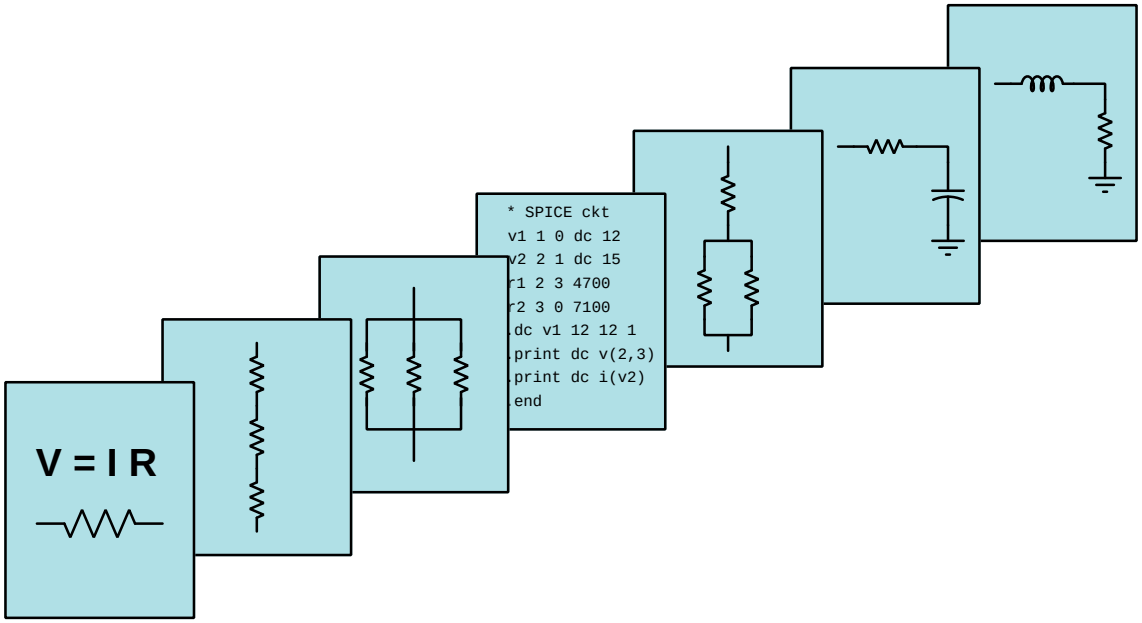
Modular Electronics Learning (ModEL) project



The S Variable

© 2018-2026 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 22 March 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 the *s* variable . . . 5
- 1.3 Recommendations for instructors . . . 6

2 Case Tutorial **7**
- 2.1 Example: series-parallel RLC impedances for complex frequencies . . . 8
- 2.2 Example: poles and zeroes of a series RC network . . . 12
- 2.3 Example: poles and zeroes of a series-parallel RC network . . . 15
- 2.4 Examples: gallery of pole-zero analyses for simple R, L, C networks . . . 18
- 2.5 Examples: gallery of pole-zero analyses for opamp networks . . . 26

3 Tutorial **33**
- 3.1 Impedance review . . . 34
- 3.2 Natural frequency, reactance, and impedance . . . 37
- 3.3 Complex frequency and impedance . . . 39
- 3.4 *s* variable summary . . . 46
- 3.5 Transfer functions . . . 48
- 3.6 AC network transfer function examples . . . 49
  - 3.6.1 Example: LR low-pass filter circuit . . . 50
  - 3.6.2 Example: RC high-pass filter circuit . . . 56
  - 3.6.3 Example: LC "tank" circuit . . . 61
  - 3.6.4 Example: RLC band-pass filter circuit . . . 66
- 3.7 Transfer function summary . . . 76

4 Derivations and Technical References **79**
- 4.1 Derivation of inductive impedance . . . 80
- 4.2 Derivation of capacitive impedance . . . 83
- 4.3 Definition of negative frequency . . . 86

5 Programming References **91**
- 5.1 Programming in C++ . . . 92
- 5.2 Programming in Python . . . 96
- 5.3 Impedance calculator with complex frequency using C++ . . . 101

5.4 Complex frequency plotter using C++ . . . 102
5.5 Pole-zero plotter using C++ . . . 113

6 Questions **117**
6.1 Conceptual reasoning . . . 121
6.1.1 Reading outline and reflections . . . 122
6.1.2 Foundational concepts . . . 123
6.1.3 Explain this joke . . . 125
6.1.4 Pole plots for pendula . . . 126
6.1.5 Pole plots for a feedback oscillator . . . 127
6.1.6 Pole plots for a bell . . . 128
6.1.7 Network zero conditions . . . 129
6.2 Quantitative reasoning . . . 130
6.2.1 Miscellaneous physical constants . . . 131
6.2.2 Introduction to spreadsheets . . . 132
6.2.3 Exploring Euler's Relation . . . 135
6.2.4 RC network impedances . . . 135
6.2.5 LR network impedances . . . 136
6.2.6 RLC network impedances . . . 137
6.2.7 HVDC filter harmonic analysis . . . 138
6.2.8 Determining s from oscillographs . . . 140
6.2.9 Transfer functions for simple networks . . . 143
6.2.10 Network response to different frequencies . . . 147

A Problem-Solving Strategies **149**

B Instructional philosophy **151**
B.1 First principles of learning . . . 152
B.2 Proven strategies for instructors . . . 153
B.3 Proven strategies for students . . . 155
B.4 Design of these learning modules . . . 156

C Tools used **159**

D Creative Commons License **163**

E References **171**

F Version history **173**

Index **175**

Chapter 1

Introduction

1.1 Recommendations for students

A powerful mathematical concept useful for analyzing practically any physical system – electrical circuits included – is something called the s variable. The s variable is closely related to Euler's Relation and phasor expressions of waveforms, which is why a discussion of it is included here.

Important concepts related to the s variable include **frequency**, **radians**, **exponential functions**, **real** versus **imaginary** versus **complex** numbers, **gain**, **impedance**, **differentiation** (calculus), **damped** response, **critical damping**, **quadratic formula**, **discriminant**, **source** versus **load**, **poles**, and **zeroes**.

Due to the heavy application of math associated with this concept, the reader is urged to apply the mathematical principles as soon as possible. The worked examples contained in the tutorial serves this purpose well, as they allow you to apply the formulae discussed previously to an example and then check your work against the completed example to see if your application was correct. Do not simply read a quantitative 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 demonstrate the existence of *zeroes* within an electrical network. 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 demonstrate the existence of *poles* within an electrical network. What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?
- What are some practical applications of transfer functions?
- What does it mean if a system is undamped, over-damped, or critically damped?

- What differentiates complex frequency from simple frequency in an AC circuit?
- Why is there such a thing as *natural frequency* measured in radians per second, as opposed to what seems to be the more clear and understandable concept of frequency measured in cycles per second?
- Why is $\frac{1}{j}$ equivalent to $-j$?
- What is the quadratic formula useful for in mathematics?
- What does a pole-zero plot represent?
- How does one add algebraic fractions with dissimilar denominators?
- 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?
- Under what condition(s) may an inductor exhibit an impedance phase angle other than $+90^o$?
- Under what condition(s) may a capacitor exhibit an impedance phase angle other than -90^o?
- How are Bode plots related to pole-zero plots?
- 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?
- What does it mean if a complex frequency has an imaginary value of zero?
- Of what significance is the *discriminant* of a polynomial expression?
- How does the concept of *time constant* relate to complex frequency?
- Do transfer functions have a unit of measurement? Why or why not?
- How might you alter one of the example analyses shown in the text, and then determine the behavior of that altered circuit?
- Devise your own question based on the text, suitable for posing to someone encountering this subject for the first time

1.2 Challenging concepts related to the s variable

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

- **Fractions** – of all the basic mathematical concepts, *fractions* seem to cause my students the greatest amount of trouble. Informal polls I've taken suggest a deficiency in our public-education math curriculum, where many students are encouraged to use electronic calculators to perform simple calculations, avoiding the use of fractions to perform mental arithmetic. Some students tell me fractions simply aren't emphasized much at all. This deficit causes much trouble when they must apply fractions to algebraic expressions, particularly when they must multiply or divide fractions together in order to manipulate an equation to solve for certain variables. As an instructor you should anticipate spending time in class reviewing and applying basic fractional arithmetic: how to add and subtract fractions, how to multiply and divide fractions, and how to reduce compound fractions to simpler forms. With regard to the application of fractions to equation manipulation, I recommend *multiplying by a fraction's reciprocal* in order to cancel out a fraction-multiplication as opposed to *dividing by that fraction*, since the division of one fraction by another naturally creates a compound fraction which then must be reduced in a subsequent step.

- **Gain calculations** – the gain for any electrical network is defined as the magnitude of its output signal to its input signal. Transfer functions are really just mathematical functions expressing the gain of any system with complex frequency (s) being the independent variable.

- **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.

The *Case Tutorial* chapter contains multiple sections showing how to develop transfer functions for various forms of passive and active (i.e. amplified) RLC networks.

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** – Apply the concept of poles and zeroes to simple systems

  Assessment – Sketch pole plots for simple mechanical systems; e.g. pose problems in the form of the "Pole plots for pendula" Conceptual Reasoning question.

  Assessment – Sketch pole plots for feedback oscillators; e.g. pose problems in the form of the "Pole plots for a feedback oscillator" Conceptual Reasoning question.

- **Outcome** – Apply the concept of complex frequency to real networks and captured signals

  Assessment – Calculate the total impedance value of a series-parallel RLC network for a given complex frequency; e.g. pose problems in the form of the "RC network impedances", "LR network impedances", and "RLC network impedances" Quantitative Reasoning questions.

  Assessment – Calculate the value of s describing a captured oscillograph for an oscillating signal that is also either growing or decaying in amplitude; e.g. pose problems in the form of the "Determining s from oscillographs" Quantitative Reasoning question.

- **Outcome** – Apply the concept of transfer functions to real networks

  Assessment – Write a mathematical transfer function for a given passive network with at least one reactive and at least one dissipative element; e.g. pose problems in the form of the "Transfer functions for simple networks" Quantitative Reasoning question.

Chapter 2

Case Tutorial

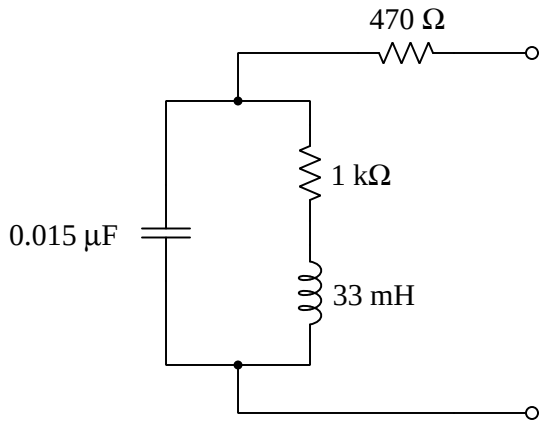
The idea behind a *Case Tutorial* is to explore new concepts by way of example. In this chapter you will read less presentation of theory compared to other Tutorial chapters, but by close observation and comparison of the given examples be able to discern patterns and principles much the same way as 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: series-parallel RLC impedances for complex frequencies

Network #1

Calculating the total impedance of the following network, assuming an excitation frequency of 5 kHz and a growth rate of 20,000 time constants per second:

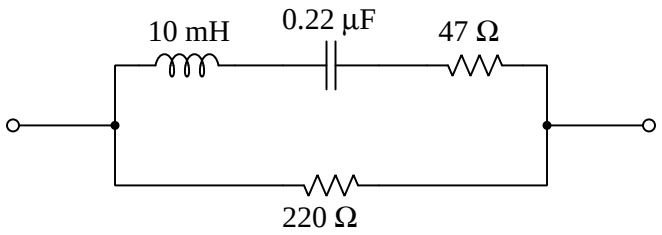


All values expressed in rectangular form:

$s = 20000 + j31415.9$ per second
$Z_L = 660 + j1036.72$ Ohms
$Z_C = 961.33 - j1510.06$ Ohms
$Z_{total} = 1738.7 - j346.98$ Ohms

Network #2

Calculating the total impedance of the following network, assuming an excitation frequency of 1 kHz and a decay rate of −4,500 time constants per second:

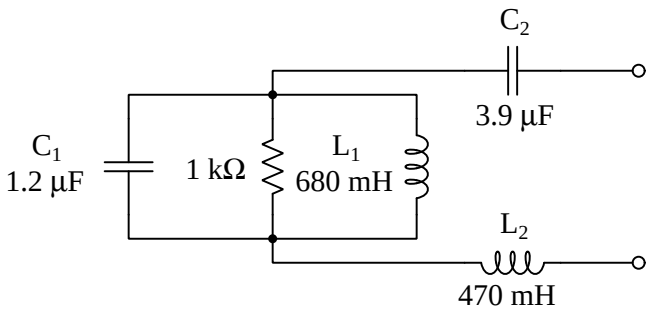


All values expressed in rectangular form:

$s = -4500.0 + j6283.19$ per second
$Z_L = -45.000 + j62.831$ Ohms
$Z_C = -342.46 - j478.16$ Ohms
$Z_{total} = 251.18 - j107.49$ Ohms

Network #3

Calculating the total impedance of the following network, assuming an excitation frequency of 400 Hz and a decay rate of −110 time constants per second:

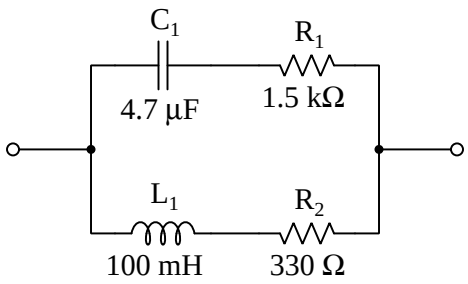


All values expressed in rectangular form:

$s = -110.00 + j2513.27$ per second
$Z_{L1} = -74.800 + j1709.03$ Ohms
$Z_{L2} = -51.700 + j1181.23$ Ohms
$Z_{C1} = -14.484 - j330.94$ Ohms
$Z_{C2} = -4.4567 - j101.83$ Ohms
$Z_{total} = 71.024 + j712.27$ Ohms

Network #4

Calculating the total impedance of the following network, assuming an excitation frequency of 60 Hz and a growth rate of 910 time constants per second:

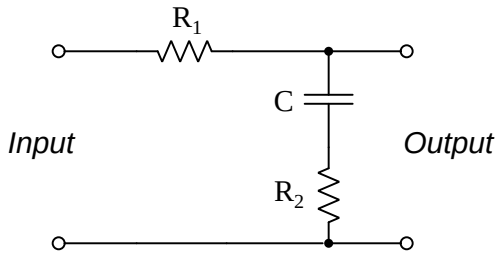


All values expressed in rectangular form:

$s = 910.00 + j376.99$ per second
$Z_{L1} = 91.000 + j37.699$ Ohms
$Z_{C1} = 199.55 - j82.673$ Ohms
$Z_{total} = 338.44 + j20.979$ Ohms

2.2 Example: poles and zeroes of a series RC network

Suppose we wished to determine the poles and zeroes of this RC network:



This network's transfer function may be written using the voltage divider equation, placing the series impedance of C and R_2 over the total network impedance as seen from the input terminals. First, writing an s-domain impedance equation for C and R_2:

$$Z_{series} = R_2 + \frac{1}{sC}$$

Now that we have an expression for the series impedance of R_2 and C, we may write the complete transfer function using the voltage divider formula format:

$$\frac{R_2 + \frac{1}{sC}}{R_1 + R_2 + \frac{1}{sC}}$$

Applying algebra to eliminate compound fractions:

$$\frac{\frac{sR_2C+1}{sC}}{\frac{sR_1C+sR_2C+1}{sC}}$$

$$\frac{sR_2C + 1}{sR_1C + sR_2C + 1}$$

$$\frac{sR_2C + 1}{sC(R_1 + R_2) + 1}$$

With this we have a tidy transfer function for the RC network.

A "zero" for any transfer function is a value of s resulting in an over-all value of zero for the transfer function. Since s appears in the numerator of this particular transfer function, we may solve for this zero-numerator condition by setting the numerator equal to zero and solving for s:

$$sR_2C + 1 = 0$$

$$sR_2C = -1$$

$$s = -\frac{1}{R_2C}$$

Therefore, this network's "zero" is that complex frequency equal to $-\frac{1}{R_2C}$. Since both R_2 and C are real numbers, the quotient $-\frac{1}{R_2C}$ will also be a real number, meaning the "zero" for this network's transfer function will be a specific decay rate based on the values of R_2 and C. We know it is a *decay* rather than a *growth* because with positive R_2 and C values the quotient $-\frac{1}{R_2C}$ will be negative.

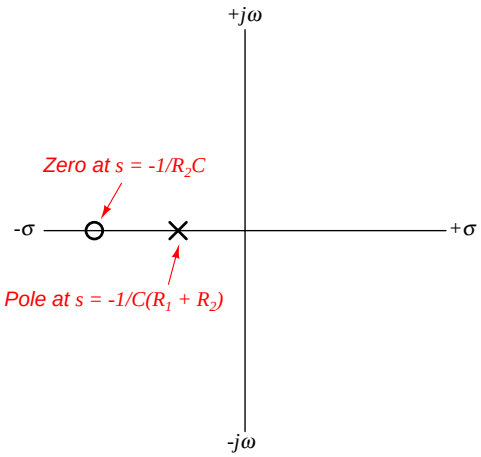
A "pole" for any transfer function is a value of s resulting in an over-all value of infinity for the transfer function; i.e. a value of s that results in the transfer function's denominator being zero. Setting the denominator equal to zero and solving for s allows us to solve for this function's pole:

$$sC(R_1 + R_2) + 1 = 0$$

$$sC(R_1 + R_2) = -1$$

$$s = -\frac{1}{C(R_1 + R_2)}$$

Just like the function's zero, this pole is also a real number with a negative value. Given the pole's larger denominator ($C(R_1 + R_2)$ versus R_2C) we can tell the pole will lie closer to the origin of the pole-zero plot than the zero will:



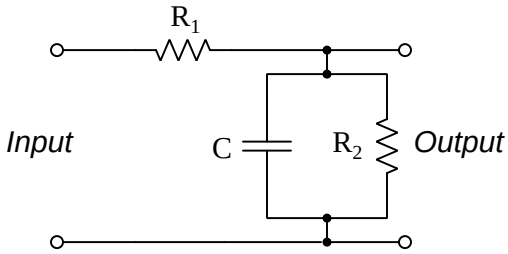
If we happen to know the actual component values for C, R_1, and R_2, we may compute complex frequency values for the network's zero and pole. Supposing C is 10 microFarads, R_1 is 1 kiloOhm, and R_2 is 2.2 kiloOhms, the zero and pole frequencies for this particular network will be as follows[1]:

- Zero frequency $s = -45.45$ per second (s^{-1})
- Pole frequency $s = -31.25$ per second (s^{-1})

[1] Just to be clear, s is the mathematical symbol for complex frequency while s^{-1} is its unit of measurement in inverse-seconds, or "per second". Using the same alphabetical letter for both the symbol and its unit of measurement is similar to how we express voltage, where the letter "V" is used both as the mathematical symbol and the abbreviation for its unit of measurement (Volts). The real portion of a complex frequency is measured in time constants per second (τ/s) while the imaginary portion of a complex frequency is measured in radians per second (rad/s). Since these pole values are wholly real, we could have specified the unit as "time constants per second", but for any complex frequency containing both real and imaginary portions the most sensible unit is simply "per second" ($\frac{1}{s}$ or more simply s^{-1}) which makes it a more generic unit of measurement.

2.3 Example: poles and zeroes of a series-parallel RC network

Suppose we wished to determine the poles and zeroes of this RC network:



This network's transfer function may be written using the voltage divider equation, placing the parallel impedance of C and R_2 over the total network impedance as seen from the input terminals. First, writing an s-domain impedance equation for C and R_2:

$$Z_{parallel} = \frac{1}{\frac{1}{R_2} + sC}$$

$$Z_{parallel} = \frac{1}{\frac{1}{R_2} + \frac{sR_2C}{R_2}}$$

$$Z_{parallel} = \frac{1}{\frac{1+sR_2C}{R_2}}$$

$$Z_{parallel} = \frac{R_2}{1 + sR_2C}$$

Now that we have an expression for the parallel impedance of R_2 and C, we may write the complete transfer function using the voltage divider formula format:

$$\frac{\frac{R_2}{1+sR_2C}}{\frac{R_2}{1+sR_2C} + R_1}$$

Applying algebra to eliminate compound fractions:

$$\frac{\frac{R_2}{1+sR_2C}}{\frac{R_2}{1+sR_2C} + \frac{R_1(1+sR_2C)}{1+sR_2C}}$$

$$\frac{\frac{R_2}{1+sR_2C}}{\frac{R_2}{1+sR_2C} + \frac{R_1+sR_1R_2C}{1+sR_2C}}$$

$$\frac{\frac{R_2}{1+sR_2C}}{\frac{R_2+R_1+sR_1R_2C}{1+sR_2C}}$$

$$\frac{R_2}{R_2 + R_1 + sR_1R_2C}$$

This transfer function's zeroes occur at infinite values of s, because only a signal with infinite frequency will cause the transfer function's value to become zero (i.e. a finite numerator and an infinite denominator). Examining the circuit schematic, we see this makes perfect sense: the only way to achieve zero-voltage output for this circuit is to apply an AC signal with an infinite frequency which will cause the capacitor to behave as a *short*. Note that with s being a complex number, infinite frequency can mean infinitely high oscillations ($\omega = \infty$) and/or infinitely fast growth/decay rates ($\sigma = \infty$).

