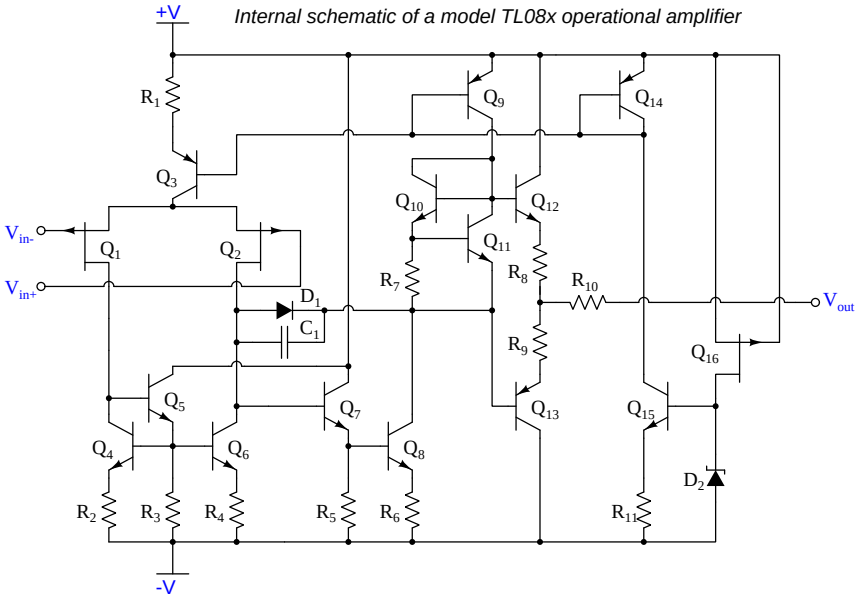
The two input terminals for the 741 connect to the bases of NPN transistors Q_1 and Q_2. In order for these transistors to turn on, current *must* enter the base terminals from outside the opamp. These *bias currents* are quite small – on the order of tens of *nano*Amperes, but they do indeed exist and must find a path through the external circuit.

Another legacy opamp model is the 324. Its internal circuitry is shown below:



Note the two input PNP transistors Q_1 and Q_2. In order for these transistors to turn on, current *must* exit their base terminals and find its way to the negative pole of the power supply through external circuitry.

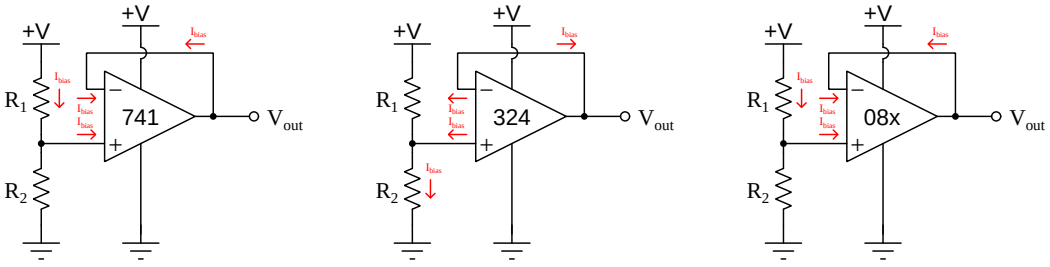
Some comparators and opamps have FET inputs rather than BJT inputs, usually in an attempt to increase input impedance (i.e. decrease bias current). The TL08x opamp series is an example of this design:



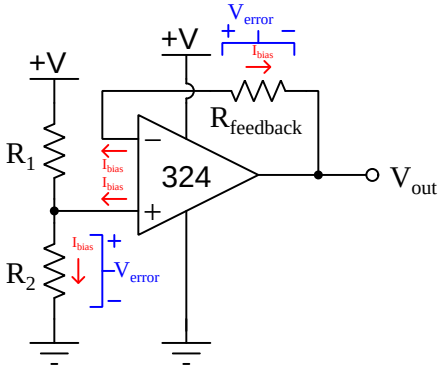
Knowing that JFETs are designed to have their gate-channel PN junctions reverse-biased, we may conclude the two P-channel JFETs Q_1 and Q_2 will exhibit extremely small bias currents *entering* their gate terminals from outside the opamp.

Some opamps utilize MOSFETs for their input transistors in order to maximize input impedance and minimize bias currents, the naturally high impedance of a MOSFET stemming from the layer of metal-oxide insulation separating each MOSFET's gate from the conducting channel beneath. In such opamps the main contribution to input terminal bias current actually comes from reverse-leakage current of *protection diodes* added to the circuit (connecting between each of those input terminals and the DC power rails) to protect the sensitive MOSFET gates from electrostatic discharge, rather than from the transistors themselves.

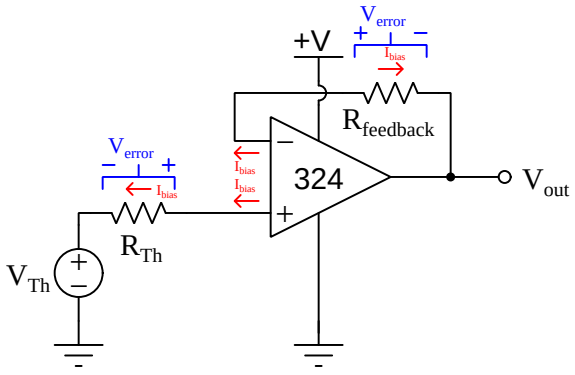
When we connect components to the input terminals of an opamp, we must be sure some pathway exists for these bias currents to flow in the proper directions. Consider the following examples, where a voltage divider provides a voltage signal which each opamp then "buffers" as a voltage follower:



Again, these currents are extremely small in magnitude, and their effects are usually negligible. However, if they must pass through a high impedance, they will cause an undesired voltage drop to develop. A common strategy to mitigate the effects of this voltage drop is to insert an equivalent amount of resistance within the feedback bias current path, assuming the two bias currents will be approximately equal in magnitude, and therefore the *two* unwanted voltage drops should be approximately equal in magnitude but opposite in polarity and so should cancel each other out as common-mode (not differential) voltage at the opamp inputs:



If we regard the voltage divider as a voltage *source* with its own internal Thévenin resistance, we see the bias current must pass through that resistance:



The value of the "source's" Thévenin resistance is equal to the parallel equivalent resistance of R_1 and R_2, and so our feedback resistance needs to be equal to the same:

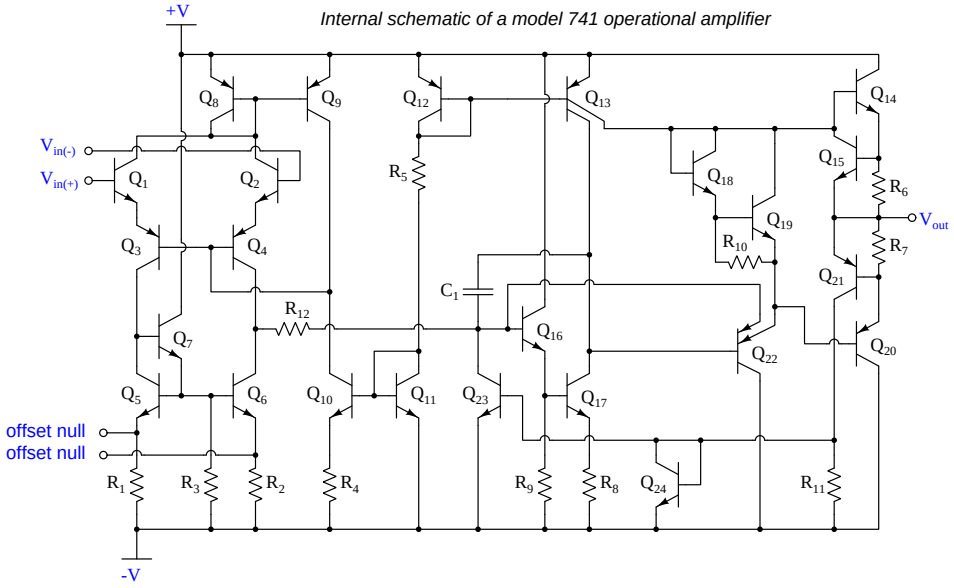
$$R_{feedback} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}$$

It is important to note that the assumption of an opamp's two bias currents being equal is exactly that: an *assumption* and not necessarily reality. Small differences in bias current from one input to the other can exist, and this difference of bias current is called *offset current*.

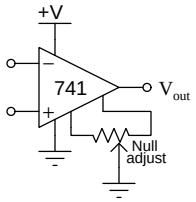
5.7.4 Input offset voltage

If the input terminals of a differential amplifier are shorted together to ensure zero differential input, the amplifier's output should ideally assume a state of zero Volts as well. However, this rarely happens in practice due to asymmetries in the internal circuitry of the amplifier. In practice, a very small amount of differential input voltage is necessary to achieve exactly zero Volts at the output, and this small differential voltage is called the *offset voltage* or *bias voltage* of the amplifier.

Some operational amplifier models provide external terminals for the user to connect a potentiometer, which may be used to "trim" the offset of the amplifier. The legacy 741 opamp is one of those designs, with *offset null* terminals labeled:



Externally, the "nulling" potentiometer connects to these two offset terminals and to the negative power supply terminal:



Essentially, the two portions of the potentiometer become connected in parallel with R_1 and R_2, respectively, inside the opamp. By externally skewing the equivalent resistances of R_1 and R_2, the bias currents through the two halves of the opamp's differential pair input stage may be likewise skewed to achieve the desired negation of offset voltage.

5.7.5 Frequency compensation

A problem common to high-gain amplifier circuits is undesired *oscillation*, and the root of this problem is *positive feedback*: where some of the amplifier's output signal becomes "fed back" to its input with a 360^o phase shift, so that it continually drives itself into oscillation. This principle finds constructive application in sinusoidal oscillator circuits, but for non-oscillator circuits it can be a serious problem.

Most operational amplifier circuits utilize *negative* feedback (i.e. 180^o phase shift) rather than positive feedback, and so it might seem at first oscillations would not be possible in these circuits. However, the 360^o of phase shift necessary for self-sustaining oscillations may occur in a negative-feedback circuit if a total of 180^o *additional* phase shift occurs elsewhere in the opamp circuit. One operating scenario inviting additional phase shift is when the opamp's output must drive a capacitive load. At high enough frequencies, the load's capacitance and stray capacitances in the circuit may conspire to provide the additional 180^o of phase shift necessary to make the total phase shift 360^o.

The criteria necessary for self-oscillation is signal feedback with 360^o of phase shift *and* a total signal gain of at least unity (1). If the 360^o of phase shift cannot be avoided, an effective solution to the problem of unintentional oscillation is to intentionally reduce the opamp's open-loop gain as frequency increases, so that at the frequency necessary to achieve 360^o phase shift the fed-back signal has a voltage gain less than one.

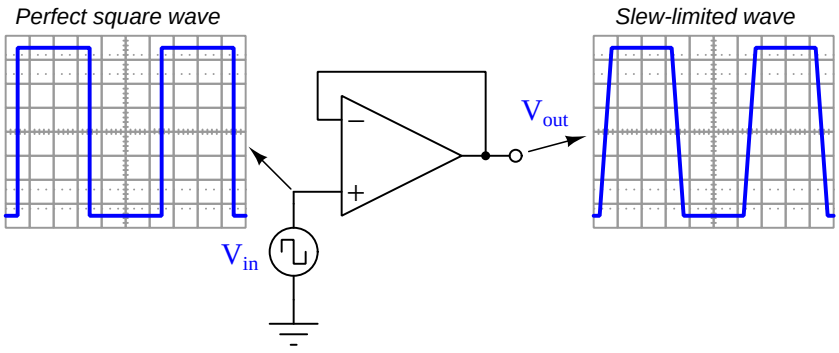
Many operational amplifiers are *internally compensated* for this by the intentional connection of a capacitor between amplification stages. The internal schematic diagrams of three models of opamp shown in previous subsections all contain compensation capacitors. They are easy to spot, being the *only* capacitor in the entire diagram!

5.7.6 Gain-bandwidth product

Operational amplifier open-loop voltage gain is very high, but degrades for multiple reasons as signal frequency increases. The mathematical product of amplifier gain and signal frequency is expressed as the amplifier's *Gain Bandwidth Product*, or *GBWP*.