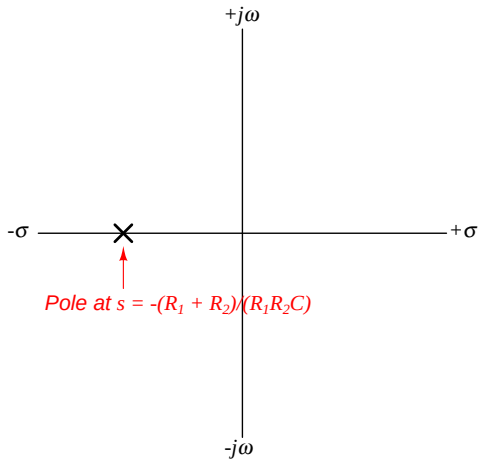
This transfer function has a finite pole, which may be found by solving for the value of s yielding a zero denominator which will make the over-all transfer function's value infinite:

$$R_2 + R_1 + sR_1R_2C = 0$$

$$sR_1R_2C = -(R_1 + R_2)$$

$$s = -\frac{R_1 + R_2}{R_1R_2C}$$

Since all the variables on the right-hand side of the equation are real, it means s for the network's pole will have a real value. This means s is all σ and no $j\omega$, indicating a DC signal condition. The negative sign indicates the DC signal will be decaying rather than growing. Shown on a pole-zero plot (with no zeroes shown because they are infinite in value and therefore off the scale of each axis):



Poles represent conditions of zero input yielding non-zero output, which for this circuit would mean short-circuiting the input terminals together with a pre-energized capacitor. Doing that would place the two resistors in parallel with each other, resulting in an RC time constant of $\tau = \frac{R_1 R_2 C}{R_1 + R_2}$ using the product-over-sum parallel resistance formula. Since we know σ is the reciprocal of τ, it should not surprise us to see the pole's decay rate being the reciprocal of $\frac{R_1 R_2 C}{R_1 + R_2}$.

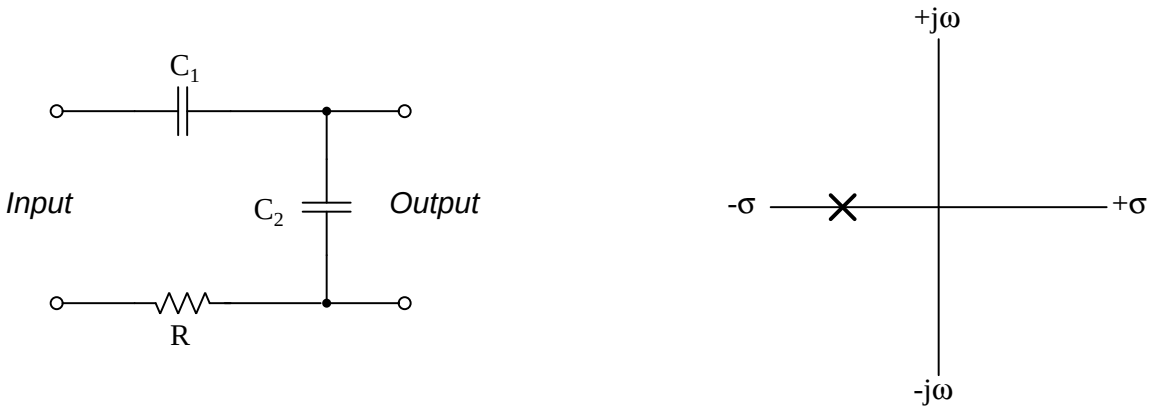
If we happen to know the actual component values for C, R_1, and R_2, we may compute complex frequency values for the network's zero and pole. Supposing C is 3.3 microFarads, R_1 is 4.7 kiloOhms, and R_2 is 1.2 kiloOhms, the zero and pole frequencies for this particular network will be as follows[2]:

- Zero frequency $s = \infty$ per second (s^{-1})
- Pole frequency $s = -317.00$ per second (s^{-1})

[2] Just to be clear, s is the mathematical symbol for complex frequency while s^{-1} is its unit of measurement in inverse-seconds, or "per second". Using the same alphabetical letter for both the symbol and its unit of measurement is similar to how we express voltage, where the letter "V" is used both as the mathematical symbol and the abbreviation for its unit of measurement (Volts). The real portion of a complex frequency is measured in time constants per second (τ/s) while the imaginary portion of a complex frequency is measured in radians per second (rad/s). Since these pole values are wholly real, we could have specified the unit as "time constants per second", but for any complex frequency containing both real and imaginary portions the most sensible unit is simply "per second" ($\frac{1}{s}$ or more simply s^{-1}) which makes it a more generic unit of measurement.

2.4 Examples: gallery of pole-zero analyses for simple R, L, C networks

Network #1



$$\text{Transfer function} = \frac{C_1}{sRC_1C_2 + C_1 + C_2}$$

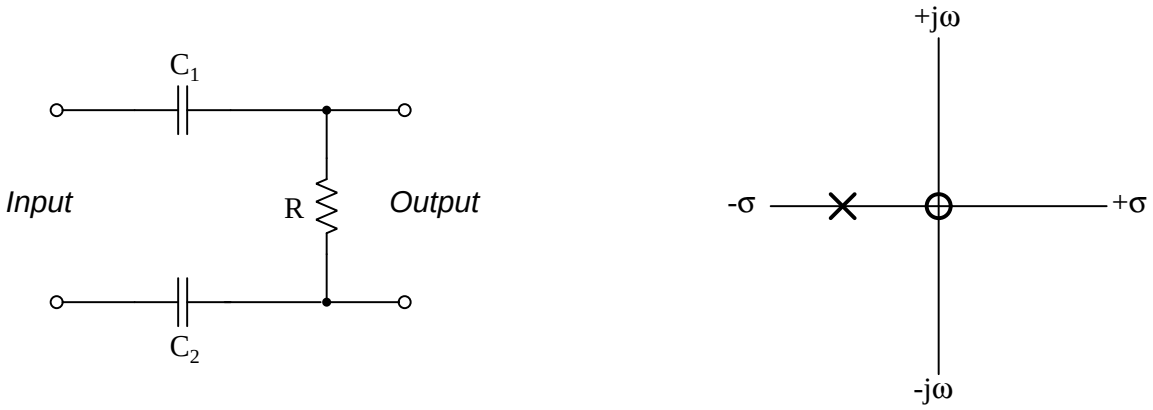
$$\text{Zero at } s = \infty$$

$$\text{Pole at } s = -\frac{C_1 + C_2}{RC_1C_2}$$

If $C_1 = 33$ μF and $C_2 = 47$ μF and $R = 2.2$ kΩ, then:

- Zero frequency $s = \infty$ per second (s^{-1})
- Pole frequency $s = -23.445$ per second (s^{-1})

Network #2



$$\text{Transfer function} = \frac{sRC_1C_2}{sRC_1C_2 + C_1 + C_2}$$

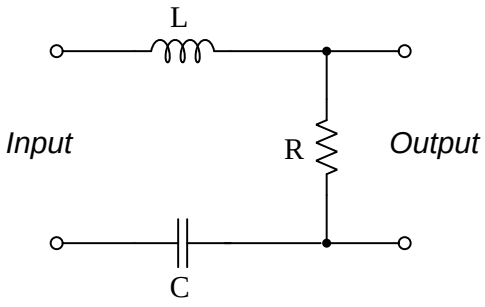
$$\text{Zero at } s = 0$$

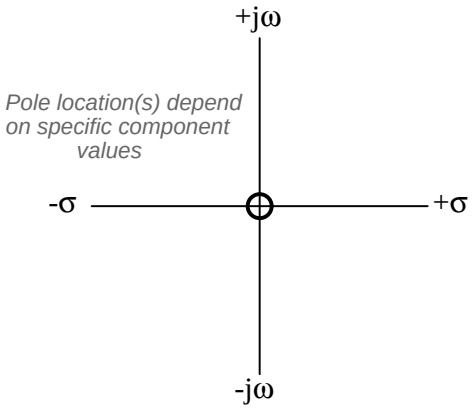
$$\text{Pole at } s = -\frac{C_1 + C_2}{RC_1C_2}$$

If $C_1 = 1.5$ μF and $C_2 = 2.2$ μF and $R = 4.7$ kΩ, then:

- Zero frequency $s = 0$ per second (s^{-1})
- Pole frequency $s = -238.56$ per second (s^{-1})

Network #3





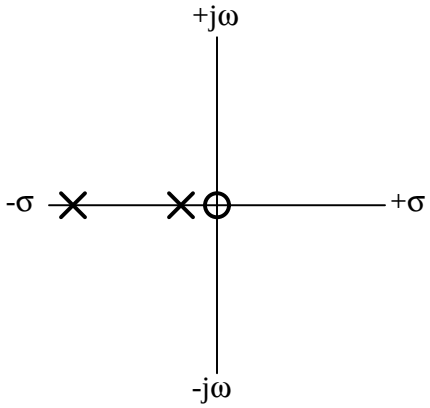
$$\text{Transfer function} = \frac{sRC}{s^2LC + sRC + 1}$$

$$\text{Zero at } s = 0$$

$$\text{Pole(s) at } s = \frac{-RC \pm \sqrt{R^2C^2 - 4LC}}{2LC}$$

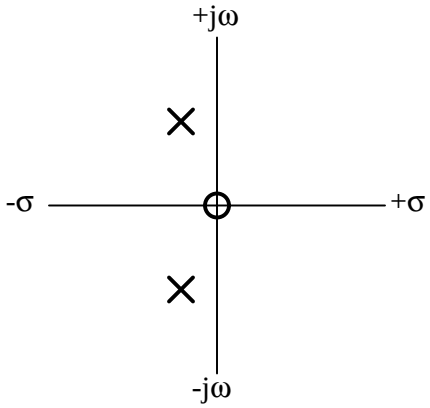
If $L = 100$ mH and $C = 68$ μF and $R = 1.5$ kΩ, then we have an *over-damped* condition:

- Zero frequency $s = 0$ per second (s^{-1})
- Pole frequency $s = -9.8103$ per second (s^{-1})
- Pole frequency $s = -14990.2$ per second (s^{-1})



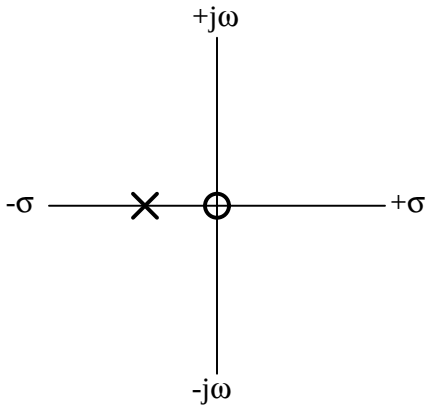
If $L = 100$ mH and $C = 68$ μF and $R = 27$ Ω, then we have an *under-damped* condition:

- Zero frequency $s = 0$ per second (s^{-1})
- Pole frequency $s = -135 + j358.9$ per second (s^{-1})
- Pole frequency $s = -135 - j358.9$ per second (s^{-1})



If $L = 100$ mH and $C = 68$ μF and $R = 76.6965$ Ω, then we have a *critically damped* condition:

- Zero frequency $s = 0$ per second (s^{-1})
- Pole frequency $s = -383.4825$ per second (s^{-1})



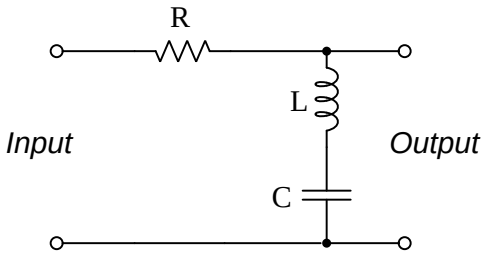
Whether an RLC network is over-damped, under-damped, or critically damped depends on the mathematical sign of the *discriminant*, that portion of the quadratic equation contained within the radicand:

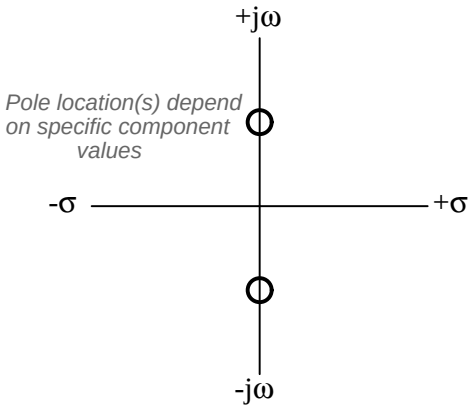
$$\text{Discriminant} = R^2C^2 - 4LC$$

If the discriminant is positive, then the system will be over-damped with two real poles only. If the discriminant is negative, then the system will be under-damped with complex poles. Only if the discriminant is zero will there only be a single pole (or if you prefer, two identical poles with $\pm\ j0$ imaginary portions) lying on the real axis of the pole-zero plot.

Under-damped systems are capable of oscillations under zero-input conditions, albeit damped oscillations eventually settling out at a zero-energy condition as the stored energy within the capacitor and inductor eventually dissipates through the resistor. Neither over-damped nor critically damped systems will oscillate unless driven by an oscillating (AC) signal at the input.

Network #4





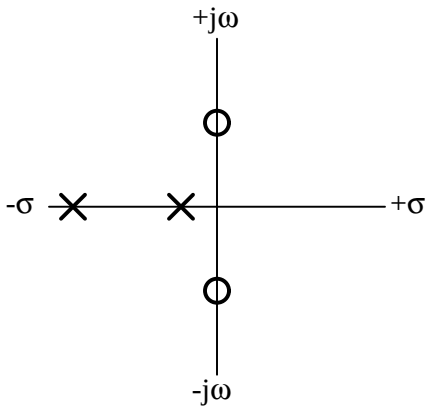
$$\text{Transfer function} = \frac{s^2LC + 1}{s^2LC + sRC + 1}$$

$$\text{Zeroes at } s = \pm j\sqrt{\frac{1}{LC}}$$

$$\text{Pole(s) at } s = \frac{-RC \pm \sqrt{R^2C^2 - 4LC}}{2LC}$$

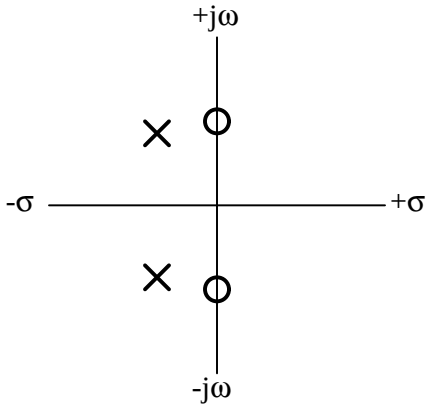
If $L = 100$ mH and $C = 68$ μF and $R = 1.5$ kΩ, then we have an *over-damped* condition:

- Zero frequency $s = +j383.48$ per second (s^{-1})
- Zero frequency $s = -j383.48$ per second (s^{-1})
- Pole frequency $s = -9.8103$ per second (s^{-1})
- Pole frequency $s = -14990.2$ per second (s^{-1})



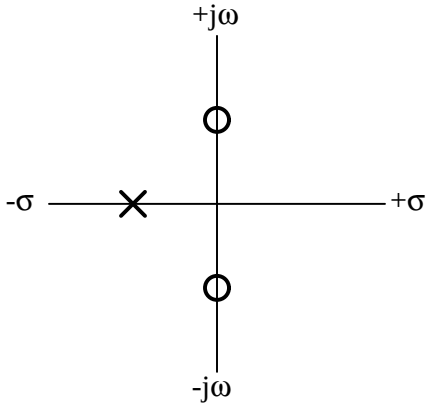
If $L = 100$ mH and $C = 68$ μF and $R = 27$ Ω, then we have an *under-damped* condition:

- Zero frequency $s = +j383.48$ per second (s^{-1})
- Zero frequency $s = -j383.48$ per second (s^{-1})
- Pole frequency $s = -135 + j358.9$ per second (s^{-1})
- Pole frequency $s = -135 - j358.9$ per second (s^{-1})



If $L = 100$ mH and $C = 68$ μF and $R = 76.6965$ Ω, then we have a *critically damped* condition:

- Zero frequency $s = +j383.48$ per second (s^{-1})
- Zero frequency $s = -j383.48$ per second (s^{-1})
- Pole frequency $s = -383.4825$ per second (s^{-1})



Whether an RLC network is over-damped, under-damped, or critically damped depends on the mathematical sign of the *discriminant*, that portion of the quadratic equation contained within the radicand:

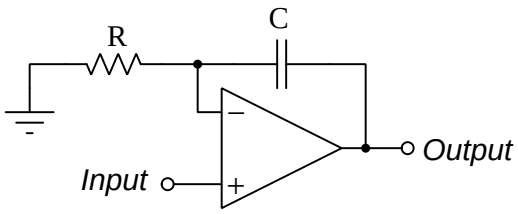
$$\text{Discriminant} = R^2C^2 - 4LC$$

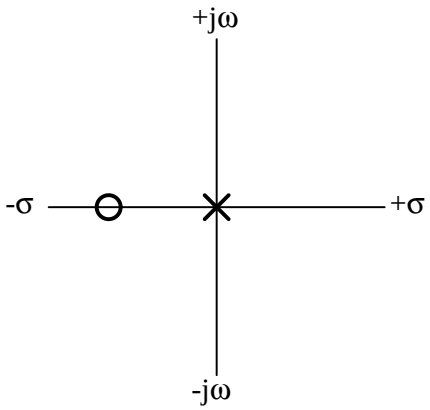
If the discriminant is positive, then the system will be over-damped with two real poles only. If the discriminant is negative, then the system will be under-damped with complex poles. Only if the discriminant is zero will there only be a single pole (or if you prefer, two identical poles with $\pm$ $j0$ imaginary portions) lying on the real axis of the pole-zero plot.

Under-damped systems are capable of oscillations under zero-input conditions, albeit damped oscillations eventually settling out at a zero-energy condition as the stored energy within the capacitor and inductor eventually dissipates through the resistor. Neither over-damped nor critically damped systems will oscillate unless driven by an oscillating (AC) signal at the input.

2.5 Examples: gallery of pole-zero analyses for opamp networks

Non-inverting integrator





$$\text{Transfer function} = \frac{sRC + 1}{sRC}$$

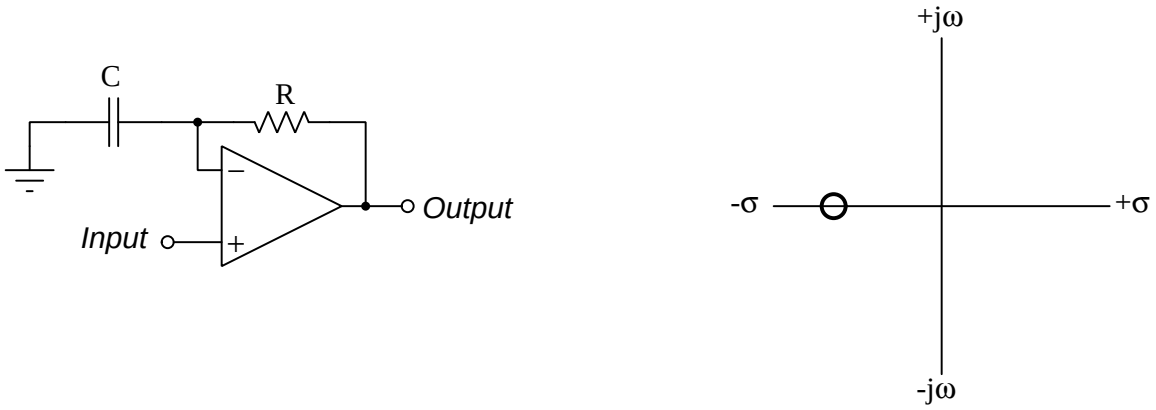
$$\text{Zero at } s = -\frac{1}{RC}$$

$$\text{Pole at } s = 0$$

If $R = 1.2$ kΩ and $C = 0.33$ μF, then:

- Zero frequency s = -2525.3 per second (s^{-1})
- Pole frequency $s = 0$ per second (s^{-1})

Non-inverting differentiator



$$\text{Transfer function} = sRC + 1$$

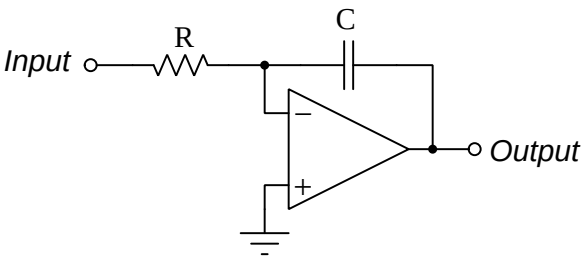
$$\text{Zero at } s = -\frac{1}{RC}$$

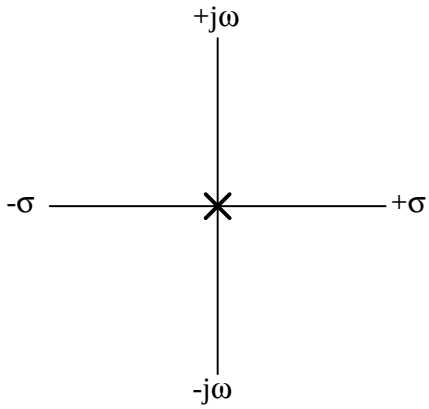
$$\text{Pole at } s = \infty$$

If $R = 6.8$ kΩ and $C = 2.2$ μF, then:

- Zero frequency s = -66.845 per second (s^{-1})
- Pole frequency $s = \infty$ per second (s^{-1})