5.7.7 Slew rate

The maximum rate at which the output voltage of a comparator or opamp is able to rise or fall is called the *slew rate*. If we drive a simple "voltage follower" opamp circuit with a perfect square-wave signal having instantaneous rise and fall times, what we will see on the output is a slew-limited waveform with definite rise and fall *slopes*:



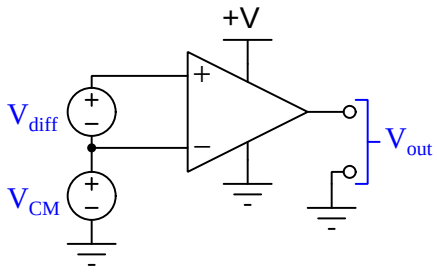
Slope is always measured as a *rise over run*, so for these voltage waveforms we may represent their rising- and falling-edge slopes as a rates-of-change of voltage over time ($\frac{dV}{dt}$), the input waveform having a *infinite* rates of change, and the output waveform having *finite* rates of change. The reason for an opamp's slew rate limitations is based on its internal current limitations and capacitances. Recall the relationship between voltage and current for any capacitance:

$$I_C = C\frac{dV_C}{dt}$$

If a current source drives a constant current through a capacitance, that voltage across that capacitance will either rise or fall (depending on current direction) at a constant *rate*. An opamp's internal transistors are current-limiting devices by nature, and when this limited current energizes or de-energizes capacitance the result is a linearly-sloping waveform rather than an instantaneous jump. A major source of capacitance for any opamp is the *frequency-compensating* capacitance previously discussed, but any capacitance connected to the opamp as part of the load of course also contributes to a limited slew rate. This is why opamp datasheets always specify load capacitance when declaring maximum slew rate (e.g. the model 741 opamp is typically limited to 0.5 Volts per microsecond, or 500,000 Volts per second, while driving a load having no more than 100 picoFarads of capacitance).

5.7.8 Common-mode rejection

Operational amplifiers are *differential* amplifiers, and as such they are only supposed to respond to *differences* in voltage between their two input terminals. Ideally, any common-mode[12] voltage is ignored (i.e. *rejected*) by the amplifier:



In other words, V_{out} for an ideal opamp is strictly a function of V_{diff} and is entirely independent of V_{CM}. However, real opamps are not ideal, and as such their output voltages will be slightly affected by V_{CM}. We may express the relationship between V_{out} and V_{CM} as a *gain*, just as we do for differential voltage:

$$A_{CM} = \frac{\Delta V_{out}}{\Delta V_{CM}} \qquad A_{diff} = \frac{\Delta V_{out}}{\Delta V_{diff}}$$

Differential voltage gain (A_{diff}) is typically referred to as open-loop voltage gain (A_{OL}) and for most operational amplifiers is an extremely large number. Common-mode voltage gain (A_{CM}) by contrast is much smaller (ideally being zero).

The degree to which a differential amplifier rejects a common-mode signal is typically gauged in comparison to its differential voltage gain. The ratio of A_{diff} to A_{CM} is called the *common-mode rejection ratio*, or *CMRR*:

$$\text{CMRR} = \frac{A_{diff}}{A_{CM}}$$

CMRR is more often found expressed as a decibel figure, so:

$$\text{CMRR(dB)} = 20 \log \left(\frac{A_{diff}}{A_{CM}} \right)$$

[12] Here, the term "common-mode" refers to a quantity shared in common by both input terminals of the amplifier. This is distinct from "differential" which by definition means something *different* or *distinct* between the input terminals.

Chapter 6

Programming References

A powerful tool for mathematical modeling is text-based *computer programming*. This is where you type coded commands in text form which the computer is able to interpret. Many different text-based languages exist for this purpose, but we will focus here on just two of them, *C++* and *Python*.

6.1 Programming in C++

One of the more popular text-based computer programming languages is called *C++*. This is a *compiled* language, which means you must create a plain-text file containing C++ code using a program called a *text editor*, then execute a software application called a *compiler* to translate your "source code" into instructions directly understandable to the computer. Here is an example of "source code" for a very simple C++ program intended to perform some basic arithmetic operations and print the results to the computer's console:

```
#include <iostream>
using namespace std;

int main (void)
{
  float x, y;

  x = 200;
  y = -560.5;

  cout << "This simple program performs basic arithmetic on" << endl;
  cout << "the two numbers " << x << " and " << y << " and then" << endl;
  cout << "displays the results on the computer's console." << endl;

  cout << endl;

  cout << "Sum = " << x + y << endl;
  cout << "Difference = " << x - y << endl;
  cout << "Product = " << x * y << endl;
  cout << "Quotient of " << x / y << endl;

  return 0;
}
```

Computer languages such as C++ are designed to make sense when read by human programmers. The general order of execution is left-to-right, top-to-bottom just the same as reading any text document written in English. Blank lines, indentation, and other "whitespace" is largely irrelevant in C++ code, and is included only to make the code more pleasing[1] to view.

[1] Although not included in this example, *comments* preceded by double-forward slash characters (`//`) may be added to source code as well to provide explanations of what the code is supposed to do, for the benefit of anyone reading it. The compiler application will ignore all comments.

Let's examine the C++ source code to explain what it means:

- `#include <iostream>` and `using namespace std;` are set-up instructions to the compiler giving it some context in which to interpret your code. The code specific to your task is located between the brace symbols ({ and }, often referred to as "curly-braces").

- `int main (void)` labels the "Main" function for the computer: the instructions within this function (lying between the { and } symbols) it will be commanded to execute. Every complete C++ program contains a `main` function at minimum, and often additional functions as well, but the `main` function is where execution always begins. The `int` declares this function will return an *integer* number value when complete, which helps to explain the purpose of the `return 0;` statement at the end of the `main` function: providing a numerical value of zero at the program's completion as promised by `int`. This returned value is rather incidental to our purpose here, but it is fairly standard practice in C++ programming.

- Grouping symbols such as (parentheses) and {braces} abound in C, C++, and other languages (e.g. Java). Parentheses typically group data to be processed by a function, called *arguments* to that function. Braces surround lines of executable code belonging to a particular function.

- The `float` declaration reserves places in the computer's memory for two *floating-point* variables, in this case the variables' names being `x` and `y`. In most text-based programming languages, variables may be named by single letters or by combinations of letters (e.g. `xyz` would be a single variable).

- The next two lines assign numerical values to the two variables. Note how each line terminates with a semicolon character (`;`) and how this pattern holds true for most of the lines in this program. In C++ semicolons are analogous to periods at the ends of English sentences. This demarcation of each line's end is necessary because C++ ignores whitespace on the page and doesn't "know" otherwise where one line ends and another begins.

- All the other instructions take the form of a `cout` command which prints characters to the "standard output" stream of the computer, which in this case will be text displayed on the console. The double-less-than symbols (`<<`) show data being sent *toward* the `cout` command. Note how verbatim text is enclosed in quotation marks, while variables such as `x` or mathematical expressions such as `x - y` are not enclosed in quotations because we want the computer to display the numerical values represented, not the literal text.

- Standard arithmetic operations (add, subtract, multiply, divide) are represented as `+`, `-`, `*`, and `/`, respectively.

- The `endl` found at the end of every `cout` statement marks the end of a line of text printed to the computer's console display. If not for these `endl` inclusions, the displayed text would resemble a run-on sentence rather than a paragraph. Note the `cout << endl;` line, which does nothing but create a blank line on the screen, for no reason other than esthetics.

After saving this *source code* text to a file with its own name (e.g. `myprogram.cpp`), you would then *compile* the source code into an *executable* file which the computer may then run. If you are using a console-based compiler such as *GCC* (very popular within variants of the Unix operating system[2], such as Linux and Apple's OS X), you would type the following command and press the Enter key:

```
g++ -o myprogram.exe myprogram.cpp
```

This command instructs the `GCC` compiler to take your source code (`myprogram.cpp`) and create with it an executable file named `myprogram.exe`. Simply typing `./myprogram.exe` at the command-line will then execute your program:

```
./myprogram.exe
```

If you are using a graphic-based C++ development system such as Microsoft Visual Studio[3], you may simply create a new console application "project" using this software, then paste or type your code into the example template appearing in the editor window, and finally run your application to test its output.

As this program runs, it displays the following text to the console:

```
This simple program performs basic arithmetic on
the two numbers 200 and -560.5 and then
displays the results on the computer's console.

Sum = -360.5
Difference = 760.5
Product = -112100
Quotient of -0.356824
```

As crude as this example program is, it serves the purpose of showing how easy it is to write and execute simple programs in a computer using the C++ language. As you encounter C++ example programs (shown as source code) in any of these modules, feel free to directly copy-and-paste the source code text into a text editor's screen, then follow the rest of the instructions given here (i.e. save to a file, compile, and finally run your program). You will find that it is generally easier to

[2] A very functional option for users of Microsoft Windows is called *Cygwin*, which provides a Unix-like console environment complete with all the customary utility applications such as GCC!

[3] Using Microsoft Visual Studio community version 2017 at the time of this writing to test this example, here are the steps I needed to follow in order to successfully compile and run a simple program such as this: (1) Start up Visual Studio and select the option to create a New Project; (2) Select the Windows Console Application template, as this will perform necessary set-up steps to generate a console-based program which will save you time and effort as well as avoid simple errors of omission; (3) When the editing screen appears, type or paste the C++ code within the `main()` function provided in the template, deleting the "Hello World" `cout` line that came with the template; (4) Type or paste any preprocessor directives (e.g. `#include` statements, `namespace` statements) necessary for your code that did not come with the template; (5) Lastly, under the Debug drop-down menu choose either Start Debugging (F5 hot-key) or Start Without Debugging (Ctrl-F5 hotkeys) to compile ("Build") and run your new program. Upon execution a console window will appear showing the output of your program.

learn computer programming by closely examining others' example programs and modifying them than it is to write your own programs starting from a blank screen.

6.2 Programming in Python

Another text-based computer programming language called *Python* allows you to type instructions at a terminal prompt and receive immediate results without having to compile that code. This is because Python is an *interpreted* language: a software application called an *interpreter* reads your source code, translates it into computer-understandable instructions, and then executes those instructions in one step.

The following shows what happens on my personal computer when I start up the Python interpreter on my personal computer, by typing `python3`[4] and pressing the Enter key:

```
Python 3.7.2 (default, Feb 19 2019, 18:15:18)
[GCC 4.1.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The `>>>` symbols represent the prompt within the Python interpreter "shell", signifying readiness to accept Python commands entered by the user.

Shown here is an example of the same arithmetic operations performed on the same quantities, using a Python interpreter. All lines shown preceded by the `>>>` prompt are entries typed by the human programmer, and all lines shown without the `>>>` prompt are responses from the Python interpreter software:

```
>>> x = 200
>>> y = -560.5
>>> x + y
-360.5
>>> x - y
760.5
>>> x * y
-112100.0
>>> x / y
-0.35682426404995538
>>> quit()
```

[4] Using version 3 of Python, which is the latest at the time of this writing.

More advanced mathematical functions are accessible in Python by first entering the line `from math import *` which "imports" these functions from Python's math *library* (with functions identical to those available for the C programming language, and included on any computer with Python installed). Some examples show some of these functions in use, demonstrating how the Python interpreter may be used as a scientific calculator:

```
>>> from math import *
>>> sin(30.0)
-0.98803162409286183
>>> sin(radians(30.0))
0.49999999999999994
>>> pow(2.0, 5.0)
32.0
>>> log10(10000.0)
4.0
>>> e
2.7182818284590451
>>> pi
3.1415926535897931
>>> log(pow(e,6.0))
6.0
>>> asin(0.7071068)
0.78539819000368838
>>> degrees(asin(0.7071068))
45.000001524425265
>>> quit()
```

Note how trigonometric functions assume angles expressed in *radians* rather than *degrees*, and how Python provides convenient functions for translating between the two. Logarithms assume a base of e unless otherwise stated (e.g. the `log10` function for common logarithms).

The interpreted (versus compiled) nature of Python, as well as its relatively simple syntax, makes it a good choice as a person's first programming language. For complex applications, interpreted languages such as Python execute slower than compiled languages such as C++, but for the very simple examples used in these learning modules speed is not a concern.

Another Python math library is `cmath`, giving Python the ability to perform arithmetic on complex numbers. This is very useful for AC circuit analysis using *phasors*[5] as shown in the following example. Here we see Python's interpreter used as a scientific calculator to show series and parallel impedances of a resistor, capacitor, and inductor in a 60 Hz AC circuit:

```
>>> from math import *
>>> from cmath import *
>>> r = complex(400,0)
>>> f = 60.0
>>> xc = 1/(2 * pi * f * 4.7e-6)
>>> zc = complex(0,-xc)
>>> xl = 2 * pi * f * 1.0
>>> zl = complex(0,xl)
>>> r + zc + zl
(400-187.38811239154882j)
>>> 1/(1/r + 1/zc + 1/zl)
(355.837695813625+125.35793777619385j)
>>> polar(r + zc + zl)
(441.717448903332, -0.4381072059213295)
>>> abs(r + zc + zl)
441.717448903332
>>> phase(r + zc + zl)
-0.4381072059213295
>>> degrees(phase(r + zc + zl))
-25.10169387356105
```

When entering a value in rectangular form, we use the `complex()` function where the arguments are the real and imaginary quantities, respectively. If we had opted to enter the impedance values in polar form, we would have used the `rect()` function where the first argument is the magnitude and the second argument is the angle in radians. For example, we could have set the capacitor's impedance (`zc`) as $X_C \angle -90^o$ with the command `zc = rect(xc,radians(-90))` rather than with the command `zc = complex(0,-xc)` and it would have worked the same.

Note how Python defaults to rectangular form for complex quantities. Here we defined a 400 Ohm resistance as a complex value in rectangular form (400 +j0 Ω), then computed capacitive and inductive reactances at 60 Hz and defined each of those as complex (phasor) values ($0 - jX_c$ Ω and $0 + jX_l$ Ω, respectively). After that we computed total impedance in series, then total impedance in parallel. Polar-form representation was then shown for the series impedance (441.717 Ω $\angle$ -25.102^o). Note the use of different functions to show the polar-form series impedance value: `polar()` takes the complex quantity and returns its polar magnitude and phase angle in *radians*; `abs()` returns just the polar magnitude; `phase()` returns just the polar angle, once again in radians. To find the polar phase angle in degrees, we nest the `degrees()` and `phase()` functions together.

The utility of Python's interpreter environment as a scientific calculator should be clear from these examples. Not only does it offer a powerful array of mathematical functions, but also unlimited

[5] A "phasor" is a voltage, current, or impedance represented as a complex number, either in rectangular or polar form.

assignment of variables as well as a convenient text record[6] of all calculations performed which may be easily copied and pasted into a text document for archival.

It is also possible to save a set of Python commands to a text file using a text editor application, and then instruct the Python interpreter to execute it at once rather than having to type it line-by-line in the interpreter's shell. For example, consider the following Python program, saved under the filename `myprogram.py`:

```
x = 200
y = -560.5

print("Sum")
print(x + y)

print("Difference")
print(x - y)

print("Product")
print(x * y)

print("Quotient")
print(x / y)
```

As with C++, the interpreter will read this source code from left-to-right, top-to-bottom, just the same as you or I would read a document written in English. Interestingly, whitespace *is* significant in the Python language (unlike C++), but this simple example program makes no use of that.

To execute this Python program, I would need to type `python myprogram.py` and then press the Enter key at my computer console's prompt, at which point it would display the following result:

```
Sum
-360.5
Difference
760.5
Product
-112100.0
Quotient
-0.35682426405
```

As you can see, syntax within the Python programming language is simpler than C++, which is one reason why it is often a preferred language for beginning programmers.

[6] Like many command-line computing environments, Python's interpreter supports "up-arrow" recall of previous entries. This allows quick recall of previously typed commands for editing and re-evaluation.

If you are interested in learning more about computer programming in *any* language, you will find a wide variety of books and free tutorials available on those subjects. Otherwise, feel free to learn by the examples presented in these modules.

6.3 Butterworth passive filter designer using C++

This simple program calculates the capacitor and inductor values necessary to form a passive ladder-topology filter network exhibiting a Butterworth polynomial characteristic:

```
#include <iostream>
#include <cmath>
using namespace std;

// The following array stores coefficients for component values based on
// the assumption of 1 rad/sec cutoff and 1 Ohm input/load impedances,
// from 1st order to 9th order filter networks.
float coeff[10][10] =
{{0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000}, // 0th
 {0.000, 2.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000}, // 1st
 {0.000, 1.414, 1.414, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000}, // 2nd
 {0.000, 1.000, 2.000, 1.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000}, // 3rd
 {0.000, 0.765, 1.848, 1.848, 0.765, 0.000, 0.000, 0.000, 0.000, 0.000}, // 4th
 {0.000, 0.618, 1.618, 2.000, 1.618, 0.618, 0.000, 0.000, 0.000, 0.000}, // 5th
 {0.000, 0.518, 1.414, 1.932, 1.932, 1.414, 0.518, 0.000, 0.000, 0.000}, // 6th
 {0.000, 0.445, 1.247, 1.802, 2.000, 1.802, 1.247, 0.445, 0.000, 0.000}, // 7th
 {0.000, 0.390, 1.111, 1.663, 1.962, 1.962, 1.663, 1.111, 0.390, 0.000}, // 8th
 {0.000, 0.347, 1.000, 1.532, 1.879, 2.000, 1.879, 1.532, 1.000, 0.347}}; // 9th

int main (void)
{
  int n, lowpass, count;
  float f, z, x[10];

  // Gathering input from the user . . .
  cout << "Enter desired order of filter = ";  cin >> n;
  cout << "Enter 1 for lowpass, 0 for highpass = ";  cin >> lowpass;
  cout << "Enter desired cutoff frequency (Hz) = ";  cin >> f;
  cout << "Enter desired input impedance (Ohms) = ";  cin >> z;

  // Ensuring a minimum filter order of 1
  if (n < 1)
    n == 1;

  // Ensuring a maximum filter order of 9
  if (n > 9)
    n == 9;
```

```
  if (lowpass == 1) // Low-pass filter design
  {
    for (count = 1 ; count <= n; ++count)
    {
      x[count] = coeff[n][count];

      if (count % 2 != 0) // Scaling capacitor values
      {
        x[count] = x[count] / z / (f * 2 * M_PI);
        cout << "Capacitor C" << count << " = " << x[count] << " Farads" << endl;
      }

      else // Scaling inductor values
      {
        x[count] = x[count] * z / (f * 2 * M_PI);
        cout << "Inductor L" << count << " = " << x[count] << " Henrys" << endl;
      }
    }
  }

  else // High-pass filter design
  {
    for (count = 1 ; count <= n; ++count)
    {
      x[count] = 1/(coeff[n][count]);

      if (count % 2 != 0) // Scaling inductor values
      {
        x[count] = x[count] * z / (f * 2 * M_PI);
        cout << "Inductor L" << count << " = " << x[count] << " Henrys" << endl;
      }

      else // Scaling capacitor values
      {
        x[count] = x[count] / z / (f * 2 * M_PI);
        cout << "Capacitor C" << count << " = " << x[count] << " Farads" << endl;
      }
    }
  }

  return 0;
}
```

Here we will run this program to show how it works to calculate component values for the 5th order high-pass Butterworth filter described in the Tutorial, with a desired cutoff frequency of 300 kHz and a desired input/load impedance of 75 Ohms:

```
Enter desired order of filter = 5
Enter 1 for lowpass, 0 for highpass = 0
Enter desired cutoff frequency (Hz) = 300e3
Enter desired input impedance (Ohms) = 75
Inductor L1 = 6.43831e-05 Henrys
Capacitor C2 = 4.37179e-09 Farads
Inductor L3 = 1.98944e-05 Henrys
Capacitor C4 = 4.37179e-09 Farads
Inductor L5 = 6.43831e-05 Henrys
```

Just as shown in the Tutorial, this program properly computes the values for the components as follows:

- $L_1 = 64.3831$ μH
- $C_2 = 4.37179$ nF
- $L_3 = 19.8944$ μH
- $C_4 = 4.37179$ nF
- $L_5 = 64.3831$ μH

Note how the program expresses all values in scientific notation, with some decimal-point shifting necessary to re-express these values using standard metric prefixes. Note also how the program is able to accept values input in power-of-ten notation, with `300e3` representing 300 kiloHertz.

Chapter 7

Questions

This learning module, along with all others in the ModEL collection, is designed to be used in an inverted instructional environment where students independently read[1] the tutorials and attempt to answer questions on their own *prior* to the instructor's interaction with them. In place of lecture[2], the instructor engages with students in Socratic-style dialogue, probing and challenging their understanding of the subject matter through inquiry.

Answers are not provided for questions within this chapter, and this is by design. Solved problems may be found in the Tutorial and Derivation chapters, instead. The goal here is *independence*, and this requires students to be challenged in ways where others cannot think for them. Remember that you always have the tools of *experimentation* and *computer simulation* (e.g. SPICE) to explore concepts!

The following lists contain ideas for Socratic-style questions and challenges. Upon inspection, one will notice a strong theme of *metacognition* within these statements: they are designed to foster a regular habit of examining one's own thoughts as a means toward clearer thinking. As such these sample questions are useful both for instructor-led discussions as well as for self-study.

[1] Technical reading is an essential academic skill for any technical practitioner to possess for the simple reason that the most comprehensive, accurate, and useful information to be found for developing technical competence is in textual form. Technical careers in general are characterized by the need for continuous learning to remain current with standards and technology, and therefore any technical practitioner who cannot read well is handicapped in their professional development. An excellent resource for educators on improving students' reading prowess through intentional effort and strategy is the book textitReading For Understanding – How Reading Apprenticeship Improves Disciplinary Learning in Secondary and College Classrooms by Ruth Schoenbach, Cynthia Greenleaf, and Lynn Murphy.

[2] Lecture is popular as a teaching method because it is easy to implement: any reasonably articulate subject matter expert can talk to students, even with little preparation. However, it is also quite problematic. A good lecture always makes complicated concepts seem easier than they are, which is bad for students because it instills a false sense of confidence in their own understanding; reading and re-articulation requires more cognitive effort and serves to verify comprehension. A culture of teaching-by-lecture fosters a debilitating dependence upon direct personal instruction, whereas the challenges of modern life demand independent and critical thought made possible only by gathering information and perspectives from afar. Information presented in a lecture is ephemeral, easily lost to failures of memory and dictation; text is forever, and may be referenced at any time.

General challenges following tutorial reading

- Summarize as much of the text as you can in one paragraph of your own words. A helpful strategy is to explain ideas as you would for an intelligent child: as simple as you can without compromising too much accuracy.
- Simplify a particular section of the text, for example a paragraph or even a single sentence, so as to capture the same fundamental idea in fewer words.
- Where did the text make the most sense to you? What was it about the text's presentation that made it clear?
- Identify where it might be easy for someone to misunderstand the text, and explain why you think it could be confusing.
- Identify any new concept(s) presented in the text, and explain in your own words.
- Identify any familiar concept(s) such as physical laws or principles applied or referenced in the text.
- Devise a proof of concept experiment demonstrating an important principle, physical law, or technical innovation represented in the text.
- Devise an experiment to disprove a plausible misconception.
- Did the text reveal any misconceptions you might have harbored? If so, describe the misconception(s) and the reason(s) why you now know them to be incorrect.
- Describe any useful problem-solving strategies applied in the text.
- Devise a question of your own to challenge a reader's comprehension of the text.

General follow-up challenges for assigned problems

- Identify where any fundamental laws or principles apply to the solution of this problem, especially before applying any mathematical techniques.
- Devise a thought experiment to explore the characteristics of the problem scenario, applying known laws and principles to mentally model its behavior.
- Describe in detail your own strategy for solving this problem. How did you identify and organized the given information? Did you sketch any diagrams to help frame the problem?
- Is there more than one way to solve this problem? Which method seems best to you?
- Show the work you did in solving this problem, even if the solution is incomplete or incorrect.
- What would you say was the most challenging part of this problem, and why was it so?
- Was any important information missing from the problem which you had to research or recall?
- Was there any extraneous information presented within this problem? If so, what was it and why did it not matter?
- Examine someone else's solution to identify where they applied fundamental laws or principles.
- Simplify the problem from its given form and show how to solve this simpler version of it. Examples include eliminating certain variables or conditions, altering values to simpler (usually whole) numbers, applying a limiting case (i.e. altering a variable to some extreme or ultimate value).
- For quantitative problems, identify the real-world meaning of all intermediate calculations: their units of measurement, where they fit into the scenario at hand. Annotate any diagrams or illustrations with these calculated values.
- For quantitative problems, try approaching it qualitatively instead, thinking in terms of "increase" and "decrease" rather than definite values.
- For qualitative problems, try approaching it quantitatively instead, proposing simple numerical values for the variables.
- Were there any assumptions you made while solving this problem? Would your solution change if one of those assumptions were altered?
- Identify where it would be easy for someone to go astray in attempting to solve this problem.
- Formulate your own problem based on what you learned solving this one.

General follow-up challenges for experiments or projects

- In what way(s) was this experiment or project easy to complete?
- Identify some of the challenges you faced in completing this experiment or project.

- Show how thorough documentation assisted in the completion of this experiment or project.
- Which fundamental laws or principles are key to this system's function?
- Identify any way(s) in which one might obtain false or otherwise misleading measurements from test equipment in this system.
- What will happen if (component X) fails (open/shorted/etc.)?
- What would have to occur to make this system unsafe?

7.1 Conceptual reasoning

These questions are designed to stimulate your analytic and synthetic thinking[3]. In a Socratic discussion with your instructor, the goal is for these questions to prompt an extended dialogue where assumptions are revealed, conclusions are tested, and understanding is sharpened. Your instructor may also pose additional questions based on those assigned, in order to further probe and refine your conceptual understanding.

Questions that follow are presented to challenge and probe your understanding of various concepts presented in the tutorial. These questions are intended to serve as a guide for the Socratic dialogue between yourself and the instructor. Your instructor's task is to ensure you have a sound grasp of these concepts, and the questions contained in this document are merely a means to this end. Your instructor may, at his or her discretion, alter or substitute questions for the benefit of tailoring the discussion to each student's needs. The only absolute requirement is that each student is challenged and assessed at a level equal to or greater than that represented by the documented questions.

It is far more important that you convey your *reasoning* than it is to simply convey a correct answer. For this reason, you should refrain from researching other information sources to answer questions. What matters here is that *you* are doing the thinking. If the answer is incorrect, your instructor will work with you to correct it through proper reasoning. A correct answer without an adequate explanation of how you derived that answer is unacceptable, as it does not aid the learning or assessment process.

You will note a conspicuous lack of answers given for these conceptual questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your conceptual answers, where applicable, is to use circuit simulation software to explore the effects of changes made to circuits. For example, if one of these conceptual questions challenges you to predict the effects of altering some component parameter in a circuit, you may check the validity of your work by simulating that same parameter change within software and seeing if the results agree.

[3] *Analytical* thinking involves the "disassembly" of an idea into its constituent parts, analogous to dissection. *Synthetic* thinking involves the "assembly" of a new idea comprised of multiple concepts, analogous to construction. Both activities are high-level cognitive skills, extremely important for effective problem-solving, necessitating frequent challenge and regular practice to fully develop.

7.1.1 Reading outline and reflections

"Reading maketh a full man; conference a ready man; and writing an exact man" – Francis Bacon

Francis Bacon's advice is a blueprint for effective education: reading provides the learner with knowledge, writing focuses the learner's thoughts, and critical dialogue equips the learner to confidently communicate and apply their learning. Independent acquisition and application of knowledge is a powerful skill, well worth the effort to cultivate. To this end, students should read these educational resources closely, journal their own reflections on the reading, and discuss in detail their findings with classmates and instructor(s). You should be able to do all of the following after reading any instructional text:

☑ Briefly SUMMARIZE THE TEXT in the form of a journal entry documenting your learning as you progress through the course of study. Share this summary in dialogue with your classmates and instructor. Journaling is an excellent self-test of thorough reading because you cannot clearly express what you have not read or did not comprehend.

☑ Demonstrate ACTIVE READING STRATEGIES, including verbalizing your impressions as you read, simplifying long passages to convey the same ideas using fewer words, annotating text and illustrations with your own interpretations, working through mathematical examples shown in the text, cross-referencing passages with relevant illustrations and/or other passages, identifying problem-solving strategies applied by the author, etc. Technical reading is a special case of problem-solving, and so these strategies work precisely because they help solve any problem: paying attention to your own thoughts (metacognition), eliminating unnecessary complexities, identifying what makes sense, paying close attention to details, drawing connections between separated facts, and noting the successful strategies of others.

☑ Identify IMPORTANT THEMES, especially GENERAL LAWS and PRINCIPLES, expounded in the text and express them in the simplest of terms as though you were teaching an intelligent child. This emphasizes connections between related topics and develops your ability to communicate complex ideas to anyone.

☑ Form YOUR OWN QUESTIONS based on the reading, and then pose them to your instructor and classmates for their consideration. Anticipate both correct and incorrect answers, the incorrect answer(s) assuming one or more plausible misconceptions. This helps you view the subject from different perspectives to grasp it more fully.

☑ Devise EXPERIMENTS to test claims presented in the reading, or to disprove misconceptions. Predict possible outcomes of these experiments, and evaluate their meanings: what result(s) would confirm, and what would constitute disproof? Running mental simulations and evaluating results is essential to scientific and diagnostic reasoning.

☑ Specifically identify any points you found CONFUSING. The reason for doing this is to help diagnose misconceptions and overcome barriers to learning.

7.1.2 Foundational concepts

Correct analysis and diagnosis of electric circuits begins with a proper understanding of some basic concepts. The following is a list of some important concepts referenced in this module's full tutorial. Define each of them in your own words, and be prepared to illustrate each of these concepts with a description of a practical example and/or a live demonstration.

Energy

Conservation of Energy

Thought experiments as a problem-solving strategy

Limiting cases as a problem-solving strategy

Annotating diagrams as a problem-solving strategy

Graphing as a problem-solving strategy

Capacitance

Inductance

Reactance

Open

Short

Voltage divider

Phasor

Phase angle

Bode plot

Cutoff

Passband

Stopband

Bandwidth

Roll-off

Frequency domain

Sinusoidal decomposition (i.e. Fourier's Theorem)

Time domain

Transfer function

Zero

Pole

Order

Angular velocity

Neper velocity