Inverting integrator





$$\text{Transfer function} = \frac{1}{sRC}$$

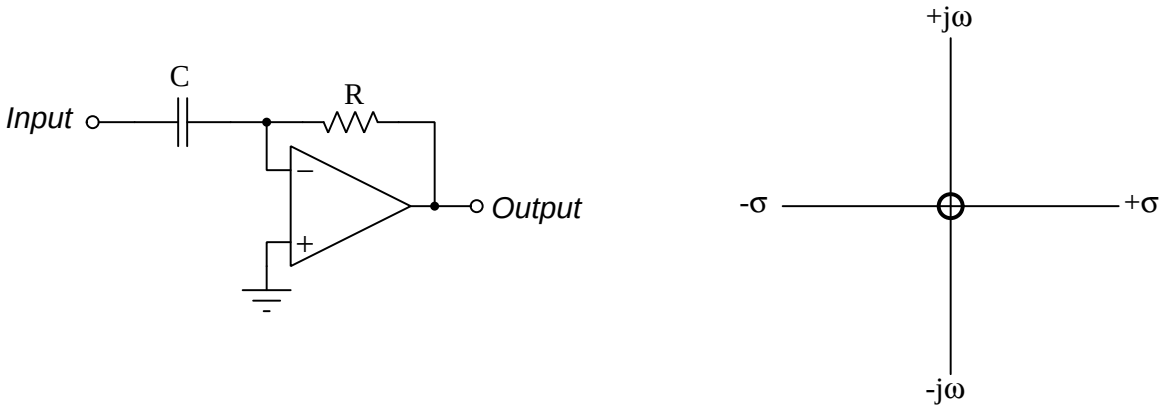
Zero at $s = \infty$

Pole at $s = 0$

Regardless of component values:

- Zero frequency $s = \infty$ per second (s^{-1})
- Pole frequency $s = 0$ per second (s^{-1})

Inverting differentiator



Transfer function $= sRC$

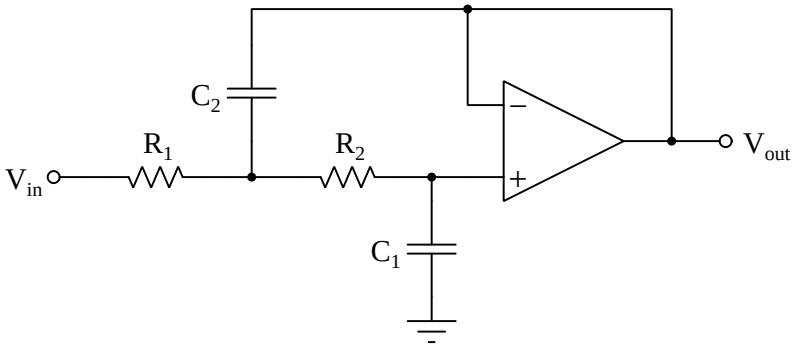
Zero at $s = 0$

Pole at $s = \infty$

Regardless of component values:

- Zero frequency $s = 0$ per second (s^{-1})
- Pole frequency $s = \infty$ per second (s^{-1})

Sallen-Key low-pass filter

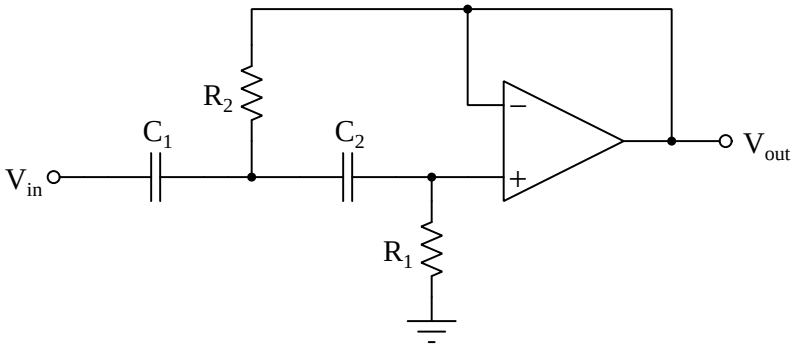


$$\text{Transfer function} = \frac{1}{s^2 C_1 C_2 R_1 R_2 + s C_1 (R_1 + R_2) + 1}$$

$$\text{Zero at } s = \infty$$

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

Sallen-Key high-pass filter



$$\text{Transfer function} = \frac{s^2 C_1 C_2 R_1 R_2}{s^2 C_1 C_2 R_1 R_2 + s R_2 (C_1 + C_2) + 1}$$

$$\text{Zero at } s = 0$$

$$\text{Pole(s) at } s = \frac{-R_2(C_1 + C_2) \pm \sqrt{\left(R_2(C_1 + C_2)\right)^2 - 4 C_1 C_2 R_1 R_2}}{2 C_1 C_2 R_1 R_2}$$

Chapter 3

Tutorial

3.1 Impedance review

Electrical engineering makes liberal use of calculus, which is the branch of mathematics specializing in *changes*. This makes sense, as components such as inductors and capacitors naturally exhibit behaviors perfectly described by such changes:

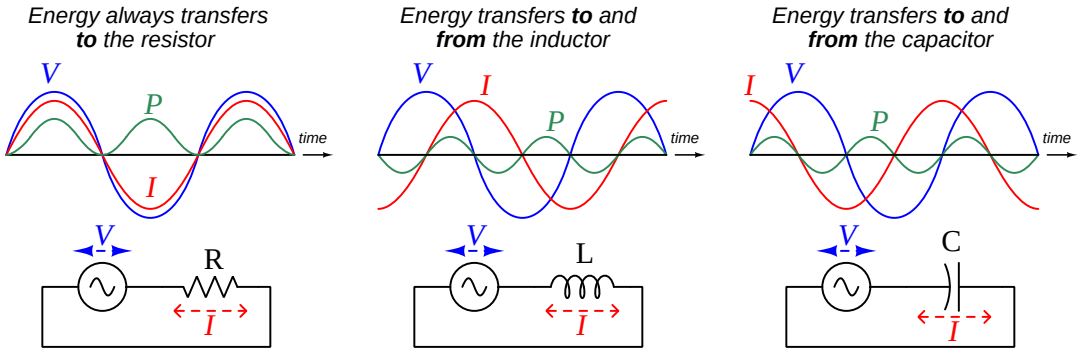
$$V = L\frac{dI}{dt} \qquad\qquad I = C\frac{dV}{dt}$$

Where,

V = voltage in Volts (V)
I = current in Amperes (A)
L = inductance in Henrys (H)
C = capacitance in Farads (F)
$\frac{dI}{dt}$ = rate-of-change of current in Amperes per second (A/s)
$\frac{dV}{dt}$ = rate-of-change of voltage in Volts per second (V/s)

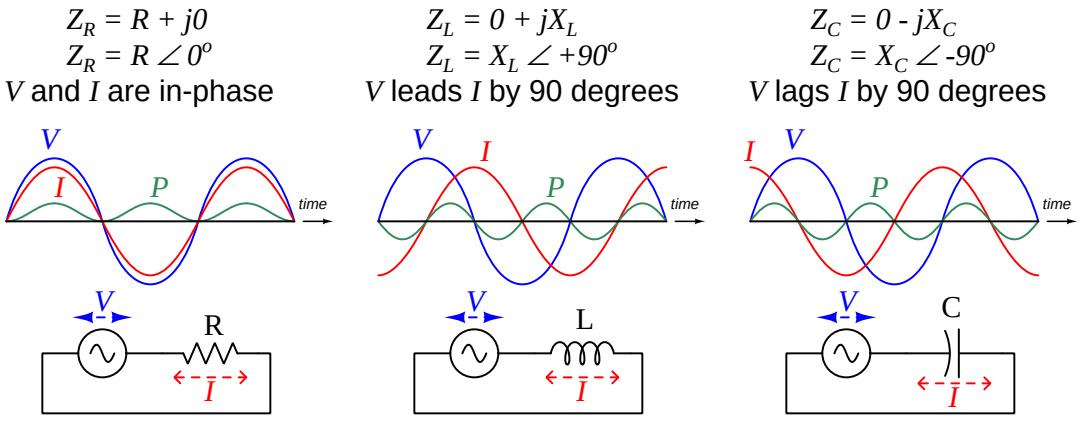
These two formulae may be thought of as "Ohm's Law" for inductors and capacitors, respectively, since they describe the relationship between voltage and current for each component type, much like the real Ohm's Law describes the relationship between voltage and current for a resistance: $V = IR$. For resistance that relationship is a simple proportionality. For inductors and capacitors, however, the relationship between voltage and current involves the rate-of-change-over-time (i.e. the "speed") of one compared to the immediate value of the other.

When we energize these three different types of components with *alternating* (AC) electricity, we see the importance of these mathematical distinctions. Examine the oscillographs below, and note how with resistors the AC voltage and current are always in-step (i.e. in-phase) with each other due to the constant proportionality we call "resistance", but for inductors and capacitors the rate-of-change of one determines the immediate value of the other. Specifically, note how inductor voltage peaks at those points in time where the current wave is *steepest*, and how capacitor current peaks where the voltage wave is steepest. The fact that one wave's values is directly related to the rate-of-change of the other wave explains why voltage and current are always phase-shifted by a quarter-cycle (90^o) for both capacitors and inductors:



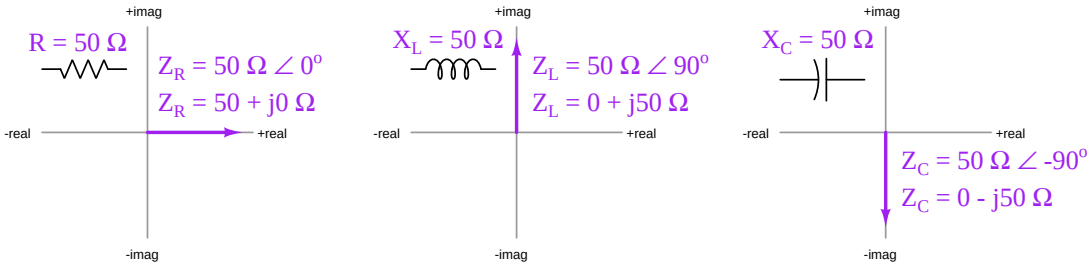
As usual, the analysis of energy proves insightful to these components. Resistors always dissipate electrical energy, mostly in the form of heat where it leaves the circuit never to return, and this is why power is always a positive value for an energized resistor. Inductors and capacitors, however, ideally dissipate no energy at all but instead absorb and release it periodically. Periods of time where power (P) is positive represent the absorption of energy from the circuit by that component, and periods of negative power represent energy released by the component to the rest of the circuit. The degree to which inductors and capacitors react to changes in voltage and current is called *reactance* (X), which we express in the unit of the Ohm just like resistance (R).

In order to apply foundational principles such as Ohm's Law, Kirchhoff's Voltage Law, Kirchhoff's Current Law, Maximum Power Transfer Theorem, Thévenin's Theorem, Norton's Theorem, and Superposition Theorem to AC circuits containing inductance and/or capacitance, it is helpful to quantify the opposition each type of component offers to the passage of electric current. To this end, we have the concept of *impedance* (Z) which encompasses both resistance and reactance, using *complex numbers* to do so.



Impedance, by definition, is the ratio of voltage to current, and so the phase angle of a complex impedance expresses the phase relationship of voltage to current for that component or network of components. Resistance always has an impedance phase angle of zero because voltage and current are always perfectly in-phase for any pure resistance, with zero shift between those two waveforms. Inductors always exhibit a $+90^o$ impedance phase angle because the voltage waveform always leads the current waveform by 90 degrees. For capacitors the impedance phase angle is always -90^o because the voltage waveform lags behind the current waveform by 90 degrees.

Complex-number impedance values may be represented graphically in the form of *phasor diagrams*, where the length of a vector arrow represents the phasor's magnitude and its angle represents phase shift, the horizontal axis representing the real number line and the vertical axis representing the imaginary number (j) line. 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:



Complex numbers contain both a *real* and an *imaginary* term, the real term representing the amount of "resistive" energy dissipation offered by the component or network of components, and the imaginary term representing the "reactive" energy absorption and release. In mechanical terms these may be likened to *friction* and *inertia*, respectively. This imaginary term is marked by the presence of j (by technologists) or i (by mathematicians), equal to the square root of negative one ($j = \sqrt{-1}$).

Introductory AC theory describes how to calculate the reactance of inductors and capacitors as a function of signal frequency ($X_L = 2\pi f L$ and $X_C = \frac{1}{2\pi f C}$ respectively), and then to express these as complete impedances either in rectangular form ($R \pm jX$) or in polar form ($Z \angle \theta$). What we will explore in subsequent sections of this Tutorial is how frequency itself may be represented as an imaginary, real, or complex number, and the various applications of this fact to circuit analysis.

3.2 Natural frequency, reactance, and impedance

Most students of electronics learn the formulae $X_L = 2\pi fL$ and $X_C = \frac{1}{2\pi fC}$ when first encountering the concepts of inductive and capacitive reactance, and some of them wonder where the 2π coefficient comes from. This is actually a *conversion factor* for frequency, to convert the common unit of Hertz (i.e. cycles per second) into the more mathematically "natural" unit of *radians per second*. *Angular velocity* or *natural frequency* expressed in radians per second is always symbolized by the lower-case Greek letter omega (ω) rather than f which implies *frequency* in cycles per second (Hertz). Both, however, describe how rapidly a waveform such as an alternating voltage or an alternating current repeatedly cycles over time.

Re-writing both of these reactance formulae to use ω instead of $2\pi f$, we immediately see just how much simpler they appear:

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

Extending these just one step further to represent *impedance* rather than merely reactance by including the imaginary operator $j = \sqrt{-1}$:

$$Z_L = j\omega L \qquad\qquad Z_C = \frac{1}{j\omega C} \quad \text{or} \quad Z_C = \frac{-j}{\omega C}$$

Refer to section 4.1 beginning on page 80 for inductance and section 4.2 beginning on page 83 for capacitance to see calculus-based derivations of these formulae. The simple justification for using ω as frequency is that the calculus operations of differentiation and integration are relatively easy to implement on imaginary-exponential functions (i.e. any function based on e^j), but more difficult to implement on trigonometric functions such as sine and cosine. Therefore, when starting with the "Ohm's Law" relations of voltage versus current for inductors ($V_L = L\frac{dI}{dt}$) and capacitors ($I_C = C\frac{dV}{dt}$) as a basis for deriving impedance, it makes practical sense to cast the waveform functions in imaginary-exponential terms because the ensuing calculus will be easier[1]. However, imaginary-exponential expressions of sinusoidal waveforms assume angles in radians rather than degrees (or complete cycles). Since there are exactly 2π radians in one complete cycle, one cycle per second (1 Hz) is equivalent to 2π radians per second.

Let's compare reactance calculations using f versus using ω. Applying 800 Hertz to a 47 milliHenry inductor yields 236.248 Ohms of reactance using the f-based formula:

$$X_L = 2\pi fL = 2\pi(800 \text{ Hz})(0.047 \text{ H}) = 236.248\,\Omega$$

If we convert 800 Hertz into radians per second ($\omega = 2\pi f = 5026.548$ radians per second) we arrive at the exact same 236.248 Ohms of inductive reactance using the ω-based formula:

$$X_L = \omega L = (5026.548 \text{ rad/s})(0.047 \text{ H}) = 236.248\,\Omega$$

[1] For example, the current through a capacitor energized by a voltage $e^{j\omega t}$ is C $\frac{d}{dt}\left(e^{j\omega t}\right)$ which is equal to $j\omega Ce^{j\omega t}$. Impedance is the ratio of voltage to current ($Z = \frac{V}{I}$), and so dividing that voltage ($e^{j\omega t}$) by the capacitor's current ($j\omega Ce^{j\omega t}$) simply results in $Z_C = \frac{1}{j\omega C}$.

Here we have a table comparing common mathematical expressions of electrical quantities, f-based versus ω-based:

Concept	**f-based**	**ω-based**
Inductive reactance	$X_L = 2\pi f L$	$X_L = \omega L$
Capacitive reactance	$X_C = \frac{1}{2\pi f C}$	$X_C = \frac{1}{\omega C}$
Inductive impedance	$Z_L = j2\pi f L$	$Z_L = j\omega L$
Capacitive impedance	$Z_C = \frac{1}{j2\pi f C}$	$Z_C = \frac{1}{j\omega C}$
Resonance	$f = \frac{1}{2\pi\sqrt{LC}}$	$\omega = \frac{1}{\sqrt{LC}}$

Clearly, formulae written in terms of ω instead of f are more compact and therefore preferred whenever the impedance of AC networks containing many inductors, capacitors, and/or resistors must be expressed in concise mathematical form.

One final note regarding capacitive impedance and the sign of j. While it is true that $Z_C = \frac{-j}{\omega C}$, it is equally true that $Z_C = \frac{1}{j\omega C}$. Showing the $-j$ term makes it more clear that the phase angle of a pure capacitance's impedance will be -90^o, but placing the j term in the denominator actually means the same thing. The following steps prove this mathematical equivalence:

$$Z_C = \frac{1}{j\omega C}$$

$$Z_C = \left(\frac{j}{j}\right)\left(\frac{1}{j\omega C}\right)$$

$$Z_C = \frac{j}{j^2\omega C}$$

$$Z_C = \frac{j}{(-1)\omega C}$$

$$Z_C = \frac{-j}{\omega C} \quad \text{or} \quad Z_C = -j\frac{1}{\omega C}$$

When writing complicated formulae describing networks containing capacitance it is customary to leave j as a positive quantity in the denominator, if for no other reason than to avoid inserting mathematically-unnecessary negative signs which make the resulting formulae easier to manipulate and solve. However, it is good to know that doing so still leaves the capacitive impedance with a negative phase angle even if that is not immediately apparent by inspection.

As we have seen, expressing the reactance and/or impedance of capacitors and inductors in terms of natural frequency (ω) is mathematically elegant, as the ω-based expressions have no need for the 2π conversion factor necessary to convert cycles per second (Hz) of conventional frequency into radians per second of natural frequency.

3.3 Complex frequency and impedance

Natural frequency (ω) expressed in units of radians per second may seem unnecessarily complicated when ordinary frequency (f) expressed in units of cycles per second (Hz) seems most intuitive, despite reactance formulae containing fewer symbols using ω instead of f. A strong mathematical justification for ω, however, is its utility when used with imaginary-exponential functions. To understand why, we must review a very important and foundational equation in mathematics known as Euler's Relation:

$$e^{jx} = \cos x + j \sin x$$

Where,
e = Euler's number (approximately equal to 2.718281828)
j = The "imaginary operator" equal to $\sqrt{-1}$
x = Angle expressed in radians

This Relation shows the fundamental equivalence between two seemingly unrelated classes of functions – exponential functions and trigonometric functions. However, it is true only when the angle in question x is cast in units of radians. If we wish to describe the evolution of a sinusoidal wave in terms of frequency and time, we must use natural frequency in units of radians per second, multiplied ω by time (t, in seconds):

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$$

Where,
ω = Natural frequency in radians per second
t = Time in seconds

Exponential functions (e^x) are also useful for describing things other than cosine and sine waves. Notably, they are also used to describe natural growth and decay processes, using *real* numbers rather than imaginary numbers in the exponential term:

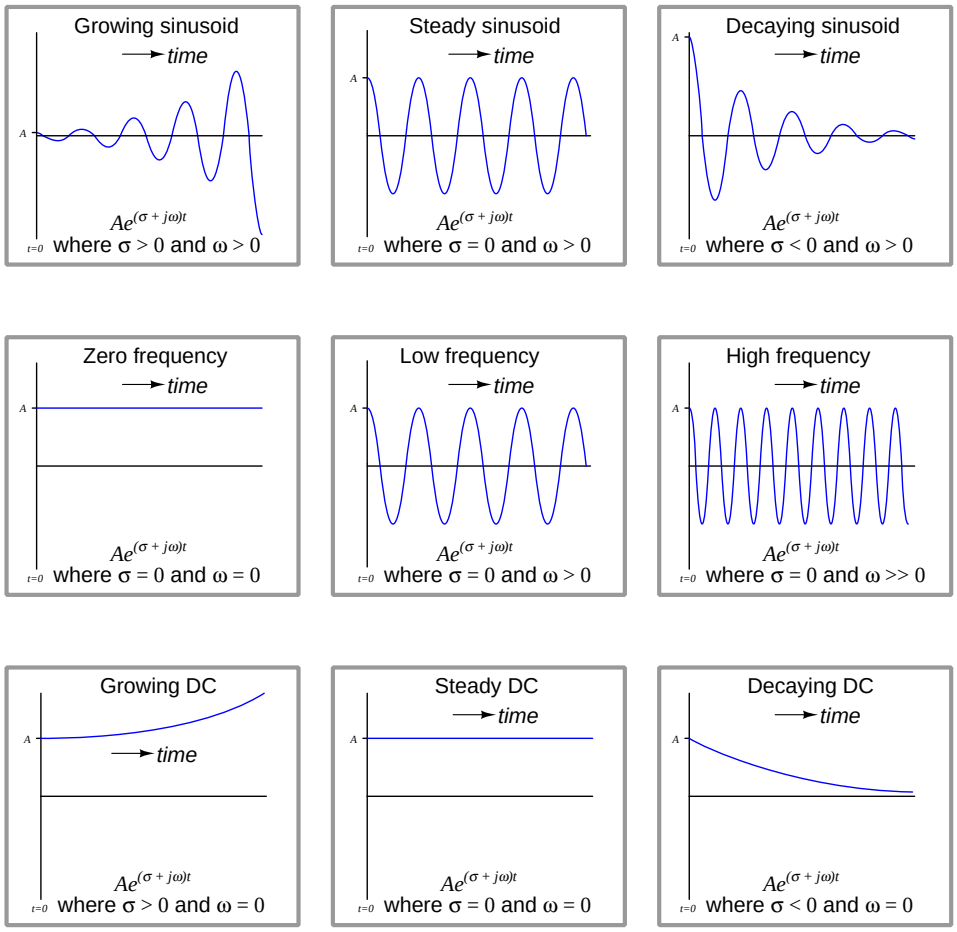
$$e^{\sigma t}$$

Where,
σ = Growth/decay rate in time constants per second ($\frac{1}{\text{sec}}$), also known as *Neper velocity*
t = Time in seconds

As every student of electronics learns, the decay of DC voltage and/or current for a de-energizing capacitor or inductor follows $e^{-\frac{t}{\tau}}$ where t is time and τ is the *time-constant* of the circuit, both measured in seconds. If we look at this familiar formula closely we will see that τ is really nothing more than the reciprocal of the growth/decay rate labeled by the lower-case Greek letter "sigma" (σ) shown above. In other words, $\tau = \frac{1}{\sigma}$ which makes $e^{\sigma t}$ equivalent to $e^{\frac{t}{\tau}}$. In fact, we may legitimately apply the unit of *time constants per second* to this rate. Thus, $e^{\sigma t}$ more generally describes anything growing exponentially in magnitude over time if σ is positive in value, or decaying toward zero over time if σ is negative. A zero value for σ means the quantity is stable over time, neither growing nor decaying.

If σ as a real number describes growth/decay rate in "time-constants per second" units (Neper velocity) and ω as an imaginary number ($j\omega$) describes the natural frequency (or angular velocity) of a sinusoid in "radians per second" units, we may combine these two values into a single complex-number sum called *complex frequency* which is able to describe a growing or decaying or stable sinusoid. By convention we use the variable s to represent this complex frequency ($s = \sigma + j\omega$), with $e^{(\sigma+j\omega)t}$ or simply e^{st} being the time-based function for that growing/decaying sinusoid.

Some time-domain graphs of sinusoids help illustrate the meaning of both σ and ω. Note the inclusion of the coefficient A describing the amplitude of the waveform at the starting time ($t = 0$):



Simple frequency, whether f or ω, merely describes the rate at which an AC signal oscillates. Complex frequency, by contrast, is additionally able to describe growth, decay, or stability. As such it is a more general and indeed more useful way to describe signals.

A practical example of a decaying sinusoid is a passive *tank circuit* gradually losing energy over time after being momentarily excited by an external energy source. A practical example of a growing sinusoid is an *oscillator* circuit starting up from an unpowered condition. The decaying DC graph

illustrates the behavior of a de-energizing inductor or capacitor network as mentioned previously, while a practical example of an exponentially growing DC signal is current through a bipolar junction transistor experiencing *thermal runaway*.

Calculations of impedance also benefit from the use of complex frequency. Instead of simply computing the number of Ohms and the phase angle of impedance for an inductor or capacitor, we may now additionally compute the number of Ohms offered by that component to the exponential growth or decay of that waveform.

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

Again, the complex variable s is used as a convenient way to represent $\sigma + j\omega$, making the previous expressions of impedance extremely simple:

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

ω-based expressions of impedance are now seen as *special cases* of s-based impedances where $\sigma = 0$, as proven by the following algebraic substitutions where we expand s into $\sigma + j\omega$ and then set σ equal to zero. These special cases assume AC signals of constant magnitude that neither grow nor decay over time:

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

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

$$Z_L = (0 + j\omega)L \qquad\qquad Z_C = \frac{1}{(0 + j\omega)C}$$

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

Numerical examples serve well to clarify abstract concepts, so let's apply some complex frequency values to realistic inductor and capacitor values.

First, let's consider a 150 Hz signal applied to a 22 milliHenry inductor. This is a natural frequency of 942.48 radians per second[2]. If this signal is stable in magnitude, its σ value will be zero, and so the impedance offered by this inductor will be:

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

$$Z_L = (0 + j942.48 \text{ s}^{-1})(0.022 \text{ H})$$

$$Z_L = 0 + j20.735\,\Omega = 20.735\,\Omega\angle 90^o$$

Next, let's make this 150 Hz signal *grow* exponentially at a rate of 1000 time constants per second. Re-calculating the inductor's impedance:

$$Z_L = (1000 + j942.48 \text{ s}^{-1})(0.022 \text{ H})$$

$$Z_L = 22 + j20.735\,\Omega = 30.231\,\Omega\angle 43.304^o$$

Our first calculation with a steady 150 Hz signal yielded an impedance of 20.735 Ohms at the customary phase angle of +90 degrees for an inductor. However, our second calculation using the exponentially growing 150 Hz signal gave us an impedance phase angle of only 43.304 degrees. Why is this the case? To answer this question we must recall the distinction between real and imaginary impedance values: real impedance values represent a net exchange of energy into or out of the circuit while imaginary impedance values represent energy alternately absorbed and released in equal measure. In a "steady-state AC" condition the inductor releases precisely as much energy as it earlier absorbed, which makes its impedance purely imaginary. However, when energized by a *growing* AC stimulus each successive cycle of that waveform is stronger than the last, which means the inductor is always absorbing more energy than it released at the close of the previous cycle. This means the inductor exhibits a net absorption of energy over time as the waveform grows stronger and stronger. Unlike a resistor which dissipates energy into heat where it never returns to the circuit, here the inductor's net intake of energy will eventually result in a release of that accumulated energy if and when the waveform decays in value.

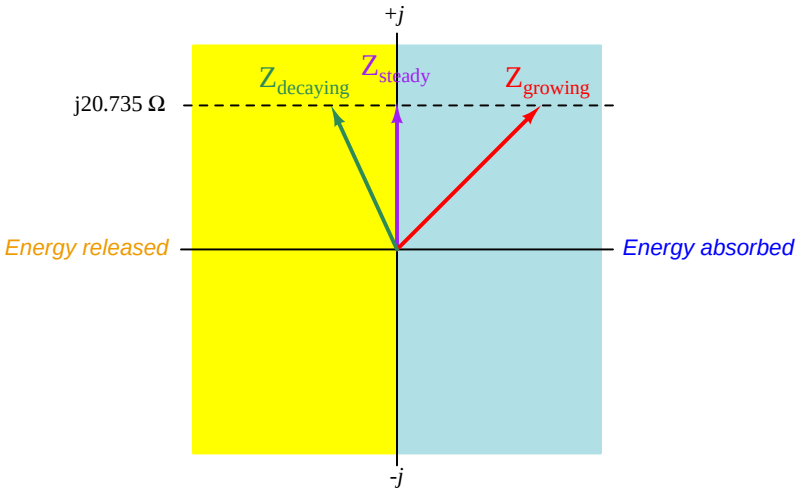
If we were to take this inductor and energize it instead with a 150 Hz signal *decaying* at a rate of -440 time constants per second, we would see its impedance phase angle surpass 90 degrees:

$$Z_L = (-440 + j942.48 \text{ s}^{-1})(0.022 \text{ H})$$

$$Z_L = -9.680 + j20.735\,\Omega = 22.883\,\Omega\angle 115.03^o$$

[2] A common way of expressing "per second" as a unit of measurement is to abbreviate "seconds" using the letter "s" and then apply a -1 power to that unit abbreviation. Therefore, 942.48 radians per second is typically written as 942.48 s^{-1}. Please note that the letter "s" here is a symbol for the unit *seconds*, not to be confused with the variable s which is complex frequency. It is customary to use different fonts to distinguish variables from unit abbreviations. Here, the variable is "s" while the unit is "s".

Plotting these three inductive impedances on a phasor diagram divided into "Energy absorbed" and "Energy released" halves, we see the effect of the signal's growth and decay rate on the inductor's impedance:



In all three cases the imaginary portion of the impedance is the same ($j20.735$ Ohms), but the growth or decay of the signal over time contributes a net absorption or net release, respectively, of energy by the inductor over multiple cycles.

Continuing our quantitative exploration of complex frequency, let us now consider a capacitor subjected to a similar treatment of stable, growing, and decaying signals. For the sake of illustration, let's assume a frequency of 2.3 kHz (14451.3 radians per second) and a capacitance of 0.01 microFarads.

First, the stable signal neither growing nor decaying over time:

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

$$Z_C = \frac{1}{(0 + j14451.3 \text{ s}^{-1})(0.01 \times 10^{-6} \text{ F})}$$

$$Z_C = 0 - j6919.8\,\Omega = 6919.8\,\Omega \angle -90^o$$

Next, calculating capacitive impedance for a signal growing at a rate of 13000 time constants per second at the same frequency of 2.3 kHz:

$$Z_C = \frac{1}{(13000 + j14451.3 \text{ s}^{-1})(0.01 \times 10^{-6} \text{ F})}$$

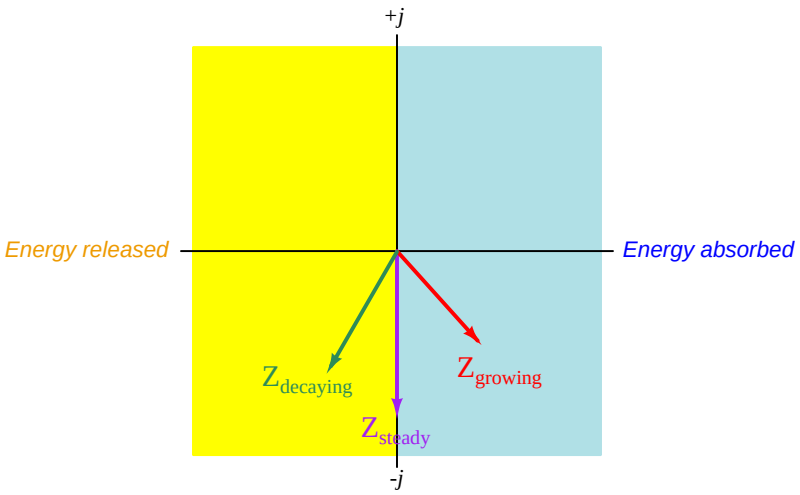
$$Z_C = 3440.6 - j3824.7\,\Omega = 5144.5\,\Omega \angle -48.026^o$$

Lastly, calculating capacitive impedance for a signal decaying at a rate of -9000 time constants per second at the same frequency of 2.3 kHz:

$$Z_C = \frac{1}{(-9000 + j14451.3 \text{ s}^{-1})(0.01 \times 10^{-6} \text{ F})}$$

$$Z_C = -3105.2 - j4986.0\,\Omega = 5873.8\,\Omega \angle -121.91^o$$

Plotting these three capacitive impedances on a phasor diagram divided into "Energy absorbed" and "Energy released" halves:



Once again we see how an AC signal of constant magnitude over time results in an impedance phase angle precisely one-quarter cycle (i.e. -90^o), which means the component absorbs and releases energy in equal measure with each cycle. However, when the AC signal's oscillations grow in strength over time the reactive component absorbs more energy than it releases over multiple cycles; when the oscillations decay the component releases more energy than it absorbs over multiple cycles.

3.4 s variable summary

Here is a summary of some of the major concepts involved with the s variable.

- **Frequency** is a measure of how often something occurs. In basic AC circuit analysis, frequency refers to the occurrence of alternating cycles, usually expressed as *cycles per second* or *Hertz* (Hz) and symbolized by the variable f.

- **Frequency** may also be expressed in units of *radians per second*, there being 2π radians in every cycle. When expressed in radians per second we call an AC frequency *natural frequency* and symbolize it using the lower-case Greek letter "omega" (ω). The conversion between these two types of frequency is $\omega = 2\pi f$.

- **Inductive reactance** typically calculated as $X_L = 2\pi f L$ may be more concisely calculated $X_L = \omega L$. **Capacitive reactance** typically calculated as $X_C = \frac{1}{2\pi f C}$ may be more concisely calculated $X_C = \frac{1}{\omega C}$.

- Electrical signals may also **grow** and **decay** as well as oscillate. When doing so in an exponential fashion over time (e^t or e^{-t}) we symbolize the rate of growth/decay by the variable σ (the lower-case Greek letter "sigma"): $e^{\sigma t}$. Positive σ values represent exponential growth, while negative values represent inverse-exponential decay. A σ value of zero means the signal neither grows nor decays over time.

- Exponential **growth / decay** rate (σ) is also a type of frequency, measured in units of *time constants per second*. This means the growth/decay rate is actually the reciprocal of time constant: $\sigma = \frac{1}{\tau}$, that concept first encountered in RC and LR time-delay circuits. σ is sometimes called the *Neper velocity* of a signal.

- We may combine these two concepts – growth/decay rate and oscillatory frequency – into a single s **variable** comprised of a real part and an imaginary part ($s = \sigma + j\omega$). The real part of s (σ) is the *growth/decay rate* in time constants per second (s^{-1}), while the imaginary part of s ($j\omega$) is the *oscillatory frequency* of the sinusoidal quantity in radians per second (s^{-1}). These parameters are also known as *Neper velocity* and *angular velocity*, respectively. All together, s represents the *complex frequency* of a signal.

- **Complex frequency** is useful for describing a wide range of signals. With $s = \sigma + j\omega$, constant-amplitude AC signal frequency is all imaginary and no real (i.e. $s = 0 + j\omega$). An AC signal growing exponentially over time has positive σ and positive ω values. An AC signal decaying over time has negative σ and positive ω values. A steady DC signal has a complex frequency of zero (i.e. $s = 0 + j0$). A DC signal growing or decaying exponentially over time is all real and no imaginary (i.e. $s = \sigma + j0$), once again the sign of σ determining either growth (+) or decay (−).

- **Inductive impedance** typically calculated as $Z_L = 0 + j(2\pi f L)$ may be more concisely calculated $Z_L = sL$. **Capacitive impedance** typically calculated as $Z_C = 0 - j\left(\frac{1}{2\pi f C}\right)$ may

be more concisely calculated $Z_C = \frac{1}{sC}$. The complex nature of s naturally lends the proper imaginary sign (or polar-form phase angle) to each of these calculated impedances.

3.5 Transfer functions

An extremely important topic in engineering is that of *transfer functions*. Simply defined, a transfer function is the ratio of output to input for any physical system, usually with both the output and input being mathematical functions of s. In other words, we express both the output of a system and the corresponding input of that system in terms of exponentially growing/decaying sinusoidal waves (i.e. *complex frequency*) and then solve for the ratio of those two expressions.

Unfortunately, the teaching of transfer functions and their relation to real-world phenomena is often obfuscated by a heavy emphasis on calculus as well as a particular mathematical analysis known as a *Laplace transform*. The intent of this section is to introduce this concept in a way that is very "gentle" and continually referenced to real-world applications. If I can do anything to help lift the veil of mystery surrounding transfer functions, the s variable, and pole-zero plots, then technicians as well as engineers will be able to appreciate the power of this analytical technique and be better able to exchange ideas in the same "language".

A simple example of a transfer function is the *gain* of an electronic amplifier. As all students of electronics learn, "gain" is the ratio of output signal to input signal for a circuit. Beginning students learn to represent circuit gains as scalar values (e.g. "The amplifier has a voltage gain of 24"), first as plain ratios and later as decibel figures (e.g. "The amplifier has a voltage gain of 27.6 dB"). One limitation to stating gain values as fixed quantities is that real gain values are often frequency-dependent: amplifiers never boost signals by precisely the same ratio at all signal frequencies. If we take the engineering approach of expressing output and input signals as functions of s, however, we obtain a more complete picture of that circuit's behavior over a wide range of conditions including not just different oscillatory frequencies but also different signal growth/decay rates. This is the essence of a *transfer function*: the system's gain as a function of complex frequency (s).

Another simple example of a transfer function is what we have seen previously in this Tutorial: the *impedance* of a reactive electrical component such as a capacitor or an inductor. There, the ratio in question was between voltage and current, as that is the definition of impedance. If we consider current through the component to be the "input" signal and voltage across the component to be the "output" signal – both expressed in terms of s – then impedance $Z(s) = \frac{V(s)}{I(s)}$ is the transfer function for that component. This raises an important point about transfer functions: what we define as the "input" and the "output" of the system is quite arbitrary, so long as there is an actual relationship between the two signals.

If we write generalized output/input transfer functions of s for an AC circuit, we may mathematically analyze that transfer function to gain insight into the behavior and characteristics of that circuit. Interesting features of transfer functions include:

- **Zeros**: any value(s) of s resulting in zero gain for the system in question
- **Poles**: any value(s) of s resulting in infinite gain for the system in question

An AC circuit's *zeros* indicate at which complex-frequency values the circuit is unresponsive to input stimuli. An AC circuit's *poles* tell us which complex frequencies the circuit generates on its own with no input stimulus (i.e. its natural or *un-driven* mode(s) of response). In the context of electrical filter networks, zeroes help define a filter's stop-band while poles help define its pass-band.

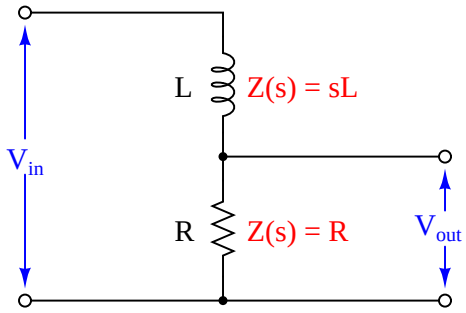
3.6 AC network transfer function examples

In order to clearly understand the concept of transfer functions, practical examples are very helpful. In this section we will explore some elementary AC networks in order to grasp what transfer functions are and how they benefit circuit analysis.

Bear in mind that although we are applying transfer function analysis to networks of resistors, inductors, and capacitors, transfer functions are a *general-purpose* mathematical form useful for describing the behavior of a great many different types of complex systems. Mechanical engineers use transfer functions to identify the vibrational modes of machinery; chemical engineers use transfer functions to model mass and heat transfer within fluid processing systems; control system engineers use transfer functions to predict the stability of closed-loop feedback regulation systems; aerospace engineers use transfer functions to predict the flight stability of aircraft and rockets. One advantage to studying transfer functions in the context of electric circuits is that the systems are generally quite easy and inexpensive to build and test, unlike many of these other system types! Count your blessings that you can safely test circuits on a breadboard for stability and not have to launch rockets to determine whether or not they will fly straight!

3.6.1 Example: LR low-pass filter circuit

First, let's begin with a simple low-pass filter circuit comprised of an inductor and a resistor connected in series:



The impedance of each component as a function of s is shown in the diagram: the inductor's impedance is sL while the resistor's impedance is simply R. It should be clear to any student of electronics that these two components will function as a *voltage divider*, with the output voltage being some fraction of the input voltage. Knowing this, we may write a transfer function for this circuit based on the *voltage divider formula*, which tells us the ratio of output voltage to input voltage is the same as the ratio of output impedance to total impedance:

$$\text{Transfer function} = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R}{R + sL} = \frac{R}{R + (\sigma + j\omega)L}$$

This transfer function allows us to calculate the "gain" of the system for any given value of s, which brings us to the next step of our analysis. At this point we will ask ourselves three questions[3]:

1. How does this system respond to a steady DC condition where $s = 0$?
2. What value(s) of s make the transfer function approach a value of zero?
3. What value(s) of s make the transfer function approach a value of infinity?

The first of these questions refers to a condition where we apply a steady DC signal to the input of the system. Steady-state DC signals neither grow nor decay nor oscillate over time, meaning both σ and ω must each be equal to zero. This necessarily means s must be zero, and when we substitute zero for s in the transfer function we get the following result:

$$\frac{R}{R + 0L}$$

$$\frac{R}{R} = 1$$

[3]What we are really doing here is applying a problem-solving technique called *limiting cases*. This is where we simplify the analysis of some system by considering scenarios where the mathematical quantities are easy to compute.

Therefore the transfer function of this circuit is unity (1) under DC conditions. This is precisely what we would expect of a low-pass filter. If there is no change in the applied signal, then the inductor's magnetic field will be unchanging as well, which means it will drop zero voltage (assuming a pure inductor with no wire resistance) leaving the entire input voltage dropped across the resistor. In other words, this network passes 100% (i.e. has a gain of one) of any DC signal applied to its input terminals.

The second question refers to a condition where the output signal of this circuit is zero. Any values of s resulting in zero output from the system are called the *zeroes* of the transfer function. Examining the transfer function for this particular low-pass LR filter circuit, we see that this can only be true if s becomes infinitely large, because s is located in the denominator of the fraction:

$$\frac{R}{R \pm \infty L} = 0$$

This is also consistent with the expected behavior of a low-pass filter: as frequency (ω) increases, the filter's output signal diminishes. The transfer function doesn't just tell us how this circuit will respond to change in natural frequency, however – it also tells us how the circuit will respond to growing or decaying signals too. Here, we see that infinitely large σ values also result in zero output: the inductor, which tends to oppose any current exhibiting a high rate of change, doesn't allow much voltage to develop across the resistor if the input signal is growing or decaying very rapidly.

The third question refers to a condition where either the transfer function's numerator approaches infinity or its denominator approaches zero. Any values of s having this result are called the *poles* of the transfer function. Since the numerator in this particular case is a constant (R), only a denominator value of zero could cause the transfer function to reach infinity:

$$\frac{R}{R + sL} = \infty \text{ only if } R + sL = 0$$

If the necessary condition for a "pole" is that $R + sL = 0$, then we may solve for s as follows:

$$R + sL = 0$$

$$sL = -R$$

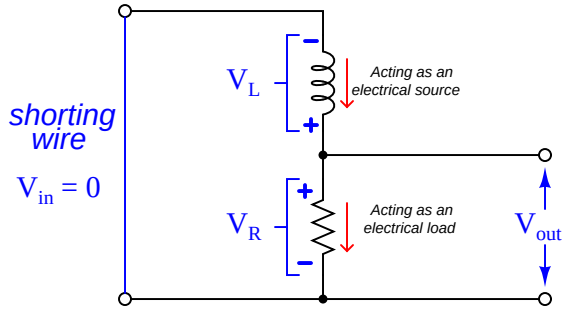
$$s = -\frac{R}{L}$$

Thus, this transfer function for this simple low-pass filter circuit has one pole located at $s = -R/L$. Since both R and L are real numbers (not imaginary) with positive values, then the value of s for the pole must be a real number with a negative value. In other words, the solution for s at this pole is all σ and no ω: this refers to an *exponentially decaying DC signal*:

$$\sigma = -\frac{R}{L} \qquad\qquad j\omega = 0$$

It is important at this point to consider what this "pole" condition means in real life. The notion that a circuit is able to produce an output signal with zero input signal may sound absurd, but it makes sense if the circuit in question has the ability to store and release energy. In this particular circuit, the inductor is the energy-storing component, and it is able to produce a voltage drop across the resistor with zero input voltage in its "discharging" mode.