"Ohm's Law" for capacitance

"Ohm's Law" for inductance

Resonance

Parasitic effect

Normalization

Monotonicity

7.1.3 Single-pole filters

Why are only filters having more than one pole capable of classification as either Butterworth, Bessel, Chebyshev, or Elliptic?

Challenges

- Why might we wish to choose a Butterworth, Bessel, Chebyshev, or Elliptic filter characteristic for a particular application?

7.1.4 Filter component value scaling

Tables of component values are extremely helpful when designing passive filter networks, but the values as given are never precisely what you will need in your specific filter's design. Rather, these tabulated values must be *scaled* into values suitable for your application.

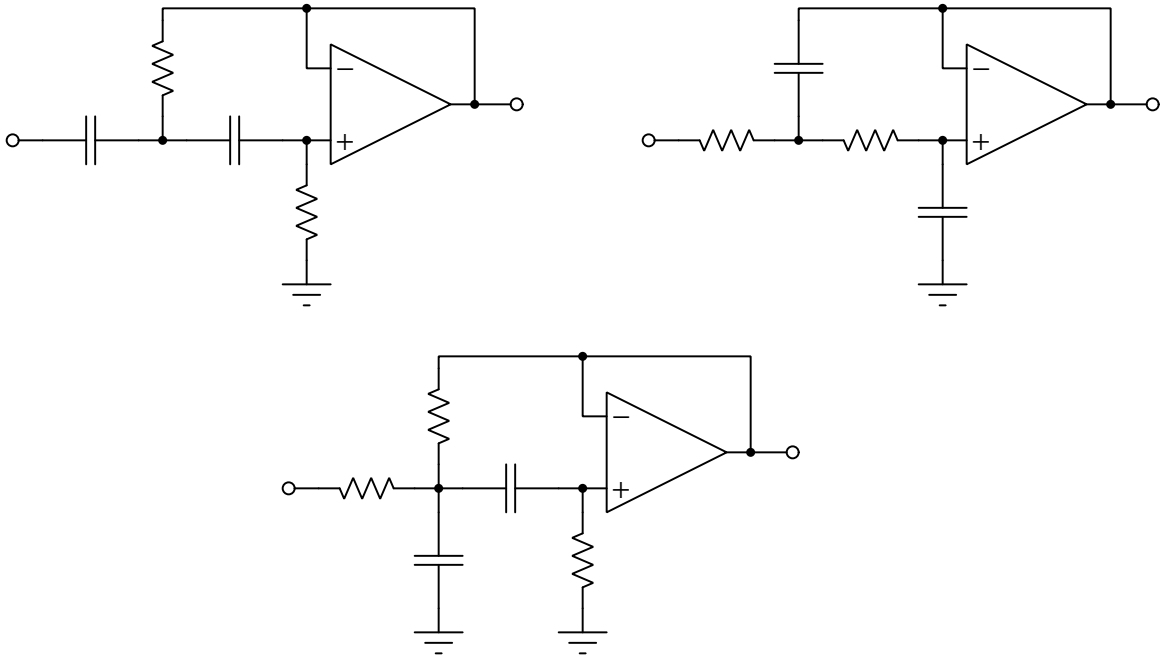
Explain why this is, and what factor(s) influence this scaling procedure.

Challenges

- Why do multi-order passive filters typically contain both capacitors and inductors?

7.1.5 Limiting-case analysis of Sallen-Key filter networks

Qualitatively analyze the following active filter networks at very low and at very high frequency values, low and high enough respectively to make the reactive components exhibit extreme impedance values (i.e. either zero Ohms or infinite Ohms), and from these analyses identify and explain the filtering characteristic of each network:



Challenges

- Why do active filter networks typically use only capacitors and no inductors?
- Is it possible for a Sallen-Key filter network to provide different filtering curves such as Butterworth versus Chebyshev? If so, how?

7.2 Quantitative reasoning

These questions are designed to stimulate your computational thinking. In a Socratic discussion with your instructor, the goal is for these questions to reveal your mathematical approach(es) to problem-solving so that good technique and sound reasoning may be reinforced. Your instructor may also pose additional questions based on those assigned, in order to observe your problem-solving firsthand.

Mental arithmetic and estimations are strongly encouraged for all calculations, because without these abilities you will be unable to readily detect errors caused by calculator misuse (e.g. keystroke errors).

You will note a conspicuous lack of answers given for these quantitative questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. My advice is to use circuit simulation software such as SPICE to check the correctness of quantitative answers. Refer to those learning modules within this collection focusing on SPICE to see worked examples which you may use directly as practice problems for your own study, and/or as templates you may modify to run your own analyses and generate your own practice problems.

Completely worked example problems found in the Tutorial may also serve as "test cases[4]" for gaining proficiency in the use of circuit simulation software, and then once that proficiency is gained you will never need to rely[5] on an answer key!

[4] In other words, set up the circuit simulation software to analyze the same circuit examples found in the Tutorial. If the simulated results match the answers shown in the Tutorial, it confirms the simulation has properly run. If the simulated results disagree with the Tutorial's answers, something has been set up incorrectly in the simulation software. Using every Tutorial as practice in this way will quickly develop proficiency in the use of circuit simulation software.

[5] This approach is perfectly in keeping with the instructional philosophy of these learning modules: *teaching students to be self-sufficient thinkers*. Answer keys can be useful, but it is even more useful to your long-term success to have a set of tools on hand for checking your own work, because once you have left school and are on your own, there will no longer be "answer keys" available for the problems you will have to solve.

7.2.1 Miscellaneous physical constants

Note: constants shown in **bold** type are *exact*, not approximations. Values inside of parentheses show one standard deviation (σ) of uncertainty in the final digits: for example, the magnetic permeability of free space value given as $1.25663706212(19) \times 10^{-6}$ H/m represents a center value (i.e. the location parameter) of $1.25663706212 \times 10^{-6}$ Henrys per meter with one standard deviation of uncertainty equal to $0.0000000000019 \times 10^{-6}$ Henrys per meter.

Avogadro's number (N_A) = **6.02214076** $\times 10^{23}$ **per mole** (mol^{-1})

Boltzmann's constant (k) = **1.380649** $\times 10^{-23}$ **Joules per Kelvin** (J/K)

Electronic charge (e) = **1.602176634** $\times 10^{-19}$ **Coulomb** (C)

Faraday constant (F) = **96,485.33212...** $\times 10^{4}$ **Coulombs per mole** (C/mol)

Magnetic permeability of free space (μ_0) = $1.25663706212(19) \times 10^{-6}$ Henrys per meter (H/m)

Electric permittivity of free space (ϵ_0) = $8.8541878128(13) \times 10^{-12}$ Farads per meter (F/m)

Characteristic impedance of free space (Z_0) = 376.730313668(57) Ohms (Ω)

Gravitational constant (G) = $6.67430(15) \times 10^{-11}$ cubic meters per kilogram-seconds squared ($m^3/kg\text{-}s^2$)

Molar gas constant (R) = **8.314462618... Joules per mole-Kelvin** (J/mol-K) = 0.08205746(14) liters-atmospheres per mole-Kelvin

Planck constant (h) = **6.62607015** $\times 10^{-34}$ **joule-seconds** (J-s)

Stefan-Boltzmann constant (σ) = **5.670374419...** $\times 10^{-8}$ **Watts per square meter-Kelvin**^{4} ($W/m^2 \cdot K^4$)

Speed of light in a vacuum (c) = **299,792,458 meters per second** (m/s) = 186282.4 miles per second (mi/s)

Note: All constants taken from NIST data "Fundamental Physical Constants – Complete Listing", from `http://physics.nist.gov/constants`, National Institute of Standards and Technology (NIST), 2018 CODATA Adjustment.

7.2.2 Introduction to spreadsheets

A powerful computational tool you are encouraged to use in your work is a *spreadsheet*. Available on most personal computers (e.g. Microsoft Excel), *spreadsheet* software performs numerical calculations based on number values and formulae entered into cells of a grid. This grid is typically arranged as lettered columns and numbered rows, with each cell of the grid identified by its column/row coordinates (e.g. cell `B3`, cell `A8`). Each cell may contain a string of text, a number value, or a mathematical formula. The spreadsheet automatically updates the results of all mathematical formulae whenever the entered number values are changed. This means it is possible to set up a spreadsheet to perform a series of calculations on entered data, and those calculations will be re-done by the computer any time the data points are edited in any way.

For example, the following spreadsheet calculates average speed based on entered values of distance traveled and time elapsed:

	A	B	C	D
1	Distance traveled	46.9	Kilometers	
2	Time elapsed	1.18	Hours	
3	Average speed	= B1 / B2	km/h	
4				
5				

Text labels contained in cells `A1` through `A3` and cells `C1` through `C3` exist solely for readability and are not involved in any calculations. Cell `B1` contains a sample distance value while cell `B2` contains a sample time value. The formula for computing speed is contained in cell `B3`. Note how this formula begins with an "equals" symbol (`=`), references the values for distance and speed by lettered column and numbered row coordinates (`B1` and `B2`), and uses a forward slash symbol for division (`/`). The coordinates `B1` and `B2` function as *variables*[6] would in an algebraic formula.

When this spreadsheet is executed, the numerical value `39.74576` will appear in cell `B3` rather than the formula `= B1 / B2`, because 39.74576 is the computed speed value given 46.9 kilometers traveled over a period of 1.18 hours. If a different numerical value for distance is entered into cell `B1` or a different value for time is entered into cell `B2`, cell `B3`'s value will automatically update. All you need to do is set up the given values and any formulae into the spreadsheet, and the computer will do all the calculations for you.

Cell `B3` may be referenced by other formulae in the spreadsheet if desired, since it is a variable just like the given values contained in `B1` and `B2`. This means it is possible to set up an entire chain of calculations, one dependent on the result of another, in order to arrive at a final value. The arrangement of the given data and formulae need not follow any pattern on the grid, which means you may place them anywhere.

[6] Spreadsheets may also provide means to attach text labels to cells for use as variable names (Microsoft Excel simply calls these labels "names"), but for simple spreadsheets such as those shown here it's usually easier just to use the standard coordinate naming for each cell.

Common[7] arithmetic operations available for your use in a spreadsheet include the following:

- Addition (`+`)
- Subtraction (`-`)
- Multiplication (`*`)
- Division (`/`)
- Powers (`^`)
- Square roots (`sqrt()`)
- Logarithms (`ln()` , `log10()`)

Parentheses may be used to ensure[8] proper order of operations within a complex formula. Consider this example of a spreadsheet implementing the *quadratic formula*, used to solve for roots of a polynomial expression in the form of $ax^2 + bx + c$:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

	A	B
1	x_1	= (-B4 + sqrt((B4^2) - (4*B3*B5))) / (2*B3)
2	x_2	= (-B4 - sqrt((B4^2) - (4*B3*B5))) / (2*B3)
3	a =	9
4	b =	5
5	c =	-2

This example is configured to compute roots[9] of the polynomial $9x^2 + 5x - 2$ because the values of 9, 5, and -2 have been inserted into cells `B3`, `B4`, and `B5`, respectively. Once this spreadsheet has been built, though, it may be used to calculate the roots of *any* second-degree polynomial expression simply by entering the new a, b, and c coefficients into cells `B3` through `B5`. The numerical values appearing in cells `B1` and `B2` will be automatically updated by the computer immediately following any changes made to the coefficients.

[7] Modern spreadsheet software offers a bewildering array of mathematical functions you may use in your computations. I recommend you consult the documentation for your particular spreadsheet for information on operations other than those listed here.

[8] Spreadsheet programs, like text-based programming languages, are designed to follow standard order of operations by default. However, my personal preference is to use parentheses even where strictly unnecessary just to make it clear to any other person viewing the formula what the intended order of operations is.

[9] Reviewing some algebra here, a *root* is a value for x that yields an overall value of zero for the polynomial. For this polynomial ($9x^2 + 5x - 2$) the two roots happen to be $x = 0.269381$ and $x = -0.82494$, with these values displayed in cells `B1` and `B2`, respectively upon execution of the spreadsheet.

Alternatively, one could break up the long quadratic formula into smaller pieces like this:

$$y = \sqrt{b^2 - 4ac} \qquad z = 2a$$

$$x = \frac{-b \pm y}{z}$$

	A	B	C
1	x_1	= (-B4 + C1) / C2	= sqrt((B4^2) - (4*B3*B5))
2	x_2	= (-B4 - C1) / C2	= 2*B3
3	a =	9	
4	b =	5	
5	c =	-2	

Note how the square-root term (y) is calculated in cell C1, and the denominator term (z) in cell C2. This makes the two final formulae (in cells B1 and B2) simpler to interpret. The positioning of all these cells on the grid is completely arbitrary[10] – all that matters is that they properly reference each other in the formulae.

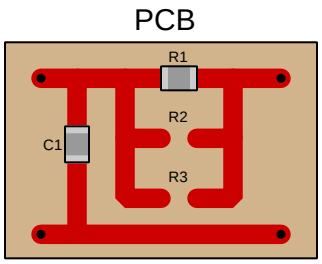
Spreadsheets are particularly useful for situations where the same set of calculations representing a circuit or other system must be repeated for different initial conditions. The power of a spreadsheet is that it automates what would otherwise be a tedious set of calculations. One specific application of this is to simulate the effects of various components within a circuit failing with abnormal values (e.g. a shorted resistor simulated by making its value nearly zero; an open resistor simulated by making its value extremely large). Another application is analyzing the behavior of a circuit design given new components that are out of specification, and/or aging components experiencing drift over time.

[10] My personal preference is to locate all the "given" data in the upper-left cells of the spreadsheet grid (each data point flanked by a sensible name in the cell to the left and units of measurement in the cell to the right as illustrated in the first distance/time spreadsheet example), sometimes coloring them in order to clearly distinguish which cells contain entered data versus which cells contain computed results from formulae. I like to place all formulae in cells below the given data, and try to arrange them in logical order so that anyone examining my spreadsheet will be able to figure out *how* I constructed a solution. This is a general principle I believe all computer programmers should follow: *document and arrange your code to make it easy for other people to learn from it.*

7.2.3 RC filter component selection program

A challenge faced when designing an analog filter network is selecting appropriate component values from a limited selection such as the common IEC "E12" series[11].

One way to make the most of limited component values on a printed circuit board (PCB) design where you might have to choose component values after the board has been manufactured is to provided pads where components may be soldered in if needed, but may be left empty if not. Consider the following PCB layout example, where three places exist for surface-mount resistors along with one place for a surface-mount capacitor, to form a simple RC filter network:



Even though the E12 series only provides us with twelve unique values within one decade of range, being able to combine multiple components in parallel like this enables far more than twelve unique combinations of resistance within the same range.

Choosing just the right combination of resistances and capacitances to achieve a desired cutoff frequency, however, is a daunting task because so many possible combinations exist even with just twelve choices per component (thirteen if you include the option of "no resistor" in the spaces for R_2 and/or R_3). Here is where computers can be a useful aid, by automating the selection process. Consider the following program written in the C language to choose component values for this RC filter network:

```
#include <stdio.h>
#include <math.h>

int main (void)
{
  float R[13] = {10e3, 12e3, 15e3, 18e3, 22e3, 27e3, 33e3,
                 39e3, 47e3, 56e3, 68e3, 82e3, 999e9};
  float C[12] = {1e-9, 1.2e-9, 1.5e-9, 1.8e-9, 2.2e-9, 2.7e-9,
                 3.3e-9, 3.9e-9, 4.7e-9, 5.6e-9, 6.8e-9, 8.2e-9};
  float f_c, rp, f, error, error_least = 999e9;
  float f_match, c1_match, r1_match, r2_match, r3_match;
```

[11] The E12 series encompasses values having significant figures of 10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, and 82. For example, a resistor that is 1.8 kΩ would fall within the E12 series, as would a 470 μF capacitor and a 0.15 mH inductor.

```
  int r1, r2, r3, c1;

  printf ("Enter desired cutoff frequency: ");
  scanf ("%f", &f_c);

  for (r1 = 0; r1 < 12; ++r1)
    for (r2 = 0; r2 < 13; ++r2)
      for (r3 = 0; r3 < 13; ++r3)
        for (c1 = 0; c1 < 12; ++c1)
        {
          rp = 1 / (1 / R[r1] + 1 / R[r2] + 1 / R[r3]);
          f = 1 / (2 * M_PI * rp * C[c1]);
          error = fabs (f - f_c) / f_c;

          if (error < error_least)
          {
            error_least = error;
            f_match = f;
            r1_match = R[r1];
            r2_match = R[r2];
            r3_match = R[r3];
            c1_match = C[c1];
          }
        }

  rp = 1 / (1 / r1_match + 1 / r2_match + 1 / r3_match);
  f = 1 / (2 * M_PI * rp * c1_match);

  printf ("Best-match cutoff frequency = %f Hertz\n", f_match);
  printf ("Best-match error = %f percent\n", (f - f_c) / f_c * 100.0);

  printf ("R1 = %.1f kiloOhms\n", r1_match / 1e3);

  if (r2_match < 100e3)
    printf ("R2 = %.1f kiloOhms\n", r2_match / 1e3);

  if (r3_match < 100e3)
    printf ("R3 = %.1f kiloOhms\n", r3_match / 1e3);

  printf ("C1 = %.1f nanoFarads\n", c1_match * 1e9);

  return 0;
}
```

Examine this program's source code and then answer the following questions:

- How exactly does the program choose the best combination of R and C values for the user-selected cutoff frequency?
- What is the lowest cutoff frequency possible given the component values available in the two floating-point arrays?
- What is the highest cutoff frequency possible given the component values available in the two floating-point arrays?
- What type of RC filter network is this, based on the PCB layout shown?
- Which terminals comprise the input for this RC filter network, and which comprise the output?

Challenges

- What constraints other than cutoff frequency might be important in the design of such a filter network?
- How could this program be altered to accommodate a different range of possible cutoff frequencies, given the lowest and highest values answered above?

7.2.4 Sallen-Key active low-pass filter design

Design an active filter circuit having a low-pass cutoff frequency of 5 kHz and an input impedance of at least 20 kΩ, using a Sallen-Key network topology. Use nothing but standard E12 series components in your design, and demonstrate its performance by computer simulation.

Challenges

- How may we increase the Q value of this filter?

7.2.5 Sallen-Key active high-pass filter design

Design an active filter circuit having a high-pass cutoff frequency of 800 Hz and an input impedance of at least 10 kΩ, using a Sallen-Key network topology. Use nothing but standard E12 series components in your design, and demonstrate its performance by computer simulation.

Challenges

- How may we increase the Q value of this filter?

7.3 Diagnostic reasoning

These questions are designed to stimulate your deductive and inductive thinking, where you must apply general principles to specific scenarios (deductive) and also derive conclusions about the failed circuit from specific details (inductive). In a Socratic discussion with your instructor, the goal is for these questions to reinforce your recall and use of general circuit principles and also challenge your ability to integrate multiple symptoms into a sensible explanation of what's wrong in a circuit. Your instructor may also pose additional questions based on those assigned, in order to further challenge and sharpen your diagnostic abilities.

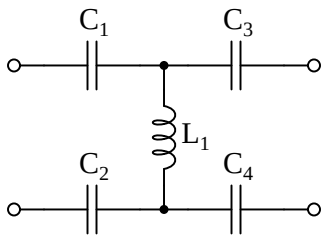
As always, your goal is to fully *explain* your analysis of each problem. Simply obtaining a correct answer is not good enough – you must also demonstrate sound reasoning in order to successfully complete the assignment. Your instructor's responsibility is to probe and challenge your understanding of the relevant principles and analytical processes in order to ensure you have a strong foundation upon which to build further understanding.

You will note a conspicuous lack of answers given for these diagnostic questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your diagnostic answers, where applicable, is to use circuit simulation software to explore the effects of faults placed in circuits. For example, if one of these diagnostic questions requires that you predict the effect of an open or a short in a circuit, you may check the validity of your work by simulating that same fault (substituting a very high resistance in place of that component for an open, and substituting a very low resistance for a short) within software and seeing if the results agree.

7.3.1 Effects of faults in a multi-order filter

Identify the type of filter network shown in the following schematic:



Identify the effect(s) that capacitor C_2 failing open will have on the performance of this filter.

Identify the effect(s) that inductor L_1 failing open will have on the performance of this filter.

Identify the effect(s) that capacitor C_3 failing shorted will have on the performance of this filter.

Identify the effect(s) that inductor L_1 failing shorted will have on the performance of this filter.

Challenges

- Re-draw an unbalanced version of this same filter network.

Appendix A

Problem-Solving Strategies

The ability to solve complex problems is arguably one of the most valuable skills one can possess, and this skill is particularly important in any science-based discipline.

- Study principles, not procedures. Don't be satisfied with merely knowing how to compute solutions – learn *why* those solutions work.
- Identify what it is you need to solve, identify all relevant data, identify all units of measurement, identify any general principles or formulae linking the given information to the solution, and then identify any "missing pieces" to a solution. Annotate all diagrams with this data.
- Sketch a diagram to help visualize the problem. When building a real system, always devise a plan for that system and analyze its function *before* constructing it.
- Follow the units of measurement and meaning of every calculation. If you are ever performing mathematical calculations as part of a problem-solving procedure, and you find yourself unable to apply each and every intermediate result to some aspect of the problem, it means you don't understand what you are doing. Properly done, every mathematical result should have practical meaning for the problem, and not just be an abstract number. You should be able to identify the proper units of measurement for each and every calculated result, and show where that result fits into the problem.
- Perform "thought experiments" to explore the effects of different conditions for theoretical problems. When troubleshooting real systems, perform *diagnostic tests* rather than visually inspecting for faults, the best diagnostic test being the one giving you the most information about the nature and/or location of the fault with the fewest steps.
- Simplify the problem until the solution becomes obvious, and then use that obvious case as a model to follow in solving the more complex version of the problem.
- Check for exceptions to see if your solution is incorrect or incomplete. A good solution will work for *all* known conditions and criteria. A good example of this is the process of testing scientific hypotheses: the task of a scientist is not to find support for a new idea, but rather to *challenge* that new idea to see if it holds up under a battery of tests. The philosophical

principle of *reductio ad absurdum* (i.e. disproving a general idea by finding a specific case where it fails) is useful here.

- Work "backward" from a hypothetical solution to a new set of given conditions.
- Add quantities to problems that are qualitative in nature, because sometimes a little math helps illuminate the scenario.
- Sketch graphs illustrating how variables relate to each other. These may be quantitative (i.e. with realistic number values) or qualitative (i.e. simply showing increases and decreases).
- Treat quantitative problems as qualitative in order to discern the relative magnitudes and/or directions of change of the relevant variables. For example, try determining what happens if a certain variable were to increase or decrease before attempting to precisely calculate quantities: how will each of the dependent variables respond, by increasing, decreasing, or remaining the same as before?
- Consider limiting cases. This works especially well for qualitative problems where you need to determine which direction a variable will change. Take the given condition and magnify that condition to an extreme degree as a way of simplifying the direction of the system's response.
- Check your work. This means regularly testing your conclusions to see if they make sense. This does *not* mean repeating the same steps originally used to obtain the conclusion(s), but rather to use some other means to check validity. Simply repeating procedures often leads to *repeating the same errors* if any were made, which is why alternative paths are better.

Appendix B

Instructional philosophy

B.1 First principles of learning

- **Anyone can learn anything** given appropriate time, effort, resources, challenges, encouragement, and expectations. Dedicating time and investing effort are the student's responsibility; providing resources, challenges, and encouragement are the teacher's responsibility; high expectations are a responsibility shared by both student and teacher.

- **Transfer is not automatic.** The human mind has a natural tendency to compartmentalize information, which means the process of taking knowledge learned in one context and applying it to another usually does not come easy and therefore should never be taken for granted.

- **Learning is iterative.** The human mind rarely learns anything perfectly on the first attempt. Anticipate mistakes and plan for multiple tries to achieve full understanding, using the lessons of those mistakes as feedback to guide future attempts.

- **Information is absorbed, but understanding is created.** Facts and procedures may be memorized easily enough by repeated exposure, but the ability to reliably apply principles to novel scenarios only comes through intense personal effort. This effort is fundamentally creative in nature: casting new concepts in one's own words, running mental experiments to perceive patterns and trends, etc. These acts of making knowledge "one's own" need not be perfect in order to be effective, as the value lies in the activity and not necessarily the finished product. Depth of understanding and permanence of memory are both directly related to mental effort during the learning process – i.e. *easy learning is shallow and fleeting.*

- **Education trumps training.** There is no such thing as an entirely isolated subject, as all fields of knowledge are connected. Training is narrowly-focused and task-oriented. Education is broad-based and principle-oriented. When preparing for a life-long technical career, education beats training every time.

- **Character matters.** Poor habits are more destructive than deficits of knowledge or skill. This is especially true in collective endeavors, where a team's ability to function depends on trust between its members. Simply put, no one wants an untrustworthy person on their team. An essential component of education then, is character development.

- **People learn to be responsible by bearing responsibility.** An irresponsible person is someone who has never *had* to be responsible for anything that mattered enough to them. Just as anyone can learn anything, anyone can become responsible if the personal cost of irresponsibility becomes high enough.

- **What gets measured, gets done.** Accurate and relevant assessment of learning is key to ensuring all students learn. Therefore, it is imperative to measure what matters.

- **Failure is nothing to fear.** Every human being fails, and fails in multiple ways at multiple times. Eventual success only happens when we don't stop trying.

B.2 Proven strategies for instructors

- Assume every student is capable of learning anything they desire given the proper conditions. Treat them as capable adults by granting real responsibility and avoiding artificial incentives such as merit or demerit points.
- Create a consistent culture of high expectations across the entire program of study. Demonstrate and encourage patience, persistence, and a healthy sense of self-skepticism. Anticipate and de-stigmatize error. Teach respect for the capabilities of others as well as respect for one's own fallibility.
- Replace lecture with "inverted" instruction, where students first encounter new concepts through reading and then spend class time in Socratic dialogue with the instructor exploring those concepts and solving problems individually. There is a world of difference between observing someone solve a problem versus actually solving a problem yourself, and so the point of this form of instruction is to place students in a position where they *cannot* passively observe.
- Require students to read extensively, write about what they learn, and dialogue with you and their peers to sharpen their understanding. Apply Francis Bacon's advice that "reading maketh a full man; conference a ready man; and writing an exact man". These are complementary activities helping students expand their confidence and abilities.
- Use artificial intelligence (AI) to challenge student understanding rather than merely provide information. Find productive ways for AI to critique students' clarity of thought and of expression, for example by employing AI as a Socratic-style interlocutor or as a reviewer of students' journals. Properly applied, AI has the ability to expand student access to critical review well outside the bounds of their instructor's reach.
- Build frequent and rapid feedback into the learning process so that students know at all times how well they are learning, to identify problems early and fix them before they grow. Model the intellectual habit of self-assessing and self-correcting your own understanding (i.e. a cognitive *feedback loop*), encouraging students to do the same.
- Use "mastery" as the standard for every assessment, which means the exam or experiment or project must be done with 100% competence in order to pass. Provide students with multiple opportunity for re-tries (different versions of the assessment every time).
- Require students to devise their own hypotheses and procedures on all experiments, so that the process is truly a scientific one. Have students assess their proposed experimental procedures for risk and devise mitigations for those risks. Let nothing be pre-designed about students' experiments other than a stated task (i.e. what principle the experiment shall test) at the start and a set of demonstrable knowledge and skill objectives at the end.
- Have students build as much of their lab equipment as possible: building power sources, building test assemblies[1], and building complete working systems (no kits!). In order to provide

[1] In the program I teach, every student builds their own "Development Board" consisting of a metal chassis with DIN rail, terminal blocks, and an AC-DC power supply of their own making which functions as a portable lab environment they can use at school as well as take home.

this same "ground-up" experience for every new student, this means either previous students take their creations with them, or the systems get disassembled in preparation for the new students, or the systems grow and evolve with each new student group.

- Incorporate external accountability for you and for your students, continuously improving the curriculum and your instructional methods based on proven results. Have students regularly network with active professionals through participation in advisory committee meetings, service projects, tours, jobshadows, internships, etc. Practical suggestions include requiring students to design and build projects for external clients (e.g. community groups, businesses, different departments within the institution), and also requiring students attend all technical advisory committee meetings and to dialogue with the industry representatives at those meetings.
- Repeatedly explore difficult-to-learn concepts across multiple courses, so that students have multiple opportunities to build their understanding.
- Relate all new concepts, whenever possible, to previous concepts and to relevant physical laws. Challenge each and every student, every day, to *reason* from concept to concept and to explain the logical connections between. Challenge students to verify their conclusions by multiple approaches (e.g. double-checking their work using different methods). Ask *"Why?"* often.
- Maintain detailed records on each student's performance and share these records privately with them. These records should include academic performance as well as professionally relevant behavioral tendencies.
- Hold mandatory "check-in" meetings between all program faculty and each new student during their first term. Offer these to all other students as an option, except for any students continuing to manifest unprofessional behaviors, poor academic performance, or who have some other need for a face-to-face meeting with faculty.
- Address problems while they are small, before they grow larger. This is equally true for tutoring technical concepts as it is for helping students build professional habits.
- Build rigorous quality control into the curriculum to ensure every student masters every important concept, and that the mastery is retained over time. This includes (1) review questions added to every exam to re-assess knowledge taught in previous terms, (2) cumulative exams at the end of every term to re-assess all important concepts back to the very beginning of the program, and (3) review assessments in practical (hands-on) coursework to ensure critically-important skills were indeed taught and are still retained. What you will find by doing this is that it actually boosts retention of students by ensuring that important knowledge gets taught and is retained over long spans of time. In the absence of such quality control, student learning and retention tends to be spotty and this contributes to drop-out and failure rates later in their education.
- Finally, *never rush learning.* Education is not a race. Give your students ample time to digest complex ideas, as you continually remind yourself of just how long it took you to achieve mastery! Long-term retention and the consistently correct application of concepts are always the result of *focused effort over long periods of time* which means there are no shortcuts to learning.

B.3 Proven strategies for students

The single most important piece of advice I have for any student of any subject is to take responsibility for your own development in all areas of life including mental development. Expecting others in your life to entirely guide your own development is a recipe for disappointment. This is just as true for students enrolled in formal learning institutions as it is for auto-didacts pursuing learning entirely on their own. Learning to think in new ways is key to being able to gainfully use information, to make informed decisions about your life, and to best serve those you care about. With this in mind, I offer the following advice to students:

- **Approach all learning as valuable**. No matter what course you take, no matter who you learn from, no matter the subject, there is something useful in every learning experience. If you don't see the value of every new experience, you are not looking closely enough!
- **Continually challenge yourself**. Let other people take shortcuts and find easy answers to easy problems. The purpose of education is to stretch your mind, in order to shape it into a more powerful tool. This doesn't come by taking the path of least resistance. An excellent analogy for an empowering education is productive physical exercise: becoming stronger, more flexible, and more persistent only comes through intense personal effort.
- **Master the use of language**. This includes reading extensively, writing every day, listening closely, and speaking articulately. To a great extent language channels and empowers thought, so the better you are at wielding language the better you will be at grasping abstract concepts and articulating them not only for your benefit but for others as well.
- **Do not limit yourself to the resources given to you**. Read books that are not on the reading list. Run experiments that aren't assigned to you. Form study groups outside of class. Take an entrepreneurial approach to your own education, as though it were a business you were building for your future benefit.
- **Express and share what you learn**. Take every opportunity to teach what you have learned to others, as this will not only help them but will also strengthen your own understanding[2].
- Realize that **no one can give you understanding**, just as no one can give you physical fitness. These both must be *built*.
- **Above all, recognize that learning is hard work, and that a certain level of frustration is unavoidable**. There are times when you will struggle to grasp some of these concepts, and that struggle is a natural thing. Take heart that it will yield with persistent and varied[3] effort, and never give up! That concepts don't immediately come to you is not a sign of something wrong, but rather of something right: that you have found a worthy challenge!

[2] On a personal note, I was surprised to learn just how much my own understanding of electronics and related subjects was strengthened by becoming a teacher. When you are tasked every day with helping other people grasp complex topics, it catalyzes your own learning by giving you powerful incentives to study, to articulate your thoughts, and to reflect deeply on the process of learning.

[3] As the old saying goes, "Insanity is trying the same thing over and over again, expecting different results." If you find yourself stumped by something in the text, you should attempt a different approach. Alter the thought experiment, change the mathematical parameters, do whatever you can to see the problem in a slightly different light, and then the solution will often present itself more readily.

B.4 Design of these learning modules

"The unexamined circuit is not worth energizing" – Socrates (if he had taught electricity)

These learning modules, although useful for self-study, were designed to be used in a formal learning environment where a subject-matter expert challenges students to digest the content and exercise their critical thinking abilities in the answering of questions and in the construction and testing of working circuits. Every effort has been made to embed the following instructional and assessment philosophies within:

- The first goal of education is to enhance clear and independent thought, in order that every student reach their fullest potential in a highly complex and inter-dependent world. Robust reasoning is *always* more important than particulars of any subject matter, because its application is universal.
- Literacy is fundamental to independent learning and thought because text continues to be the most efficient way to communicate complex ideas over space and time. Those who cannot read with ease are limited in their ability to acquire knowledge and perspective.
- Articulate communication is fundamental to work that is complex and interdisciplinary.
- Faulty assumptions and poor reasoning are best corrected through challenge, not presentation. The rhetorical technique of *reductio ad absurdum* (disproving an assertion by exposing an absurdity) works well to discipline student's minds, not only to correct the problem at hand but also to learn how to detect and correct future errors.
- Important principles should be repeatedly explored and widely applied throughout a course of study, not only to reinforce their importance and help ensure their mastery, but also to showcase the interconnectedness and utility of knowledge.

These learning modules were expressly designed to be used in an "inverted" teaching environment[4] where students first read the introductory and tutorial chapters on their own, then individually attempt to answer the questions and construct working circuits according to the experiment and project guidelines. The instructor never lectures, but instead meets regularly with each individual student to review their progress, answer questions, identify misconceptions, and challenge the student to new depths of understanding through further questioning. Regular meetings between instructor and student should resemble a Socratic[5] dialogue, where questions serve as scalpels to dissect topics and expose assumptions. The student passes each module only after consistently demonstrating their ability to logically analyze and correctly apply all major concepts in each question or project/experiment. The instructor must be vigilant in probing each student's understanding to ensure they are truly *reasoning* and not just *memorizing*. This is why "Challenge" points appear throughout, as prompts for students to think deeper about topics and as starting points for instructor queries. Sometimes these challenge points require additional knowledge that hasn't been covered in the series to answer in full. This is okay, as the major purpose of the Challenges is to stimulate analysis and synthesis on the part of each student.