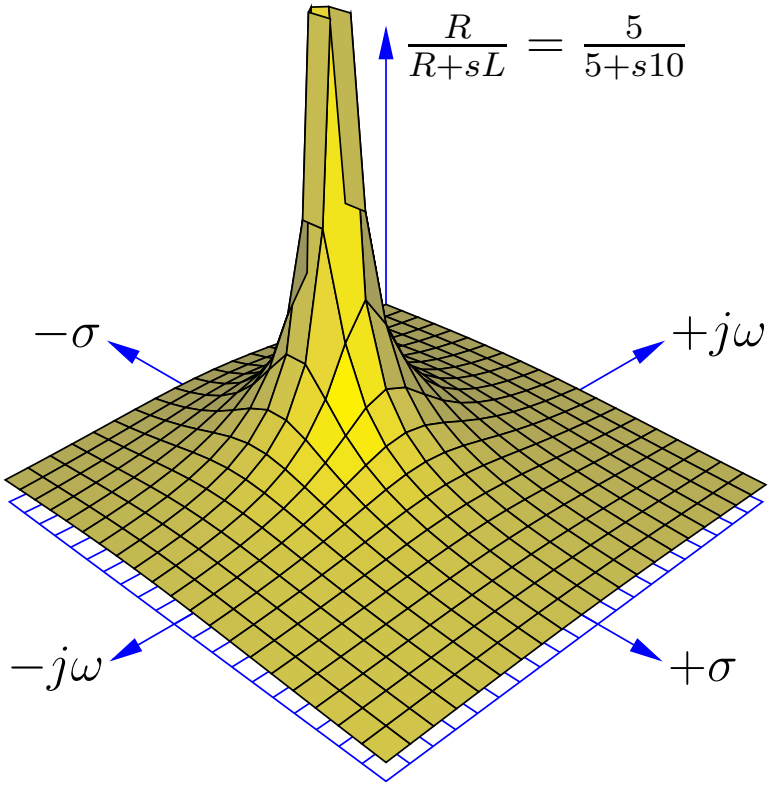
An illustration helps make this clear. If the "pole" condition is such that $V_{in}(s) = 0$, we may show this by short-circuiting the input of our filter circuit to ensure a zero-input condition:



Assuming the inductor has been "charged" with energy previous to the short-circuiting of the input, an output voltage will surely develop across the resistor as the inductor discharges. In other words, the inductor behaves as an electrical *source* while the resistor behaves as an electrical *load*: connected in series they must of course share the same current, but their respective voltages are equal in magnitude and opposing in polarity in accordance with Kirchhoff's Voltage Law. Furthermore, the value of s in this "pole" condition tells us exactly how rapidly the output signal will decay: it will do so at a rate $\sigma = -R/L$. Recall that the growth/decay term of the s variable the reciprocal of the system's time constant ($\sigma = 1/\tau$). Therefore, a σ value of R/L is equivalent to a time constant of L/R, which as all beginning students of electronics learn[4] is how we calculate the time constant for a simple inductor-resistor circuit.

[4]Some liberties are taken for the sake of simplicity when introducing the concept of time constant to new students. For one, the correct definition of τ as being the reciprocal of growth/decay rate σ implies that for rates of *decay*, which are necessarily *negative* σ values, τ should be negative as well. This is mathematically correct, but to avoid extra confusion when introducing the already-confusing notion of time constant to students first learning about capacitors and inductors, time constant (τ) is always assumed to be a positive quantity.

Transfer functions are easier to understand when graphically plotted as three-dimensional surfaces: the real and imaginary portions of the s variable occupying perpendicular axes located on a plane (a flat two-dimensional surface), with the magnitude of the transfer function fraction displayed on a third (vertical) axis perpendicular to the s plane. Here is a *pole-zero plot* of this low-pass filter circuit's transfer function, with a resistor value of $R = 5\,\Omega$ and an inductor value of $L = 10$ H:

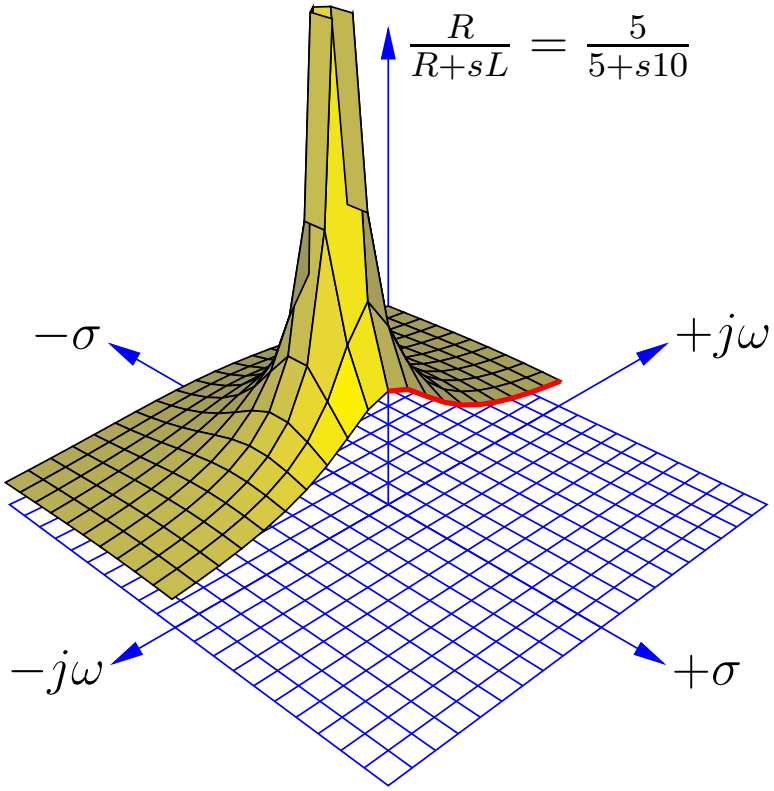


This surface plot makes the meaning of the term "pole" quite obvious: the shape of the function looks just like a rubber mat stretched up at one point by a vertical pole. Here, the "pole" rises to an infinite[5] height at a value of s where $\sigma = -0.5$ time constants per second and $\omega = 0$ radians per second. The surface is seen to decrease in height at all edges of the plot, as σ and ω increase in value.

The "zero" of this transfer function is not as obvious as the pole, since the function's value does not equal zero unless and until s becomes infinite, which of course cannot be plotted on any finite domain. Suffice to say that the zero of this transfer function lies in all horizontal directions at an infinite distance away from the plot's origin (center), explaining why the surface slopes down to zero everywhere with increasing distance from the pole.

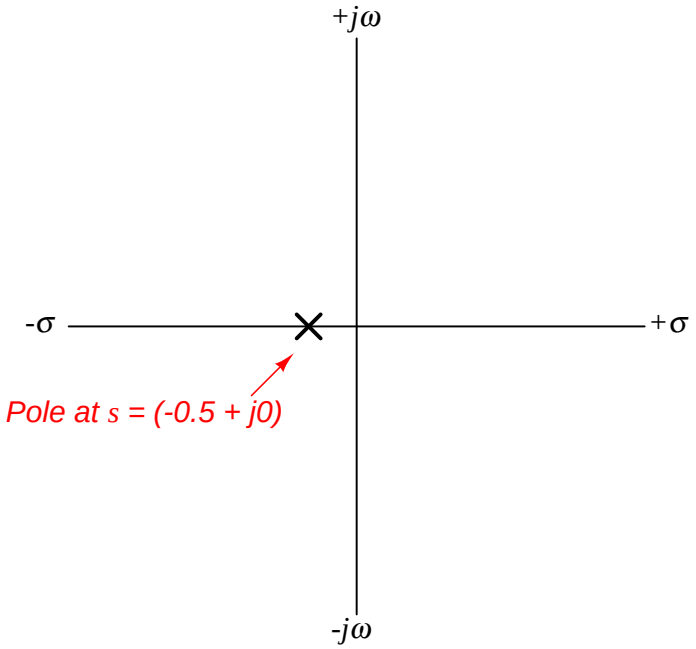
[5] Of course, the mathematical plotting software cannot show a pole of truly infinite height, and so the pole has been truncated. This is why it appears to have a "flat" top.

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, its graphical representation called 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 plane where $\sigma = 0$ (i.e. showing how the system responds to sinusoidal waves of varying frequency that don't grow or decay 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 low-pass filter, starting at a maximum value of 1 ($V_{out} = V_{in}$ for a DC input signal) and approaching zero 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[6] 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 straight down toward the s plane. Since this view hides any features of height, poles and zeros are instead located on the s plane by $\times$ and $\circ$ symbols, respectively. An example of a traditional pole-zero plot for our low-pass filter appears here:



Admittedly, this type of pole-zero plot is much less interesting to look at 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. $\sigma < 0$), that value being 0.5 time constants per second (i.e. a time constant τ of 2 seconds).

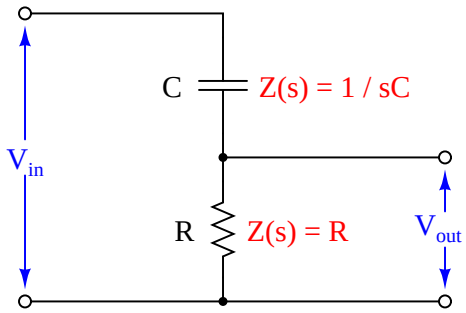
It is worth noting once again that transfer functions and pole-zero plots apply to much more than just electrical filter circuits. In fact, *any* physical system having the same "low-pass" characteristic as this filter circuit is describable by the same transfer function and the same pole-zero plots. Electric circuits just happen to be convenient applications because their individual component characteristics are so easy to represent as functions of s. However, if we are able to characterize the components of a different physical system in the same terms[7], the same mathematical tools apply.

[6] 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.

[7] A powerful mathematical technique known as a *Laplace transform* does this very thing: translate any differential equation describing a physical system into functions of s, which may then be analyzed in terms of transfer functions

3.6.2 Example: RC high-pass filter circuit

For our next exploratory example we will consider another simple filter circuit, this time comprised of a capacitor and a resistor, with the output signal taken across the resistor. As before, we may derive a transfer function by expressing V_{out}/V_{in} as a ratio of the resistor's impedance to the total series resistor-capacitor impedance (treating this as a voltage divider circuit):



$$\text{Transfer function} = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R}{R + \frac{1}{sC}} = \frac{R}{R + \frac{1}{(\sigma + j\omega)C}}$$

After writing this initial transfer function based on component impedances, we will algebraically manipulate it to eliminate compound fractions. This will aid our analysis of the circuit's DC response, zeros, and poles:

$$\frac{R}{R + \frac{1}{sC}}$$

$$\frac{R}{\frac{sRC}{sC} + \frac{1}{sC}}$$

$$\frac{R}{\frac{1+sRC}{sC}}$$

$$\frac{sRC}{1 + sRC}$$

This transfer function allows us to calculate the "gain" of the system for any given value of s, which brings us to the next step of our analysis. Once again we will ask ourselves three questions about the transfer function:

1. How does this system respond to a steady DC condition where $s = 0$?
2. What value(s) of s make the transfer function approach a value of zero?
3. What value(s) of s make the transfer function approach a value of infinity?

---

and pole-zero plots.

In answer to the first question – applying a steady DC signal to the input of this network – we see that the transfer function is equal to zero when $s = 0$:

$$\frac{0RC}{1 + 0RC}$$

$$\frac{0}{1 + 0} = \frac{0}{1} = 0$$

Of course, a value of 0 for s means exposure to a steady DC signal: one that neither grows nor decays over time, nor oscillates. Therefore, this resistor-capacitor circuit will output zero voltage when exposed to a purely DC input signal. This makes conceptual sense when we examine the circuit itself: a DC input signal voltage means the capacitor will not experience any change in voltage over time, which means it will not pass any current along to the resistor. With no current through the resistor, there will be no output voltage. Thus, the capacitor "blocks" the DC input signal, preventing it from reaching the output. This behavior is exactly what we would expect from such a circuit, which any student of electronics should immediately recognize as being a simple *high-pass* filter: DC is a condition of zero frequency, which should be completely blocked by any filter circuit with a high-pass characteristic.

The answer to our first question is also the answer to the second question: "what value of s makes the transfer function equal to zero?" Here we see that it is only at a value of $s = 0$ that the entire transfer function's value will be zero. Any other values for s – even infinite – yield non-zero results. In contrast to the last circuit (the resistor-inductor low-pass filter) this circuit exhibits a singular "zero" point in its transfer function: one specific location on the pole-zero plot where the function's value diminishes to nothing.

When we consider the third question ("What value(s) of s make the transfer function approach a value of infinity?") we proceed the same as before: by finding value(s) of s which will make the denominator of the transfer function fraction equal to zero. If we set the denominator portion equal to zero and solve for s, we will obtain the *pole* for the circuit:

$$1 + sRC = 0$$

$$sRC = -1$$

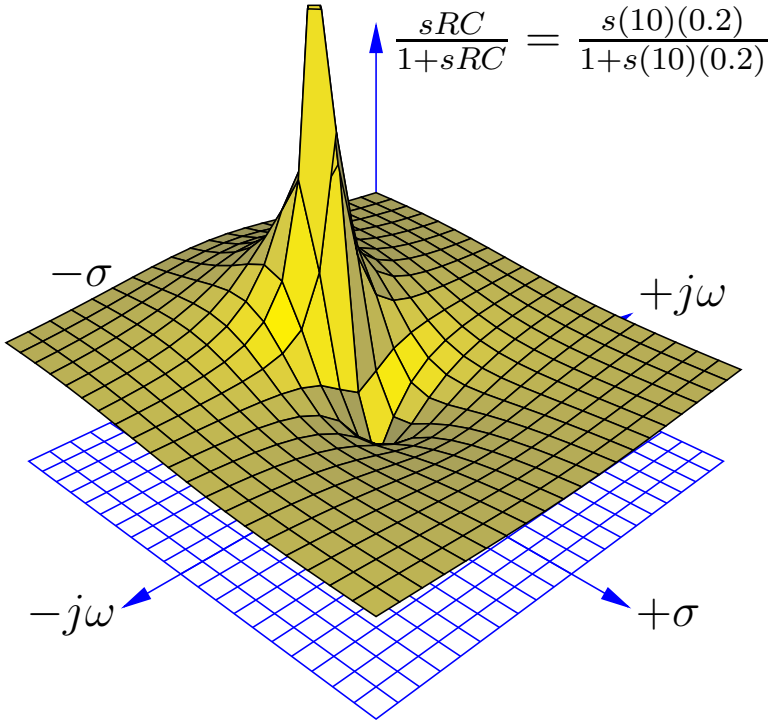
$$s = -\frac{1}{RC}$$

We know that both R and C are real numbers, not imaginary. This tells us that s will likewise be a real number at the pole. That is to say, s will be comprised of all σ and no ω. The fact that the value of σ is negative tells us the pole represents a condition of *exponential decay*, just the same as in the case of the resistor-inductor low-pass filter. As before, this means the circuit will produce an output voltage signal with no[8] input voltage signal when the rate of signal decay is $\sigma = -1/RC$.

[8] As before, this counter-intuitive condition is possible only because the capacitor in this circuit has the ability to store energy. If the capacitor is charged by some previous input signal event and then allowed to discharge through the resistor, it becomes possible for this circuit to develop an output voltage even with short-circuited input terminals.

Recall that the rate of decay in the s variable (σ) is nothing more than the reciprocal of the system's time constant (τ). Thus, a rate of decay equal to $1/RC$ equates to a time constant $\tau = RC$, which as all electronics students know[9] is how we calculate the time constant for any simple resistor-capacitor circuit.

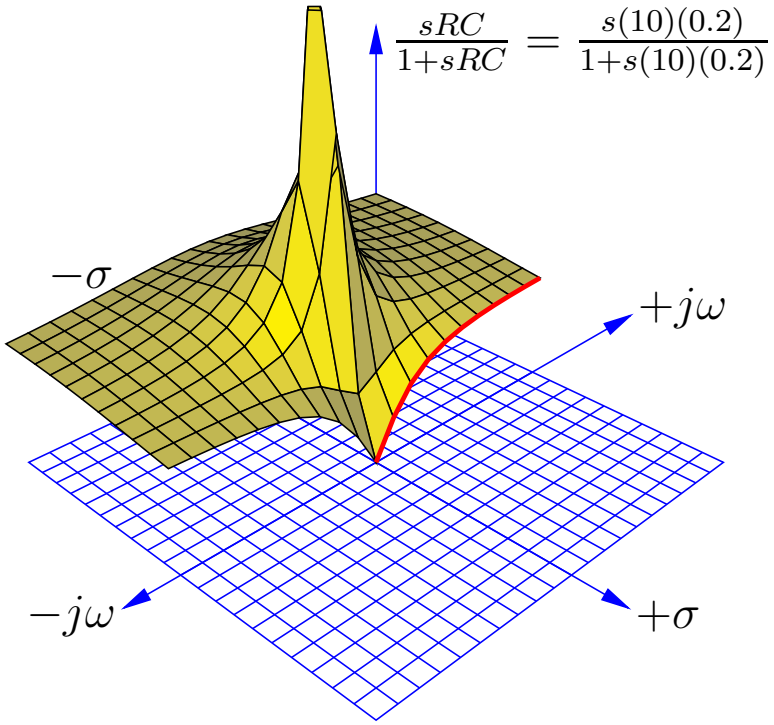
Using a computer to plot a three-dimensional representation of this transfer function, we clearly see both the pole and the zero as singularities. Here I have assumed a 10 Ω resistor and a 0.2 F capacitor to place the pole at the same location as with the low-pass filter circuit $s = -0.5 + j0$, for an equitable comparison:



Here we see both the pole at $s = -0.5 + j0$ and the zero at $s = 0 + j0$ quite clearly: the pole is a singular point of infinite height while the zero is a singular point of zero height. The three-dimensional surface of the transfer function looks like a rubber sheet that has been stretched to an infinite height at the pole and stretched to ground level at the zero.

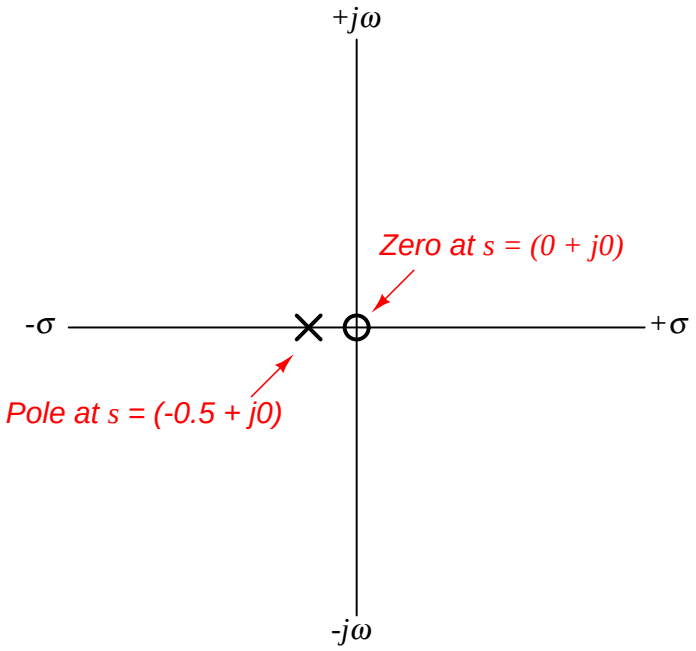
[9] Some liberties are taken for the sake of simplicity when introducing the concept of time constant to new students. For one, the correct definition of τ as being the reciprocal of growth/decay rate σ implies that for rates of *decay*, which are necessarily *negative* σ values, τ should be negative as well. This is mathematically correct, but to avoid extra confusion when introducing the already-confusing notion of time constant to students first learning about capacitors and inductors, time constant (τ) is always assumed to be a positive quantity.

As in the last example, we may re-plot the transfer function in a way that shows a cross-sectional view at $\sigma = 0$ in order to reveal the frequency response of this high-pass filter circuit:



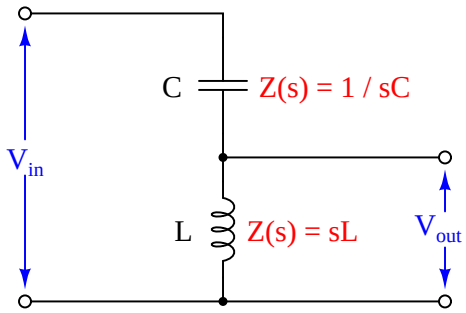
Once again the bold, red curve traces the edge of the transfer function surface as it begins at zero frequency (DC) to increasingly positive values of $j\omega$. The red trace is therefore the Bode plot for this high-pass filter, starting at a minimum value of 0 ($V_{out} = 0$ for a DC input signal) and approaching unity (1) as frequency increases. Naturally, this is the type of response we would expect to see exhibited by a high-pass filter circuit.

A more traditional two-dimensional pole-zero plot for this circuit locates the pole with a "×" symbol and the zero with a "○" symbol:



3.6.3 Example: LC "tank" circuit

Next, we will explore the transfer function for a *tank circuit*, comprised of a capacitor and an inductor. We will assume the use of pure reactances here with no electrical resistance or other energy losses of any kind, just to analyze an ideal case. The output voltage in this particular circuit will be taken across the inductor:



Writing the transfer function for this tank circuit is (once again) a matter of expressing the ratio between the output component's impedance versus the total circuit impedance:

$$\text{Transfer function} = \frac{V_{out}(s)}{V_{in}(s)} = \frac{sL}{sL + \frac{1}{sC}}$$

Algebraically manipulating this function to eliminate compound fractions:

$$\frac{sL}{sL + \frac{1}{sC}}$$

$$\frac{sL}{\frac{sLsC}{sC} + \frac{1}{sC}}$$

$$\frac{sL}{\frac{s^2LC}{sC} + \frac{1}{sC}}$$

$$\frac{sL}{\frac{s^2LC+1}{sC}}$$

$$\frac{sLsC}{s^2LC + 1}$$

$$\frac{s^2LC}{s^2LC + 1}$$

Note how this transfer function contains s^2 terms rather than s terms. This makes it a *second-order function*, which will yield very different results on the pole-zero plot than what we saw with either the resistor-inductor or resistor-capacitor filter circuits.

Asking ourselves the same three questions again:

1. How does this system respond to a steady DC condition where $s = 0$?
2. What value(s) of s make the transfer function approach a value of zero?
3. What value(s) of s make the transfer function approach a value of infinity?

In answer to the first question (DC voltage for the input signal) we see that the transfer function is equal to zero when $s = 0$:

$$\frac{s^2LC}{s^2LC + 1}$$

$$\frac{0}{0 + 1} = \frac{0}{1} = 0$$

As with the RC low-pass filter, its response at DC also happens to be a "zero" for the transfer function. With a DC input signal, the output signal of this circuit will be zero Volts.

In order to find poles for this transfer function, we must solve for values of s that will make the denominator term of the transfer function equal to zero:

$$s^2LC + 1 = 0$$

$$s^2LC = -1$$

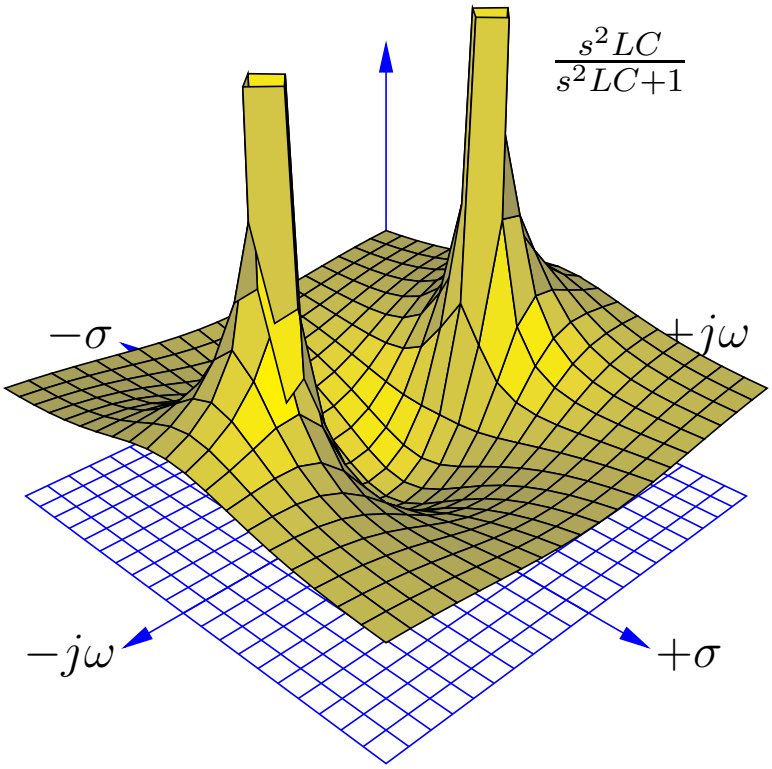
$$s^2 = -\frac{1}{LC}$$

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

$$s = \pm j\sqrt{\frac{1}{LC}}$$

Given the fact that both L and C are real, positive numbers, and therefore solving for s requires we take the square root of a negative real number, we see that the value of s must be imaginary. We also see here that there are *two* poles in this transfer function: one at $s = 0 + j\sqrt{\frac{1}{LC}}$ and another at $s = 0 - j\sqrt{\frac{1}{LC}}$.

Using a computer to plot a three-dimensional representation of this transfer function, we clearly see a single zero at $s = 0$ and two poles symmetrically positioned along the $j\omega$ axis. Here I have assumed a 0.2 F capacitor and a 5 H inductor for component values:



The two[10] poles located on the $j\omega$ axis (one at $s = 0 + j1$ and the other at $s = 0 - j1$) tell us the circuit is able to generate an oscillating output signal ($\omega = 1$ radian per second frequency) at constant magnitude ($\sigma = 0$) with no input signal. This is only possible because we have assumed a perfect capacitor and a perfect inductor with no energy losses whatsoever. If we charge up either or both of these components and then immediately short-circuit the input of the circuit to ensure $V_{in} = 0$, it will oscillate at its resonant frequency forever.

[10] The *two* solutions for ω (one at +1 radian per second and the other at −1 radian per second) merely indicate the circuit is able to oscillate "forward" as well as "backward". In other words, it is able to oscillate sinusoidally where the positive peak occurs at time $t = 0$ (+1 rad/s) as well as oscillate sinusoidally where the negative peak occurs at time $t = 0$ (−1 rad/s). We will find that solutions for s in general are symmetrical about the real axis, meaning if there is any solution for s requiring an imaginary number value, there will be *two* of them: one with a *positive* imaginary value and the other with a *negative* imaginary value.

Earlier we noted that the poles in this circuit were $s = 0 + j\sqrt{\frac{1}{LC}}$ and $s = 0 - j\sqrt{\frac{1}{LC}}$. In other words, its resonant frequency is $\omega = \sqrt{\frac{1}{LC}}$. Recalling that the definition for ω is radians of rotation per second, and that there are 2π radians in one complete revolution (cycle), we can derive the familiar resonant frequency formula for a simple LC circuit:

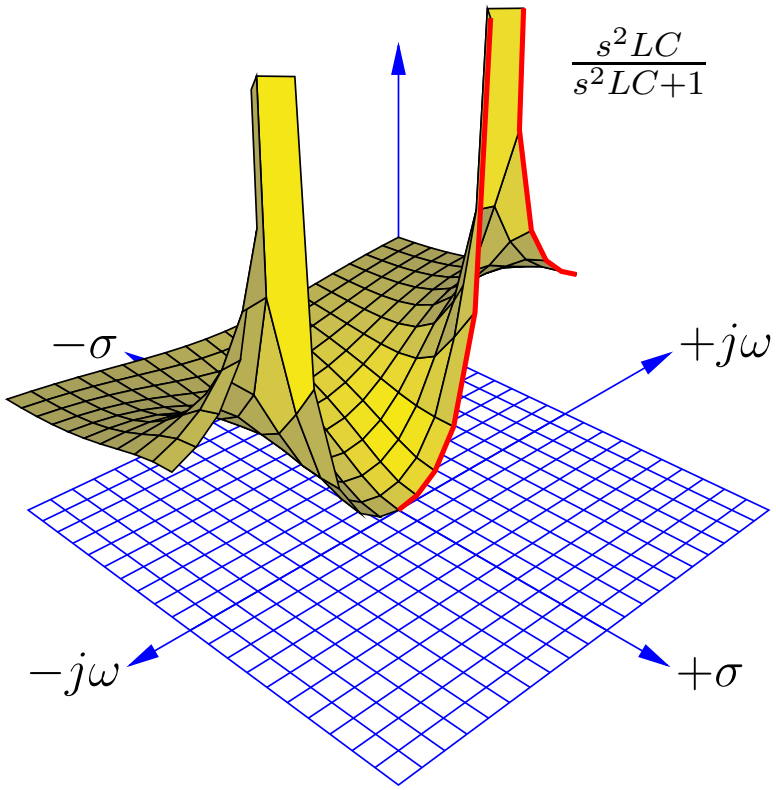
$$\omega = \sqrt{\frac{1}{LC}}$$

. . . substituting $2\pi f$ for ω . . .

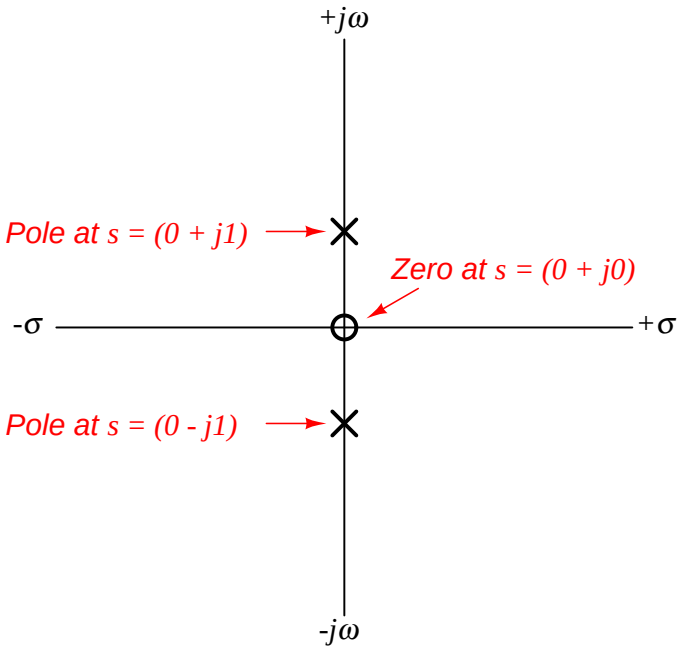
$$2\pi f = \sqrt{\frac{1}{LC}}$$

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

Taking a cross-section of this surface plot at $\sigma = 0$ to obtain a frequency response (Bode plot) of the LC tank circuit, we see the output of this circuit begin at zero when the frequency (ω) is zero, then the output peaks at the resonant frequency ($\omega = 1$ rad/s), then the output approaches unity (1) as frequency increases past resonance:



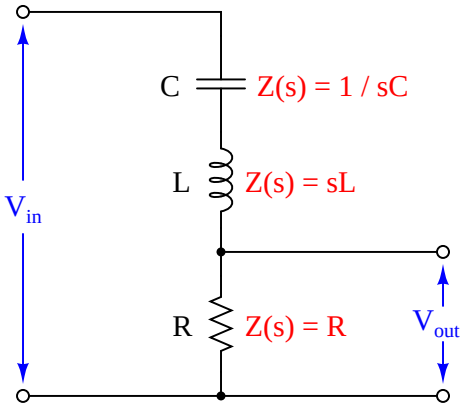
A more traditional two-dimensional pole-zero plot for this circuit locates the zero and the two poles using "$\circ$" and "$\times$" symbols:



The placement of these poles on the imaginary axis means their real values are zero, which in turn means the oscillations neither grow nor decay over time. This, of course, is possible only for an LC network with no parasitic losses (e.g. resistance) whatsoever. If we were to include dissipative elements into the network we would see a decay rate as well, shifting the poles to the left so as to have negative real ($-\sigma$) values.

3.6.4 Example: RLC band-pass filter circuit

For our next example circuit, we will add a resistor in series with the inductor and capacitor to explore its effects on the transfer function. Taking our output voltage across the resistor, we should expect to see *band-pass* filtering behavior from this circuit, with maximum voltage developing across the resistor at one frequency where the inductor's and capacitor's impedances cancel:



As usual, the transfer function for this circuit is the ratio between the output component's impedance (R) and the total series impedance, functioning as a voltage divider:

$$\text{Transfer function} = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R}{sL + \frac{1}{sC} + R}$$

Algebraically manipulating this function to eliminate compound fractions:

$$\frac{R}{sL + \frac{1}{sC} + R}$$

$$\frac{R}{\frac{s^2LC}{sC} + \frac{1}{sC} + \frac{sRC}{sC}}$$

$$\frac{R}{\frac{s^2LC+sRC+1}{sC}}$$

$$\frac{sRC}{s^2LC + sRC + 1}$$

As with the pure tank circuit analyzed previously, we can see that this LC circuit exhibits a *second-order* transfer function because it contains an s^2 term. Running through our three questions again:

1. How does this system respond to a steady DC condition where $s = 0$?
2. What value(s) of s make the transfer function approach a value of zero?
3. What value(s) of s make the transfer function approach a value of infinity?

The answers to the first two questions are one and the same: the numerator of the transfer function will be zero when $s = 0$, this being the single *zero* of the function. Recalling that a condition of $s = 0$ represents a DC input signal (no growth or decay over time, and no oscillation), this makes perfect sense: the presence of the DC-blocking series capacitor in this circuit ensures the output voltage under steady-state conditions must be zero.

In answering the third question to identify any poles for this circuit, we encounter a more complicated mathematical problem than seen with previous example circuits. The denominator of the transfer function's fraction is a *second-degree polynomial function* of the variable s. As you may recall from your study of algebra, any solution resulting in a polynomial having an over-all value of zero is called a *root* of that polynomial expression. Since this particular expression is found in the denominator of the transfer function where we know zero values mark poles of the system, and solutions for s resulting are roots of the polynomial, then roots of the expression $s^2LC + sRC + 1$ must mark the locations of the poles on the s plane.

A very useful algebraic tool for finding roots of a second-degree polynomial expression is the *quadratic formula*:

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

Where,

$ax^2 + bx + c$ is a polynomial expression

x is the independent variable of that polynomial expression

a is the coefficient of the second-degree (x^2) term – *in this case, LC*

b is the coefficient of the first-degree (x) term – *in this case, RC*

c is the coefficient of the zero-degree (constant) term – *in this case,* 1

Reviewing the denominator of our transfer function again, we see that s is the independent variable, and so LC must be the "a" coefficient, RC must be the "b" coefficient, and 1 must be the "c" coefficient. Substituting these variables into the quadratic formula will give us a formula for computing the poles of this resistor-inductor-capacitor circuit:

$$s = \frac{-RC \pm \sqrt{(RC)^2 - 4LC}}{2LC}$$

Perhaps the most interesting part of this formula is what lies beneath the radicand (square-root) symbol: $(RC)^2 - 4LC$. This portion of the quadratic formula is called the *discriminant*, and its value determines both the number of roots as well as their real or imaginary[11] character. If the discriminant is equal to zero, there will be a single real root for our polynomial and therefore only one pole for our circuit. If the discriminant is greater than zero (i.e. a positive value), then there will be two real roots and therefore two poles lying on the σ axis (i.e. no imaginary $j\omega$ parts). If the discriminant is less than zero (i.e. a negative value), then there will be two complex roots for our polynomial and therefore two complex poles having both real and imaginary parts.

Let us consider for a moment what the sign of the discriminant means in practical terms. A pole that is purely real means a value for s that is all σ and no ω: representing a condition of growth

[11] The only way to obtain a purely imaginary root for this polynomial is for the "b" coefficient to be equal to zero. For our example circuit, it means either R or C would have to be zero, which is impossible if both of those components are present and functioning. Thus, our RLC filter circuit will have either *real* poles or *complex* poles.

or decay but with no oscillation. This is similar to what we saw with the LR or RC low/high pass filter circuits, where the circuit in a state of discharge could generate an output signal even with its input terminals shorted to ensure no input signal.

If we have a positive value for the discriminant which yields *two* real poles, it means two different possible values for σ (rate of growth/decay) that could occur with no signal input to the circuit. This behavior is only possible with two energy-storing components in the circuit: a kind of *double time-constant* where different portions of the circuit discharge at different rates. The lack of any imaginary part within s means the circuit still will not self-oscillate.

If we have a negative value for the discriminant which yields *two* complex poles, it means two different values for s both having real and imaginary parts. Since the real part (σ) represents growth/decay while the imaginary part (ω) represents oscillation, complex poles tell us the circuit will be able to self-oscillate but not at a constant magnitude as with an ideal (lossless) tank circuit. In fact, intuition should tell us these complex poles must have negative real values representing decaying oscillations, because it would violate the Law of Energy Conservation for our circuit to self-oscillate with increasing magnitude.

Looking at the discriminant $(RC)^2 - 4LC$ we see that it is possible to push the circuit into any one of these three modes of operation merely by adjusting the value of R and leaving both L and C unchanged. If we wish to calculate the critical value of R necessary to produce a single real pole for any given values of L and C, we may set the discriminant equal to zero and algebraically solve for R as follows:

$$(RC)^2 - 4LC = 0$$

$$(RC)^2 = 4LC$$

$$RC = \sqrt{4LC}$$

$$RC = 2\sqrt{L}\sqrt{C}$$

$$R = \frac{2\sqrt{L}\sqrt{C}}{C}$$

$$R = \frac{2\sqrt{L}\sqrt{C}}{\sqrt{C}\sqrt{C}}$$

$$R = \frac{2\sqrt{L}}{\sqrt{C}} = 2\sqrt{\frac{L}{C}}$$

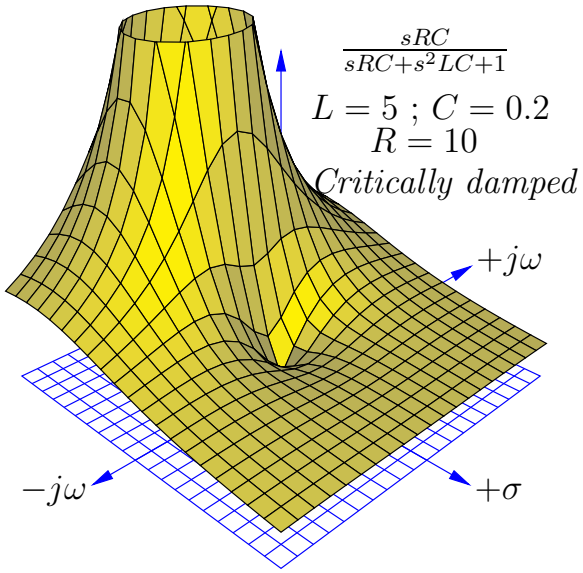
This critical value of R resulting in one real pole is the minimum amount of resistance necessary to prevent self-oscillation. If the circuit is operating at this point, it is said to be *critically damped*. Larger values of R will result in multiple real poles, where the circuit is said to be *over-damped*. Smaller values of R will permit some self-oscillation to occur, and the circuit is said to be *under-damped*.

Sometimes electrical engineers intentionally install resistors into circuits containing both inductance and capacitance for the express purpose of damping oscillations. In such cases, the resistor is called an *anti-resonance* resistor, because its purpose is to combat resonant oscillations that would otherwise occur as the inductive and capacitive elements of the circuit exchange energy back and forth with each other. If the engineer's intent is to install just enough resistance into the circuit to prevent oscillations without creating unnecessary time delays, then the best value of the resistor will be that which causes critical damping.

Recall that the subject of transfer functions, poles, and zeros applies to *any* system that is both *linear* (i.e. its output and input magnitudes will be proportional to each other for all conditions) and *time-invariant* (i.e. the essential characteristics of the system do not change with the passage of time), not just AC circuits. Mechanical systems, feedback control loops, and many other physical systems may be characterized in the same way using the same mathematical tools. This particular subject of damping is extremely important in applications where oscillations are detrimental. Consider the design of an automobile's suspension system, where the complementary energy-storing phenomena of spring tension and vehicle mass give rise to oscillations following impact with a disturbance in the road surface. It is the job of the shock absorber to act as the "resistor" in this system and dissipate energy in order to minimize oscillations following a bump in the road. An under-sized shock absorber won't do a good enough job dissipating the energy of the disturbance, and so the vehicle's suspension will exhibit complex poles (i.e. there will be some lingering oscillations following a bump). An over-sized shock absorber will be too "stiff" and allow too much of the bump's energy to pass through to the vehicle frame and passengers. A perfectly-sized shock absorber, however, will "critically damp" the system to completely prevent oscillation while presenting the smoothest ride possible.

In order to fully illustrate the characteristics of this circuit's transfer function, we will do so for three different resistor values: one where R yields critical damping (one real pole), one where R makes the circuit over-damped (two real poles), and one where R makes the circuit under-damped (two complex poles). We will use three-dimensional plotting to show the transfer function response in each case. To be consistent with our former tank circuit example, we will assume the same capacitor value of 0.2 Farads and the same inductor value of 5 Henrys. The resistor value will be modified in each case to create a different damping condition.

First, the critically-damped example, with a resistor value of 10 Ohms:



As expected, a single zero appears at $s = 0$, and a single[12] pole at $s = -1 + j0$. Thus, this circuit has a decay rate of -1 time constants per second ($\tau = 1$ second) as seen when we use the quadratic formula to solve for s:

$$s = \frac{-RC \pm \sqrt{(RC)^2 - 4LC}}{2LC}$$

$$s = \frac{-(10)(0.2) \pm \sqrt{[(10)(0.2)]^2 - (4)(5)(0.2)}}{(2)(5)(0.2)}$$

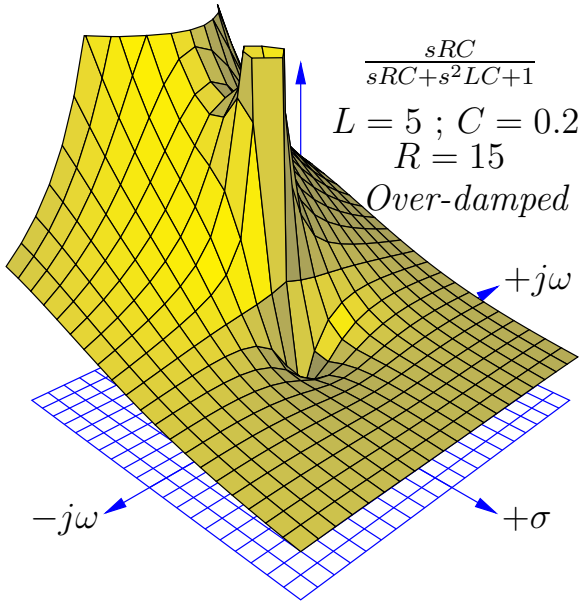
$$s = \frac{-2 \pm \sqrt{0}}{2} = (-1 + j0)\ \text{s}^{-1}$$

Interestingly, only L and R determine the decay rate (σ, the real part of s) in the critically damped condition. This is clear to see if we set the discriminant to zero in the quadratic formula and look for variables to cancel:

$$s = \frac{-RC \pm \sqrt{0}}{2LC} = -\frac{R}{2L}$$

[12] Or, one might argue there are two *superimposed* poles, one at $s = -1 + j0$ and another at $s = -1 - j0$.