The instructor must possess enough mastery of the subject matter and awareness of students' reasoning to generate their own follow-up questions to practically any student response. Even completely correct answers given by the student should be challenged by the instructor for the purpose of having students practice articulating their thoughts and defending their reasoning. Conceptual errors committed by the student should be exposed and corrected not by direct instruction, but rather by reducing the errors to an absurdity[6] through well-chosen questions and thought experiments posed by the instructor. Becoming proficient at this style of instruction requires time and dedication, but the positive effects on critical thinking for both student and instructor are spectacular.

An inspection of these learning modules reveals certain unique characteristics. One of these is a bias toward thorough explanations in the tutorial chapters. Without a live instructor to explain concepts and applications to students, the text itself must fulfill this role. This philosophy results in lengthier explanations than what you might typically find in a textbook, each step of the reasoning process fully explained, including footnotes addressing common questions and concerns students raise while learning these concepts. Each tutorial seeks to not only explain each major concept in sufficient detail, but also to explain the logic of each concept and how each may be developed

[4] In a traditional teaching environment, students first encounter new information via *lecture* from an expert, and then independently apply that information via *homework*. In an "inverted" course of study, students first encounter new information via *homework*, and then independently apply that information under the scrutiny of an expert. The expert's role in lecture is to simply *explain*, but the expert's role in an inverted session is to *challenge*, *critique*, and if necessary *explain* where gaps in understanding still exist.

[5] Socrates is a figure in ancient Greek philosophy famous for his unflinching style of questioning. Although he authored no texts, he appears as a character in Plato's many writings. The essence of Socratic philosophy is to leave no question unexamined and no point of view unchallenged. While purists may argue a topic such as electric circuits is too narrow for a true Socratic-style dialogue, I would argue that the essential thought processes involved with scientific reasoning on *any* topic are not far removed from the Socratic ideal, and that students of electricity and electronics would do very well to challenge assumptions, pose thought experiments, identify fallacies, and otherwise employ the arsenal of critical thinking skills modeled by Socrates.

[6] This rhetorical technique is known by the Latin phrase *reductio ad absurdum*. The concept is to expose errors by counter-example, since only one solid counter-example is necessary to disprove a universal claim. As an example of this, consider the common misconception among beginning students of electricity that voltage cannot exist without current. One way to apply *reductio ad absurdum* to this statement is to ask how much current passes through a fully-charged battery connected to nothing (i.e. a clear example of voltage existing without current).

from "first principles". Again, this reflects the goal of developing clear and independent thought in students' minds, by showing how clear and logical thought was used to forge each concept. Students benefit from witnessing a model of clear thinking in action, and these tutorials strive to be just that.

Another feature of these learning modules is that they do not treat topics in isolation. Rather, important concepts are introduced early in the series, and appear repeatedly as stepping-stones toward other concepts in subsequent modules. This helps to avoid the "compartmentalization" of knowledge, demonstrating the inter-connectedness of concepts and simultaneously reinforcing them. Each module is fairly complete in itself, reserving the beginning of its tutorial to a review of foundational concepts.

To high standards of education,

Tony R. Kuphaldt

Appendix C

Tools used

I am indebted to the developers of many open-source software applications in the creation of these learning modules. The following is a list of these applications with some commentary on each.

You will notice a theme common to many of these applications: a bias toward *code*. Although I am by no means an expert programmer in any computer language, I understand and appreciate the flexibility offered by code-based applications where the user (you) enters commands into a plain ASCII text file, which the software then reads and processes to create the final output. Code-based computer applications are by their very nature *extensible*, while WYSIWYG (What You See Is What You Get) applications are generally limited to whatever user interface the developer makes for you.

The `GNU/Linux` computer operating system

> There is so much to be said about Linus Torvalds' `Linux` and Richard Stallman's `GNU` project. First, to credit just these two individuals is to fail to do justice to the *mob* of passionate volunteers who contributed to make this amazing software a reality. I first learned of `Linux` back in 1996, and have been using this operating system on my personal computers almost exclusively since then. It is *free*, it is completely *configurable*, and it permits the continued use of highly efficient `Unix` applications and scripting languages (e.g. shell scripts, Makefiles, `sed`, `awk`) developed over many decades. `Linux` not only provided me with a powerful computing platform, but its open design served to inspire my life's work of creating open-source educational resources.

Bram Moolenaar's `Vim` text editor

> Writing code for any code-based computer application requires a *text editor*, which may be thought of as a word processor strictly limited to outputting plain-ASCII text files. Many good text editors exist, and one's choice of text editor seems to be a deeply personal matter within the programming world. I prefer `Vim` because it operates very similarly to `vi` which is ubiquitous on `Unix`/`Linux` operating systems, and because it may be entirely operated via keyboard (i.e. no mouse required) which makes it fast to use.

Donald Knuth's TEX typesetting system

Developed in the late 1970's and early 1980's by computer scientist extraordinaire Donald Knuth to typeset his multi-volume magnum opus *The Art of Computer Programming*, this software allows the production of formatted text for screen-viewing or paper printing, all by writing plain-text code to describe how the formatted text is supposed to appear. TEX is not just a markup language for documents, but it is also a Turing-complete programming language in and of itself, allowing useful algorithms to be created to control the production of documents. Simply put, TEX *is a programmer's approach to word processing*. Since TEX is controlled by code written in a plain-text file, this means anyone may read that plain-text file to see exactly how the document was created. This openness afforded by the code-based nature of TEX makes it relatively easy to learn how other people have created their own TEX documents. By contrast, examining a beautiful document created in a conventional WYSIWYG word processor such as Microsoft `Word` suggests nothing to the reader about *how* that document was created, or what the user might do to create something similar. As Mr. Knuth himself once quipped, conventional word processing applications should be called WYSIAYG (What You See Is *All* You Get).

Leslie Lamport's LATEX extensions to TEX

Like all true programming languages, TEX is inherently extensible. So, years after the release of TEX to the public, Leslie Lamport decided to create a massive extension allowing easier compilation of book-length documents. The result was LATEX, which is the markup language used to create all ModEL module documents. You could say that TEX is to LATEX as `C` is to `C++`. This means it is permissible to use any and all TEX commands within LATEX source code, and it all still works. Some of the features offered by LATEX that would be challenging to implement in TEX include automatic index and table-of-content creation.

Tim Edwards' `Xcircuit` drafting program

This wonderful program is what I use to create all the schematic diagrams and illustrations (but not photographic images or mathematical plots) throughout the ModEL project. It natively outputs PostScript format which is a true vector graphic format (this is why the images do not pixellate when you zoom in for a closer view), and it is so simple to use that I have never had to read the manual! Object libraries are easy to create for `Xcircuit`, being plain-text files using PostScript programming conventions. Over the years I have collected a large set of object libraries useful for drawing electrical and electronic schematics, pictorial diagrams, and other technical illustrations.

`Gimp` graphic image manipulation program

Essentially an open-source clone of Adobe's `PhotoShop`, I use `Gimp` to resize, crop, and convert file formats for all of the photographic images appearing in the ModEL modules. Although `Gimp` does offer its own scripting language (called `Script-Fu`), I have never had occasion to use it. Thus, my utilization of `Gimp` to merely crop, resize, and convert graphic images is akin to using a sword to slice bread.

`SPICE` circuit simulation program

`SPICE` is to circuit analysis as TEX is to document creation: it is a form of markup language designed to describe a certain object to be processed in plain-ASCII text. When the plain-text "source file" is compiled by the software, it outputs the final result. More modern circuit analysis tools certainly exist, but I prefer `SPICE` for the following reasons: it is *free*, it is *fast*, it is *reliable*, and it is a fantastic tool for *teaching* students of electricity and electronics how to write simple code. I happen to use rather old versions of `SPICE`, version 2g6 being my "go to" application when I only require text-based output. `NGSPICE` (version 26), which is based on Berkeley `SPICE` version 3f5, is used when I require graphical output for such things as time-domain waveforms and Bode plots. In all `SPICE` example netlists I strive to use coding conventions compatible with all `SPICE` versions.

Andrew D. Hwang's `ePiX` mathematical visualization programming library

This amazing project is a `C++` library you may link to any `C/C++` code for the purpose of generating PostScript graphic images of mathematical functions. As a completely free and open-source project, it does all the plotting I would otherwise use a Computer Algebra System (CAS) such as `Mathematica` or `Maple` to do. It should be said that `ePiX` is *not* a Computer Algebra System like `Mathematica` or `Maple`, but merely a mathematical *visualization* tool. In other words, it won't determine integrals for you (you'll have to implement that in your own `C/C++` code!), but it can graph the results, and it does so beautifully. What I really admire about `ePiX` is that it is a `C++` programming library, which means it builds on the existing power and toolset available with that programming language. Mr. Hwang could have probably developed his own stand-alone application for mathematical plotting, but by creating a `C++` library to do the same thing he accomplished something much greater.

`gnuplot` mathematical visualization software

Another open-source tool for mathematical visualization is `gnuplot`. Interestingly, this tool is *not* part of Richard Stallman's GNU project, its name being a coincidence. For this reason the authors prefer "gnu" *not* be capitalized at all to avoid confusion. This is a much "lighter-weight" alternative to a spreadsheet for plotting tabular data, and the fact that it easily outputs directly to an X11 console or a file in a number of different graphical formats (including PostScript) is very helpful. I typically set my `gnuplot` output format to default (X11 on my Linux PC) for quick viewing while I'm developing a visualization, then switch to PostScript file export once the visual is ready to include in the document(s) I'm writing. As with my use of `Gimp` to do rudimentary image editing, my use of `gnuplot` only scratches the surface of its capabilities, but the important points are that it's *free* and that it *works well*.

`Python` programming language

Both Python and C++ find extensive use in these modules as instructional aids and exercises, but I'm listing Python here as a *tool* for myself because I use it almost daily as a *calculator*. If you open a Python interpreter console and type `from math import *` you can type mathematical expressions and have it return results just as you would on a hand calculator. Complex-number (i.e. *phasor*) arithmetic is similarly supported if you include the complex-math library (`from cmath import *`). Examples of this are shown in the Programming References chapter (if included) in each module. Of course, being a fully-featured programming language, Python also supports conditionals, loops, and other structures useful for calculation of quantities. Also, running in a console environment where all entries and returned values show as text in a chronologically-ordered list makes it easy to copy-and-paste those calculations to document exactly how they were performed.

Appendix D

Creative Commons License

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or

limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

A. reproduce and Share the Licensed Material, in whole or in part; and

B. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures.

For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors,

whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully

be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at `creativecommons.org/policies`, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at `creativecommons.org`.

Appendix E

References

Bond, C.R., "Bessel Filters – Polynomials, Poles and Circuit Elements", 2003.

Bond, C.R., "Butterworth Filters – Polynomials, Poles and Circuit Elements", 2003.

Bond, C.R., "Optimum L Filters – Polynomials, Poles and Circuit Elements", 2004.

Butterworth, S., "On The Theory of Filter Amplifiers", The Wireless Engineer pages 536-541, October 1930.

Kadah, Yasser Mostafa, "Electronic System Design – Part 1: Analog Filters".

Sanchez-Sinencio, Edgar, "Filter Approximations & Frequency Transformations" lecture notes, ECEN 622 Active Filters Analysis and Design course, Texas A&M University, 2013.

White Electromagnetics, Inc., *A Handbook On Electrical Filters*, Rockville, MD, 1963.

Zumbahlen, Hank, *Basic Linear Design*, Analog Devices Incorporated, Norwood, MA, 2007.

Zumbahlen, Hank, "The Bessel Response", Mini Tutorial MT-204, Analog Devices Incorporated, Norwood, MA, 2012.

Zumbahlen, Hank, "The Butterworth Response", Mini Tutorial MT-224, Analog Devices Incorporated, Norwood, MA, 2012.

Zumbahlen, Hank, "The Chebyshev Response", Mini Tutorial MT-206, Analog Devices Incorporated, Norwood, MA, 2012.

Appendix F

Version history

This is a list showing all significant additions, corrections, and other edits made to this learning module. Each entry is referenced by calendar date in reverse chronological order (newest version first), which appears on the front cover of every learning module for easy reference. Any contributors to this open-source document are listed here as well.

15-20 April 2026 – added content to the Tutorial chapter regarding variable Q for active filters.

6-10 December 2025 – added a new Tutorial section showing how to derive transfer functions for active filter networks. Also edited component labeling in the "Low-pass active filter design example" and "High-pass active filter design example" subsections of the Tutorial to be consistent with component labeling in the "Active filter transfer functions" section, and did some minor editing of footnotes in those subsections too. Also edited the two SPICE simulations to use `vdb` (decibel) plotting instead of generic logarithmic (`loglog`) plotting.

30 November 2025 – added some new questions on active filters, and also made some minor additions to the "Active filter design" section of the Tutorial explaining in more detail how to scale resistor and capacitor values to use standard-sized components. Also re-named the filter design example subsections to specify either "passive" or "active" to improve searchability.

4-6 November 2025 – added some instructor notes.

9 April 2025 – grammatical error correction in the Tutorial courtesy of Jacob Stormes ("are other some of the other"). Also edited some of the instructor notes.

28 March 2025 – added an instructor recommendation to the Introduction chapter.

7 November 2024 – minor edits to the Tutorial.

2-3 November 2024 – added a new Tutorial section on the relationship between transfer functions and filter Bode plots.

30 October 2024 – divided the Introduction chapter into sections, one with recommendations for

students, one with a listing of challenging concepts, and one with recommendations for instructors.

5 April 2024 – added content to the (empty) Introduction chapter.

10 March 2024 – added some questions to the module and fixed on mis-spelling of a word.

23 December 2023 – minor edit to image_6112, inverting the phase on two of the AC waveforms to match the phase of the others.

12 November 2023 – included a Technical Reference section on operational amplifier imperfections.

20 October 2023 – added a new Tutorial section on output-limited filter networks.

9 August 2023 – minor change to C code in the "RC filter component selection program" Quantitative Reasoning question to avoid potential compiler warnings related to literal formatting of a "percent" symbol. Also added formatting to the floating-point numerical outputs to show fewer (unnecessary) trailing zeros.

23 July 2023 – added a Quantitative Reasoning question showing a program written in C for selecting common (E12) component values to achieve precise cutoff frequencies in simple single-pole RC filter circuits.

6-7 September 2022 – added a Programming References chapter with a section showing a C++ program for lowpass and highpass analog filter design, as well as a Case Tutorial chapter with examples of passive filter designs.

1-13 April 2022 – added more content to the Tutorial chapter, and decided to fold the Case Tutorial section written on 31 March 2022 into the Tutorial instead.

31 March 2022 – added more content to the Case Tutorial chapter.

30 March 2022 – added more content to the Tutorial chapter.

29 March 2022 – document first created.

Index

s variable, 29, 81, 107

Active versus passive components, 71
Adding quantities to a qualitative problem, 170
Admittance, 13
Angular velocity, 29, 81
Annotating diagrams, 169
Anti-aliasing filter, 75

Bass, 16
Bel, 94
Bessel characteristic, 27, 59, 62, 76
Bias current, 124
Bias voltage, opamp, 128
Bode plot, 16, 34, 50, 81
Brick wall filter response, 16
Butterworth characteristic, 8, 9, 27, 58, 60, 62, 64, 65, 67, 69, 75–77, 79, 115

C++, 134
Capacitive reactance, 17, 19
Cardinal rule of negative feedback, 82
Cauer, Wilhelm, 62
Center frequency, 19
Chebyshev characteristic, 27, 60, 62, 76, 117
Checking for exceptions, 170
Checking your work, 170
CMRR, 131
Code, computer, 179
Common logarithm, 94
Common-mode rejection ratio, 131
Common-mode voltage, 123, 131
Compiler, C++, 134
Component values, IEC standard, 104
Computer programming, 133
Conductance, 13
Current, bias, 124
Cutoff frequency, 17, 19, 81

dB, 94
dBm, 98
dBW, 99
Decibel, 94
Differentiation, applied to capacitive voltage and current, 110, 111
Dimensional analysis, 169
Diode, protection, 122, 126

Edwards, Tim, 180
Elliptic characteristic, 27, 62, 76
ePiX C++ library, 35
Euler's Relation, 106

Feedback, positive, 72
Filter topology, 24
Fourier transform, 100
Frequency response, 50
Frequency, center, 19
Frequency, cutoff, 17, 19, 81

Gain, amplifier, 94
Gain-bandwidth product, 129
GBWP, 129
Graph values to solve a problem, 170
Greenleaf, Cynthia, 147
Ground, 23

How to teach with these modules, 177
Hwang, Andrew D., 181

Identify given data, 169
Identify relevant principles, 169
IEC 60063 standard, 104
IEC standard component values, 104
Impedance, 17, 110, 111

Inductive reactance, 17, 19
Input protection diode, 122, 126
Input, rail-to-rail, 123
Intermediate results, 169
Interpreter, Python, 138
Inverted instruction, 177

Java, 135
Joule's Law, 101

Kirchhoff's Current Law, 13, 83
Kirchhoff's Voltage Law, 13, 84
Knuth, Donald, 180

Ladder filter topology, 64
Lamport, Leslie, 180
Laplace transform, 100
Latch-up, 123
Limiting cases, 5, 18, 21, 170
Logarithm, common, 94

Maxwell, James Clerk, 89
Metacognition, 152
MFB filter topology, 73
Mho, 13
Monotonicity, 28
Moolenaar, Bram, 179
Multiple feedback filter topology, 73
Murphy, Lynn, 147

Natural frequency, 29, 81
Neper frequency, 30, 81
Network topology, 24

Offset currrent, opamp, 127
Offset voltage, opamp, 128
Ohm, 13
Ohm's Law, 13, 83
Opamp, 73
Open-source, 179
Operational amplifier, 73
Output, rail-to-rail, 121

Passband, 17, 81
Passive versus active components, 71
Phasor, 12, 106
Phasor diagram, 12
Pole, 33, 81
Pole-zero plot, 33
Positive feedback, 72
Problem-solving: annotate diagrams, 169
Problem-solving: check for exceptions, 170
Problem-solving: checking work, 170
Problem-solving: dimensional analysis, 169
Problem-solving: graph values, 170
Problem-solving: identify given data, 169
Problem-solving: identify relevant principles, 169
Problem-solving: interpret intermediate results, 169
Problem-solving: limiting cases, 5, 18, 21, 170
Problem-solving: qualitative to quantitative, 170
Problem-solving: quantitative to qualitative, 170
Problem-solving: reductio ad absurdum, 170
Problem-solving: simplify the system, 169
Problem-solving: thought experiment, 169
Problem-solving: track units of measurement, 169
Problem-solving: visually represent the system, 169
Problem-solving: work in reverse, 170
Programming, computer, 133
Protection diode, 122, 126
Python, 138

Qualitatively approaching a quantitative problem, 170
Quality factor, 71, 72, 74

Rail-to-rail input, 123
Rail-to-rail output, 121
Reactance, 17, 19, 29
Reading Apprenticeship, 147
Reductio ad absurdum, 170, 176, 177
Resonance, 19
Rolloff, 17, 81

Sallen-Key filter topology, 73
Schoenbach, Ruth, 147
Scientific method, 152
Second Law of Thermodynamics, 74
Siemens, 13
Siemens, Werner von, 13
Simplifying a system, 169

Socrates, 176
Socratic dialogue, 177
Source code, 134
SPICE, 147
Stallman, Richard, 179
Standard component values, IEC, 104
Stopband, 17, 81

Thévenin impedance, 19
Thévenin resistance, 127
Thought experiment, 169
Topology, 24
Torvalds, Linus, 179
Transfer function, 29, 32, 81
Transform function, 100
Treble, 16

Unit phasor, 13
Units of measurement, 169

Visualizing a system, 169
Voltage divider, 17, 32
Voltage follower, 126, 130
Voltage, common-mode, 123, 131

Whitespace, C++, 134, 135
Whitespace, Python, 141
Work in reverse to solve a problem, 170
WYSIWYG, 179, 180

Zero, 33, 81