Next, we will plot the same transfer function with a larger resistor value (15 Ohms) to ensure over-damping:



We clearly see *two* poles[13] centered along the σ axis in this plot, representing the two real roots of the transfer function's denominator. Again, we will use the quadratic formula to solve for these two values of s:

$$s = \frac{-RC \pm \sqrt{(RC)^2 - 4LC}}{2LC}$$

$$s = \frac{-(15)(0.2) \pm \sqrt{[(15)(0.2)]^2 - (4)(5)(0.2)}}{(2)(5)(0.2)}$$

$$s = \frac{-3 \pm \sqrt{(3)^2 - 4}}{2}$$

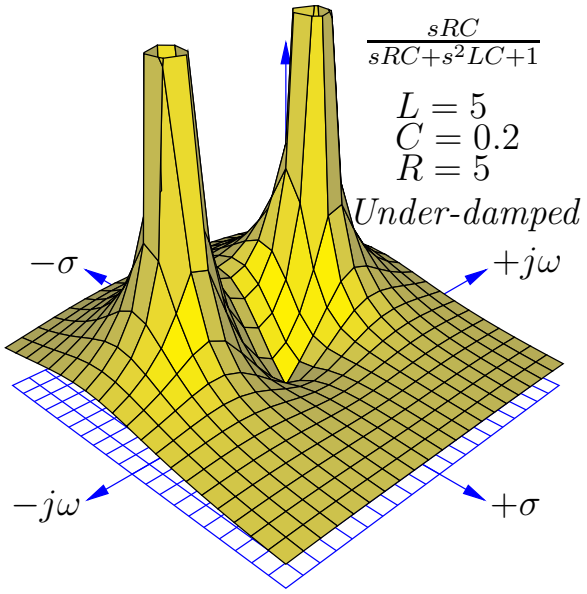
$$s = \frac{-3 + \sqrt{5}}{2} = (-0.382 + j0)\ \text{s}^{-1}$$

$$s = \frac{-3 - \sqrt{5}}{2} = (-2.618 + j0)\ \text{s}^{-1}$$

These real poles represent two different decay rates (time constants) for the over-damped circuit: a fast decay rate of $\sigma = -2.618\ \text{s}^{-1}$ and a slow decay rate of $\sigma = -0.382\ \text{s}^{-1}$, the slower of these two decay rates dominating the circuit's transient response over long periods of time.

[13] The center of the pole farthest from the plot's origin actually lies outside the plotted area, which is why that pole appears to be vertically sliced. This plot's domain was limited to the same values (± 2) as previous plots for the sake of visual continuity, the compromise here being an incomplete mapping of one pole.

Next, we will plot the same transfer function with a smaller resistor value (5 Ohms) to ensure under-damping:



We clearly see *two* poles once again, but neither of them are located on an axis. These represent two *complex* values for s describing the circuit's behavior with zero input. The non-zero imaginary ($j\omega$) part of s tells us the circuit has the ability to self-oscillate. The negative, real (σ) part of s tells us these oscillations decrease in magnitude over time. Using the quadratic formula to solve for these two poles:

$$s = \frac{-RC \pm \sqrt{(RC)^2 - 4LC}}{2LC}$$

$$s = \frac{-(5)(0.2) \pm \sqrt{[(5)(0.2)]^2 - (4)(5)(0.2)}}{(2)(5)(0.2)}$$

$$s = \frac{-1 \pm \sqrt{(1)^2 - 4}}{2}$$

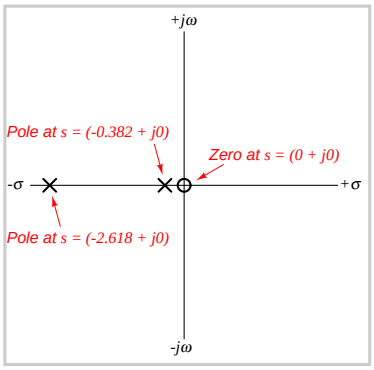
$$s = \frac{-1 + \sqrt{-3}}{2} = (-0.5 + j0.866) \text{ s}^{-1}$$

$$s = \frac{-1 - \sqrt{-3}}{2} = (-0.5 - j0.866) \text{ s}^{-1}$$

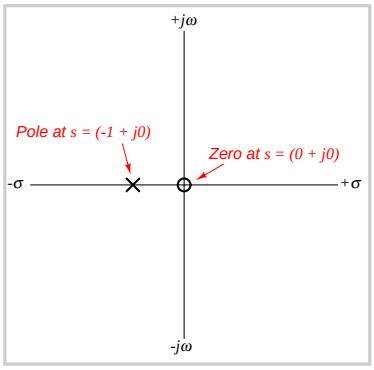
The calculated ω value of 0.866 radians per second is slower than the 1 radian per second resonant frequency calculated for the pure tank circuit having the same L and C values, revealing that the damping resistor skews the "center" frequency of this RLC band-pass filter. The calculated σ value of -0.5 time constants per second (equivalent to a time constant of $\tau = 2$ seconds) describes the

rate at which the sinusoidal oscillations decay in magnitude. Here as well we see the under-damped decay rate ($\sigma = -0.5$ s^{-1}) is slower than the critically damped decay rate ($\sigma = -1$ s^{-1}).

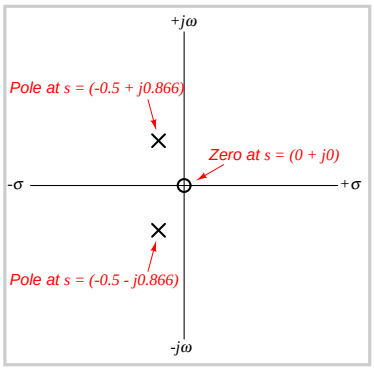
If we compare two-dimensional pole-zero plots for each of the three resistor values in this RLC circuit, we may contrast the over-damped, critically-damped, and under-damped responses:



Over-damped (R = 15 Ω)

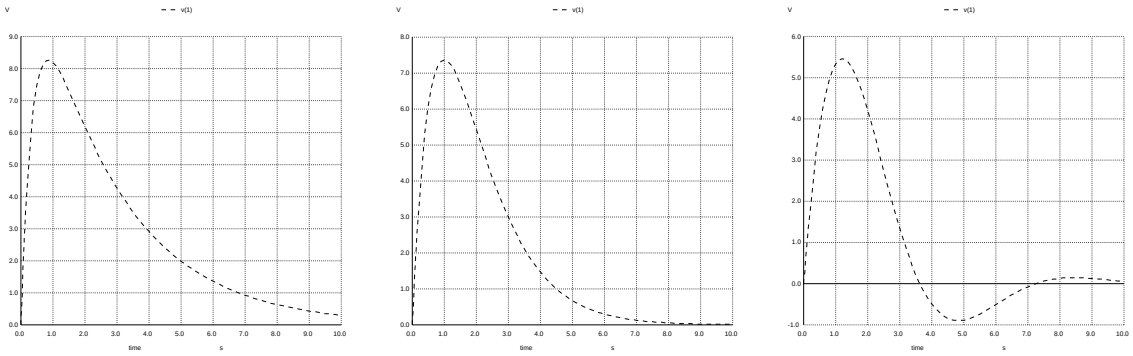


Critically damped (R = 10 Ω)



Under-damped (R = 5 Ω)

Time-domain plots of these networks' responses clearly show the consequences of over-damping (left) with the 15-Ohm resistor, critical damping (center) with the 10-Ohm resistor, and under-damping (right) with the 5-Ohm resistor:



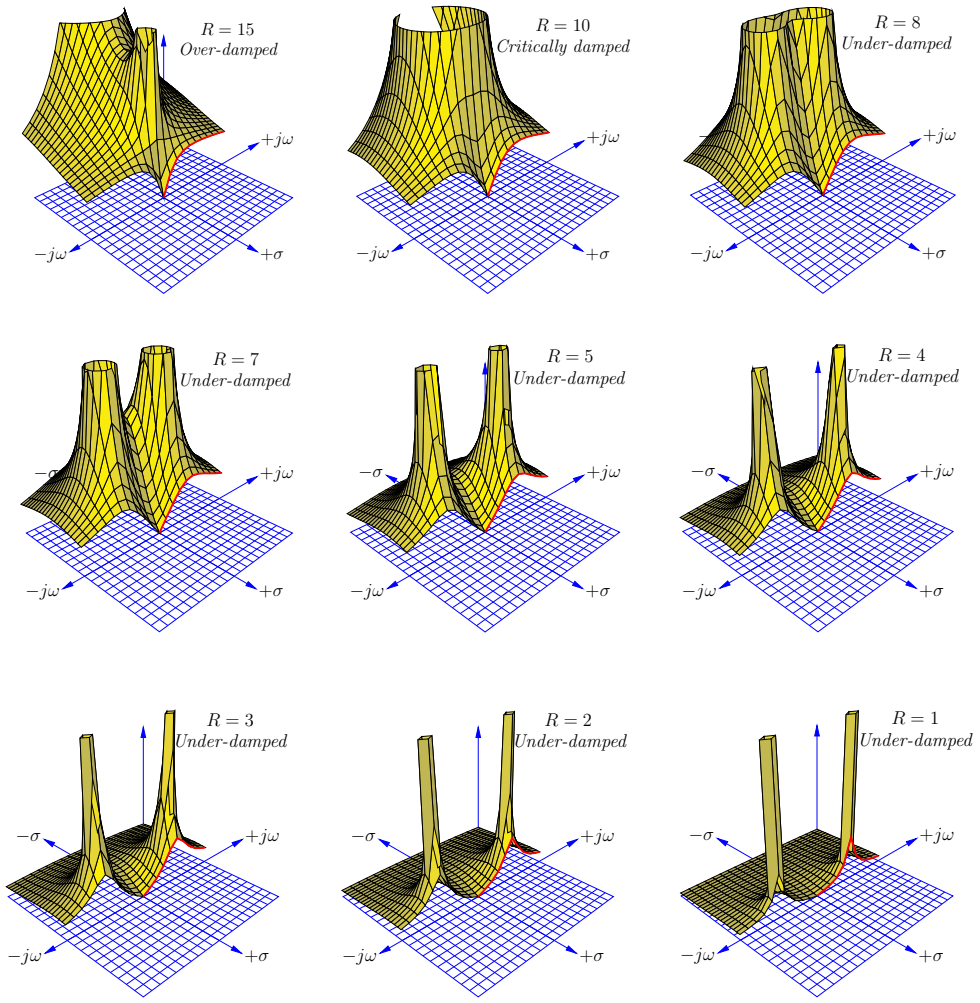
Note how the under-damped case (right) is the only one where the output signal actually oscillates about the 0-Volt axis. In the critically damped case (middle) the output decays rapidly to zero, while in the over-damped case (left) the output's decay to zero is more gradual.

To generate these plots, I created a SPICE netlist of the RLC network and then performed a transient analysis assuming the capacitor begins in a "charged" state with an initial condition of 10 Volts (`ic=10`) and that the inductor begins in a completely de-energized state of 0 Amperes (`ic=0`). The SPICE netlist for the critically damped analysis is shown below:

```
* Critically damped RLC network
r1 1 0 5
c1 1 2 0.2 ic=10
l1 2 0 5 ic=0
.tran 100m 10 uic
.plot tran v(1)
.end
```

Recall the automobile suspension system example discussed earlier, where the task of the "shock absorber" components was to dissipate mechanical energy applied to the vehicle in the form of the "shock" experienced after striking a bump in the road. Too much shock-absorber dissipation leads to over-damping where too much of the bump's energy transfers to the chassis and where the suspension system takes too long to settle to a neutral condition following that bump. Too little shock-absorption causes under-damping where the vehicle bounces up and down after striking a bump. Critical damping is an ideal response to such a transient.

If we take cross-sectional plots of the transfer function at $\sigma = 0$ to show the frequency response of this RLC band-pass filter, we see the response become "sharper" (more selective) as the resistor's value decreases and the poles move closer to the $j\omega$ axis. This is related to the *quality factor* or Q of the band-pass filter circuit, the circuit exhibiting a higher "quality" of band-pass selection as the ratio of reactance to resistance increases:



Although each and every pole in a pole-zero plot is has the same (infinite) height, poles grow narrower when moved farther away from each other, and wider when closely spaced. As the resistance in this circuit decreases and the poles move farther away from each other and closer to the $j\omega$ axis, their widths narrow and the frequency response curve's peak becomes narrower and steeper. This narrowing of the pole-peaks represents the band-pass filter having a higher *quality factor* (Q) value, making it more selective to its center-frequency.

3.7 Transfer function summary

Here is a summary of some of the major concepts important to transfer function analysis:

- The s **variable** is an expression of growing/decaying sinusoidal waves, comprised of a real part and an imaginary part ($s = \sigma + j\omega$). The real part of s (σ) is the growth/decay rate, telling us how rapidly the signal's magnitude grows or decays over time, with positive values of σ representing growth and negative values of σ representing decay. This growth/decay rate is the reciprocal of *time constant* ($\sigma = 1/\tau$), and is measured in reciprocal units of time (time constants per second, or s^{-1}). The imaginary part of s ($j\omega$) represents the *frequency* of the sinusoidal quantity, measured in radians per second (also s^{-1}). All together, s represents the *complex frequency* of a wave.

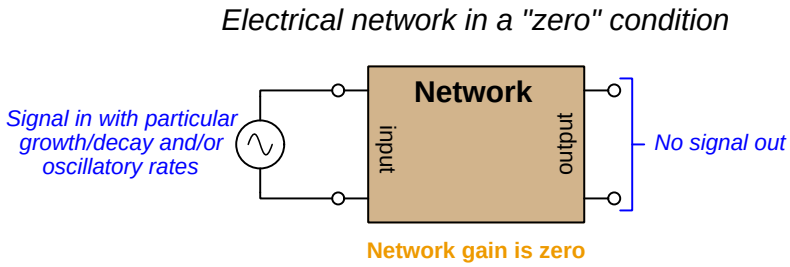
- An important assumption we make when analyzing any system's transfer function(s) is that the system is **linear** (i.e. its output and input magnitudes will be proportional to each other for all conditions) and **time-invariant** (i.e. the essential characteristics of the system do not change with the passage of time). If we wish to analyze a non-linear system using these tools, we must limit ourselves to ranges of operation where the system's response is approximately linear, and then accept small errors between the results of our analysis and the system's real-life response.

- For any linear time-invariant system (an "LTI" system), s is descriptive throughout the system. In other words, for a certain value of s describing the input to this system, that same value of s will also describe the output of that system: i.e. the system will not create any new frequency(ies) of its own when excited by one at the input.

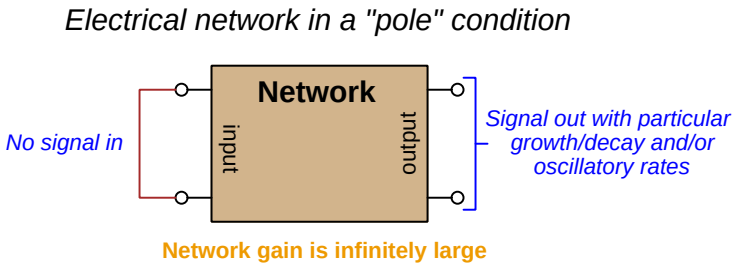
- A **transfer function** is an expression of a systems' *gain*, measured as a ratio of output over input. In engineering, transfer functions are typically mathematical functions of s (i.e. s is the independent variable in the formula). When expressed in this way, the transfer function for a system tells us how much gain the system will have for any given value of s.

- Transfer functions are useful for analyzing the behavior of electric circuits, but they are not limited to this application. Any linear system, whether it be electrical, mechanical, chemical, or otherwise, may be characterized by transfer functions and analyzed using the same mathematical techniques. Thus, transfer functions and the s variable are general tools, not limited to electric circuit analysis.

- A **zero** is any value of s resulting in zero gain, i.e. zero output for any magnitude of input. This tells us where the system will be least responsive. On a three-dimensional pole-zero plot, each zero appears as a low point where the surface touches the s plane. On a traditional two-dimensional pole-zero plot, each zero is marked with a circle symbol ($\circ$). We may solve for the zero(s) of a system by solving for value(s) of s that will make the *numerator* of the transfer function equal to zero, since the numerator of the transfer function represents the *output* term of the system. For electrical networks, a "zero" is any input signal resulting in no output signal:



- A **pole** is any value of s resulting in infinite gain, i.e. any output with zero input. This tells us what the system is capable of doing when it is not being "driven" by any input stimulus. Poles are typically associated with energy-storing elements in a passive system, because the only way an unpowered system could possibly generate an output with zero input is if there are energy-storing elements within that system discharging themselves to the output. On a three-dimensional pole-zero plot, each pole appears as a vertical spike on the surface reaching to infinity. On a traditional two-dimensional pole-zero plot, each pole is marked with a ($\times$) symbol. We may solve for the pole(s) of a system by solving for value(s) of s that will make the *denominator* of the transfer function equal to zero, since the denominator of the transfer function represents the *input* term of the system. For electrical networks, a "pole" is any output signal resulting from energy stored within the network with the input shorted:



- Only **multi-order** systems are capable of self-oscillation. This is revealed by poles containing imaginary (i.e. non-zero $j\omega$) values. These oscillations may be completely undamped (i.e. s is entirely imaginary, with $\sigma = 0$), in which case the system is able to oscillate forever on its own. If energy-dissipating elements are present in a second-order system, the oscillations will

be damped (i.e. decay in magnitude over time). By contrast, first-order systems can never oscillate because their poles do not contain imaginary values (i.e. $j\omega = 0$) but only real values (σ).

- An **under-damped** system exhibits complex poles, with s having both imaginary ($j\omega$) frequency values and real (σ) decay values. This means the system can self-oscillate, but only with decreasing magnitude over time. In practical terms this means the system will oscillate as it returns to a stable condition after being perturbed.

- A **critically damped** system is one having just enough dissipative behavior to completely prevent self-oscillation, exhibiting a single pole having only a real (σ) value and no imaginary ($j\omega$) value. In practical terms this means the system returns to a stable condition as rapidly as possible after being perturbed.

- An **over-damped** system is one having excessive dissipation, exhibiting multiple poles having only real (σ) values and no imaginary ($j\omega$) values. Each of these real poles represents a different decay rate (σ) or time constant ($\tau = 1/\sigma$) in the system. When these decay rates differ in value substantially from one another, the slowest one (i.e. the pole having the σ value nearest zero) dominates the behavior of the system over long periods of time. In practical terms this means the system returns to a stable condition without oscillating after being perturbed, just not as rapidly as it would if it were critically damped.

Chapter 4

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.

4.1 Derivation of inductive impedance

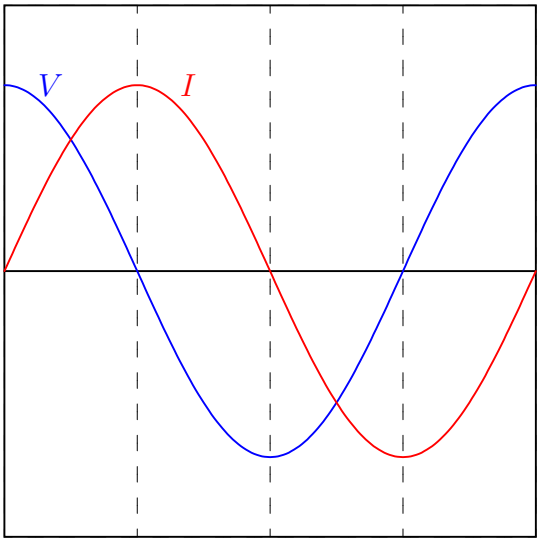
The relationship between voltage (V) and current (I) for a pure[1] inductance (L) is based on rates of change, specifically the rate-of-change of current over time ($\frac{dI}{dt}$). One might think of the following equation as being the inductive equivalent of Ohm's Law, because it fundamentally relates voltage and current for any inductance:

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

Where,

V = Voltage across the inductance's terminals (Volts)
L = Inductance (Henrys)
$\frac{dI}{dt}$ = Rate-of-change of current over time (Amperes per second)

If we pass a sinusoidal AC current through an inductance, the voltage at any given point in time will be proportional to the current's instantaneous rate-of-change. This means the voltage will be zero at those points in time when the current waveform is *level* (i.e. zero slope, at every peak) and maximum when the current waveform crosses the zero line (i.e. at maximum steepness). These notable points in time are marked on the following plot by dashed lines:



Therefore, when we apply a sinusoidal current through a pure inductor, the voltage waveform is also a sinusoid but phase-shifted by 90 degrees leading. In other words, *voltage leads current by 90 degrees for a pure inductance.*

[1] By "pure" it is meant that the inductance in question dissipates no energy whatsoever, but losslessly absorbs and releases energy. This is nearly impossible to achieve in practice, but superconducting inductors come very close to this ideal.

This same conclusion may be reached symbolically through the application of simple calculus, if we substitute a time-based sine function for I:

$$V = L\frac{dI}{dt} \qquad \text{and} \qquad I = \sin t$$

$$V = L\frac{d}{dt}(\sin t)$$

$$V = L\cos t$$

Since the derivative ($\frac{d}{dt}$) of a sine function is a cosine function, the voltage waveform must be a cosine wave, and we know cosine waves lead sine waves by 90 degrees.

We may use a more sophisticated approach of defining the AC current as a complex exponential function[2] of time rather than merely as a trigonometric function of time:

$$V = L\frac{dI}{dt} \qquad \text{and} \qquad I = e^{j\omega t}$$

$$V = L\frac{d}{dt}(e^{j\omega t})$$

$$V = j\omega L e^{j\omega t}$$

Just as electrical resistance is mathematically defined as the ratio of voltage to current ($R = \frac{V}{I}$), impedance is also defined as the ratio of voltage to current. Taking the voltage and current exponential functions together as a ratio, we see that inductive impedance reduces to a simple expression lacking any exponential term:

$$Z = \frac{V}{I}$$

$$Z = \frac{j\omega L e^{j\omega t}}{e^{j\omega t}}$$

$$Z = j\omega L$$

Unlike V and I which are both rotating phasors having both real and imaginary components alternating over time, Z is a *constant* imaginary quantity. The magnitude of Z depends on both frequency (ω) and inductance (L): the higher the frequency, the greater the impedance; the greater the inductance, the greater the impedance. The positive imaginary "direction" of Z (i.e. a phasor angle of 90^o) reflects the constant 90 degree phase shift between voltage (leading) and current (lagging) for an inductance. Expressed in non-phasor form, ωL is the *reactance* (X) of the inductor. When expressing frequency in Hertz rather than radians per second, it is more common to see reactance computed as $X_L = 2\pi f L$.

[2] Recall Euler's Relation, where $e^{ix} = \cos x + i \sin x$. Using j rather than i to signify the imaginary unit quantity, and using ω to signify the angular velocity (frequency) of the waveform, the time-domain wave function will be $e^{j\omega t}$ which is equal to $\cos \omega t + j \sin \omega t$. This is the *phasor* expression of the current waveform, having both a real and an imaginary component each of which varies over time. The advantage of this approach is that exponential functions are extremely simple to differentiate, as $\frac{d}{dx}e^{kx} = ke^{kx}$.

An even more sophisticated and versatile expression of an AC voltage uses *complex frequency* represented by the variable s, where s is the complex sum of a real growth/decay rate (σ) and an imaginary natural frequency ($j\omega$):

$$s = \sigma + j\omega$$

Where,
s = Complex frequency in $\frac{1}{\text{seconds}}$
σ = Growth/decay rate in time constants per second
ω = Natural frequency in radians per second

Applying this complex frequency as a current through the inductor and differentiating the inductive "Ohm's Law" formula:

$$V = L\frac{dI}{dt} \qquad \text{and} \qquad I = e^{st}$$

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

$$V = sLe^{st}$$

Defining inductive impedance as the ratio of inductor voltage to inductor current:

$$Z = \frac{V}{I}$$

$$Z = \frac{sLe^{st}}{e^{st}}$$

$$Z = sL$$

The impedance (in Ohms) is now a function of complex frequency, varying with growth/decay rate as well as with plain frequency. The real growth/decay rate dictates whether the component will experience a net absorption or a net release of energy over time (i.e. behaving as a load versus as a source) while the imaginary frequency dictates how often energy gets absorbed and released.

4.2 Derivation of capacitive impedance

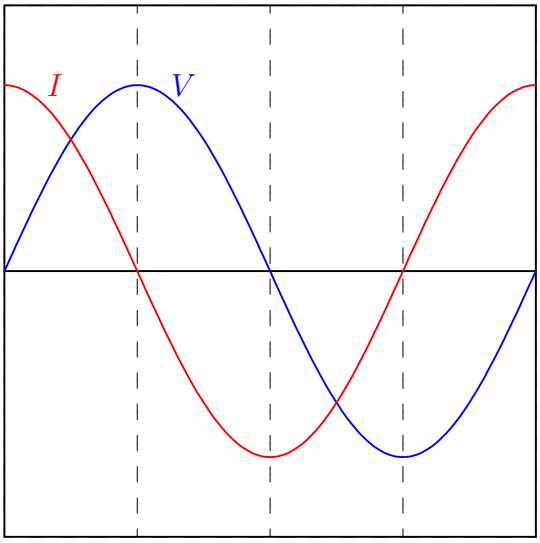
The relationship between voltage (V) and current (I) for a pure[3] capacitance (C) is based on rates of change, specifically the rate-of-change of voltage over time ($\frac{dV}{dt}$). One might think of the following equation as being the capacitive equivalent of Ohm's Law, because it fundamentally relates voltage and current for any capacitance:

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

Where,

I = Current through the capacitance's terminals (Amperes)
C = Capacitance (Farads)
$\frac{dV}{dt}$ = Rate-of-change of voltage over time (Volts per second)

If we impress a sinusoidal AC voltage across a capacitance, the current at any given point in time will be proportional to the voltage's instantaneous rate-of-change. This means the current will be zero at those points in time when the voltage waveform is *level* (i.e. zero slope, at every peak) and maximum when the voltage waveform crosses the zero line (i.e. at maximum steepness). These notable points in time are marked on the following plot by dashed lines:



Therefore, when we apply a sinusoidal voltage across a pure capacitance, the current waveform is also a sinusoid but phase-shifted by 90 degrees leading. In other words, *current leads voltage by 90 degrees for a pure capacitance.*

[3] By "pure" it is meant that the capacitance in question dissipates no energy whatsoever, but losslessly absorbs and releases energy. Real capacitors come remarkably close to this ideal.

This same conclusion may be reached symbolically through the application of simple calculus, if we substitute a time-based sine function for V:

$$I = C\frac{dV}{dt} \qquad \text{and} \qquad V = \sin t$$

$$I = C\frac{d}{dt}(\sin t)$$

$$I = C\cos t$$

Since the derivative ($\frac{d}{dt}$) of a sine function is a cosine function, the current waveform must be a cosine wave, and we know cosine waves lead sine waves by 90 degrees.

We may use a more sophisticated approach of defining the AC voltage as a complex exponential function[4] of time rather than merely as a trigonometric function of time:

$$I = C\frac{dV}{dt} \qquad \text{and} \qquad V = e^{j\omega t}$$

$$I = C\frac{d}{dt}(e^{j\omega t})$$

$$I = j\omega C e^{j\omega t}$$

Just as electrical resistance is mathematically defined as the ratio of voltage to current ($R = \frac{V}{I}$), impedance is also defined as the ratio of voltage to current. Taking the voltage and current exponential functions together as a ratio, we see that capacitive impedance reduces to a simple expression lacking any exponential term:

$$Z = \frac{V}{I}$$

$$Z = \frac{e^{j\omega t}}{j\omega C e^{j\omega t}}$$

$$Z = \frac{1}{j\omega C} \qquad \text{or} \qquad Z = -j\frac{1}{\omega C}$$

Unlike V and I which are both rotating phasors having both real and imaginary components alternating over time, Z is a *constant* imaginary quantity. The magnitude of Z depends on both frequency (ω) and capacitance (C): the higher the frequency, the less the impedance; the greater the capacitance, the less the impedance. The negative imaginary "direction" of Z (i.e. a phasor angle of -90^o) reflects the constant 90 degree phase shift between voltage (lagging) and current (leading) for a capacitance. Expressed in non-phasor form, $\frac{1}{\omega C}$ is the *reactance* (X) of the capacitance. When

[4] Recall Euler's Relation, where $e^{ix} = \cos x + i \sin x$. Using j rather than i to signify the imaginary unit quantity, and using ω to signify the angular velocity (frequency) of the waveform, the time-domain wave function will be $e^{j\omega t}$ which is equal to $\cos \omega t + j \sin \omega t$. This is the *phasor* expression of the current waveform, having both a real and an imaginary component each of which varies over time. The advantage of this approach is that exponential functions are extremely simple to differentiate, as $\frac{d}{dx}e^{kx} = ke^{kx}$.

expressing frequency in Hertz rather than radians per second, it is more common to see capacitive reactance computed as $X_C = \frac{1}{2\pi f C}$.

An even more sophisticated and versatile expression of an AC voltage uses *complex frequency* represented by the variable s, where s is the complex sum of a real growth/decay rate (σ) and an imaginary natural frequency ($j\omega$):

$$s = \sigma + j\omega$$

Where,
s = Complex frequency in $\frac{1}{\text{seconds}}$
σ = Growth/decay rate in time constants per second
ω = Natural frequency in radians per second

Applying this complex frequency as a voltage across the capacitor and differentiating the capacitive "Ohm's Law" formula:

$$I = C\frac{dV}{dt} \qquad \text{and} \qquad V = e^{st}$$

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

$$I = sCe^{st}$$

Defining capacitive impedance as the ratio of capacitor voltage to capacitor current:

$$Z = \frac{V}{I}$$

$$Z = \frac{e^{st}}{sCe^{st}}$$

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

The impedance (in Ohms) is now a function of complex frequency, varying with growth/decay rate as well as with plain frequency. The real growth/decay rate dictates whether the component will experience a net absorption or a net release of energy over time (i.e. behaving as a load versus as a source) while the imaginary frequency dictates how often energy gets absorbed and released.

4.3 Definition of negative frequency

A bewildering concept one encounters when determining transfer functions for electrical circuits is something known as *negative frequency*. We see this in the transfer functions of circuits containing both inductance and capacitance, where multiple poles exist having imaginary (or complex) values. In such cases, the poles having imaginary values always exist as pairs with equal positive and negative imaginary frequency values.

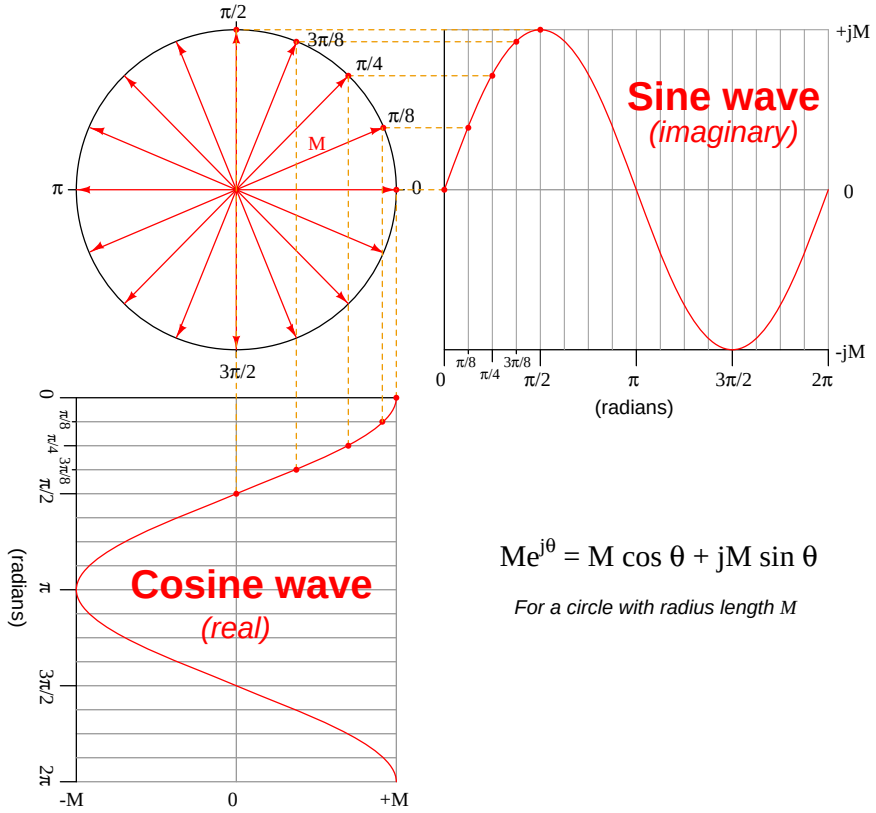
Complex frequency, as we know, consists of both real and imaginary terms, each measured in units of "per second" or s^{-1}. The real portion of a complex frequency value describes how rapidly the signal grows or decays over time, while the imaginary portion describes how rapidly the signal oscillates over time. While the distinction between a positive and negative *real* value for a complex frequency is easy to grasp – positive being growth over time and negative being decay over time – the distinction between positive versus negative imaginary (oscillatory) complex-frequency values is less clear.

Positive versus negative oscillatory frequency is a direct consequence of defining sinusoidal oscillations by Euler's Relation, which shows the mathematical equivalence between imaginary-exponential functions and trigonometric (sine and cosine) functions:

$$e^{j\theta} = \cos\theta + j\sin\theta$$

In this basic form of Euler's Relation, the angle (θ) is necessarily expressed in mathematically natural unit of *radians* as opposed to the more familiar unit of *degrees* most people use. To see how Euler's Relation gives rise to the possibility of both positive and negative frequency, it helps to graph the results of Euler's Relation as we will do next.

When we mathematically define sinusoidal oscillations by $e^{j\theta}$ we necessarily invoke *two* sinusoidal functions, cosine and sine. These two trigonometric functions, when associated with the real and imaginary axes of a graph, result in a *circle* being traced by the real (horizontal, cosine) and imaginary (vertical, sine) function values as shown in the following diagram:



At an angle (θ) of zero the value of cosine is +1 and the value of sine is zero, placing the coordinate point at the right-hand edge of the circle (marked by a 0). As the angle grows in a positive direction, the cosine function begins to decrease while the sine function increases. The resulting progression of that coordinate point along the circle's circumference is in the counter-clockwise direction, as indicated by the progressively larger angle values shown in the diagram ($\frac{\pi}{8}$, $\frac{\pi}{4}$, etc.).

The fact that combining two out-of-phase sinusoidal functions results in a definite *rotation* should be familiar to anyone who has ever studied polyphase AC circuits. This is precisely why polyphase AC circuits were originally invented, and that was to make possible the creation of a *rotating magnetic field* by sets of wire coils energized by the out-of-phase AC currents. When two or more out-of-phase sinusoids interact within a two-dimensional space, the result is something that *moves* circularly. Any single sinusoid would be incapable of expressing a definite rotational direction, but merely a back-and-forth oscillation.

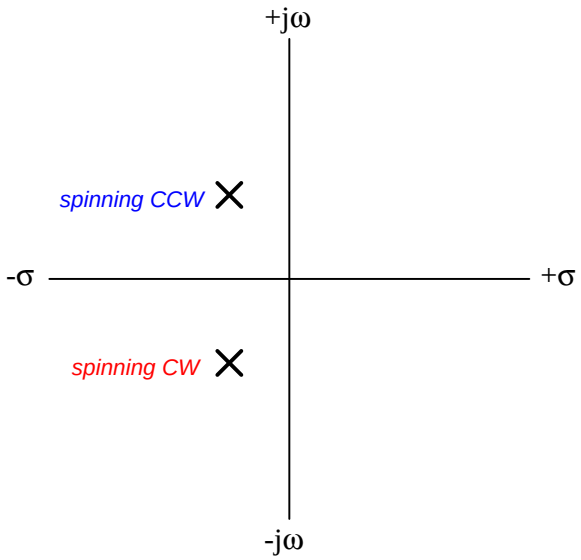
Euler's Relation, then, with its two out-of-phase (by 90 degrees, or $\frac{\pi}{2}$ radians) sinusoids traces a circle with a counter-clockwise rotation as the angle θ grows larger and more positive. A *negative* frequency, then, would have to be one where the direction of rotation is opposite: clockwise instead of counter-clockwise. This is precisely what is being expressed by a transfer function yielding pairs of complex frequency values where the imaginary portions are opposite in sign: *equal and opposite positive- versus negative-rotation oscillations.*

We may mathematically express these positive- and negative-rotating signals by modifying Euler's Relation:

$$e^{j\theta} = \cos\theta \pm j \sin\theta$$

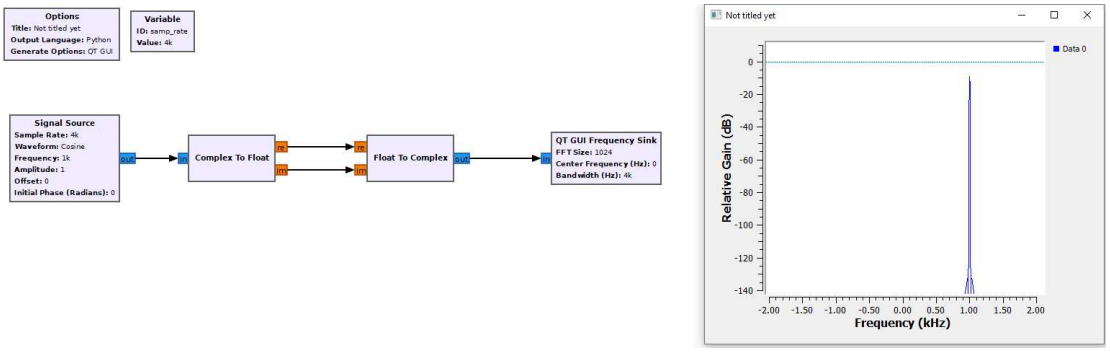
The $\pm$ symbol suggests both positive and negative imaginary terms, which is what we get when solving for roots of a transfer function with a negative discriminant quantity. A negative discriminant in the quadratic equation means the square-root will be of a negative quantity (i.e. imaginary), and then the $\pm$ sign in the quadratic equation results in positive *and* negative imaginary portions for the resulting complex frequency.

In the context of transfer functions, it means the system will have the exact same response to a signal having a negative oscillatory frequency as it would to a signal having a positive oscillatory frequency, which is why these exist in $\pm$ pairs and not alone. For example, if a rotating machine's vibration peaks at a particular shaft speed (i.e. a pole for the vibration transfer function) while spinning clockwise, its vibration will also peak when the shaft spins the same speed in the opposite direction. Those two poles lying opposite from each other across the real axis of the s plane describe the machine's ability to resonate equally well in both rotational directions:

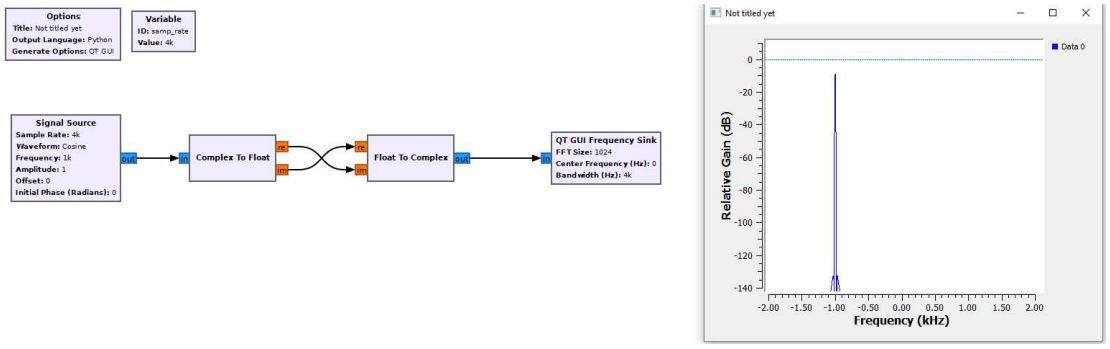


Similarly, an LC "tank" network will resonate if excited by the output of an AC generator at just the right frequency, regardless of which way the generator spins at that speed.

In digital signal processing where sinusoidal signals are often represented as complex numbers, we may demonstrate positive and negative frequencies by taking a simulated signal source and separating its complex output into two floating-point outputs, then recombining those floating-point outputs into complex again and plotting the result on a spectrum display. When the real and imaginary floating-point number streams are recombined in the same manner as they were separated, the resulting complex frequency will have the same (positive) value of +1 kHz as the original source value as shown[5] below:



However, if we swap real and imaginary components of the original complex signal value when recombining into a complex number, the result is a negative frequency of −1 kHz:



Changing the imaginary sign of a complex frequency is a very useful technique in digital signal processing, and as we can see here it may be accomplished by the same technique we would use to reverse the rotation of a polyphase electric motor: in the case of a two-phase motor we would simply swap connections between the two sets of stator coils (windings), or in a three-phase motor swap any two out of the three line conductors.

[5] The digital signal processing software used to create these signal "flowgraphs" is `GNU Radio Companion`.

Chapter 5

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*.

5.1 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.

5.2 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.

5.3 Impedance calculator with complex frequency using C++

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

int main (void)
{
  double f, L, C, omega, sigma;
  complex <double> s, Zc, Zl;

  cout << "Enter frequency (f) in Hertz = " ;
  cin >> f;
  omega = 2.0 * M_PI * f;

  cout << "Enter growth/decay rate (sigma) in time constants per second = " ;
  cin >> sigma;

  cout << "Enter inductor value in Henrys = " ;
  cin >> L;

  cout << "Enter capacitor value in Farads = " ;
  cin >> C;

  s = complex <double> (sigma, omega);

  Zl = s * L;
  Zc = 1.0 / (s * C);

  cout << "Z_L = " << abs (Zl) << " Ohms @ " << arg (Zl) * 180 / M_PI
    << " deg " << endl;
  cout << "Z_L = " << real (Zl) << " Ohms + " << imag (Zl) << "j Ohms" << endl;
  cout << endl;
  cout << "Z_C = " << abs (Zc) << " Ohms @ " << arg (Zc) * 180 / M_PI
    << " deg " << endl;
  cout << "Z_C = " << real (Zc) << " Ohms " << imag (Zc) << "j Ohms" << endl;

  return 0;
}
```

5.4 Complex frequency plotter using C++

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

int main (void)
{
  double sigma = -0.3, omega = 6.0;
  double time, value, incr = 0.04;

  cout << "Time , Value" << endl;

  for (time = 0 ; time < 10.0 ; time = time + incr)
  {
    value = exp(sigma * time) * cos(omega * time);
    cout << time << " , " << value << endl;
  }

  return 0;
}
```

This particular program as written generates comma-separated value (CSV) plain-text output which may be imported into a spreadsheet or into mathematical visualization software such as Gnuplot to create a simple time-domain plot. For simplicity, values for growth/decay rate (σ) and oscillatory rate (ω) are fixed in the C++ source code rather than input at run-time by the user. To plot different $\sigma + j\omega$ values simply edit the source code, re-compile, and re-run the executable file.

I happened to use `gnuplot` to generate the spectra. My gnuplot script is as follows, saved to a file named `script.txt`:

```
set datafile separator ","
set style line 1 lw 2
plot 'data.csv' using 1:2 with lines ls 1
```

The C++ program was compiled using `g++` and run with text output redirected to a file named `data.csv` using the following command-line instructions:

```
g++ main.cpp ; ./a.out > data.csv
```

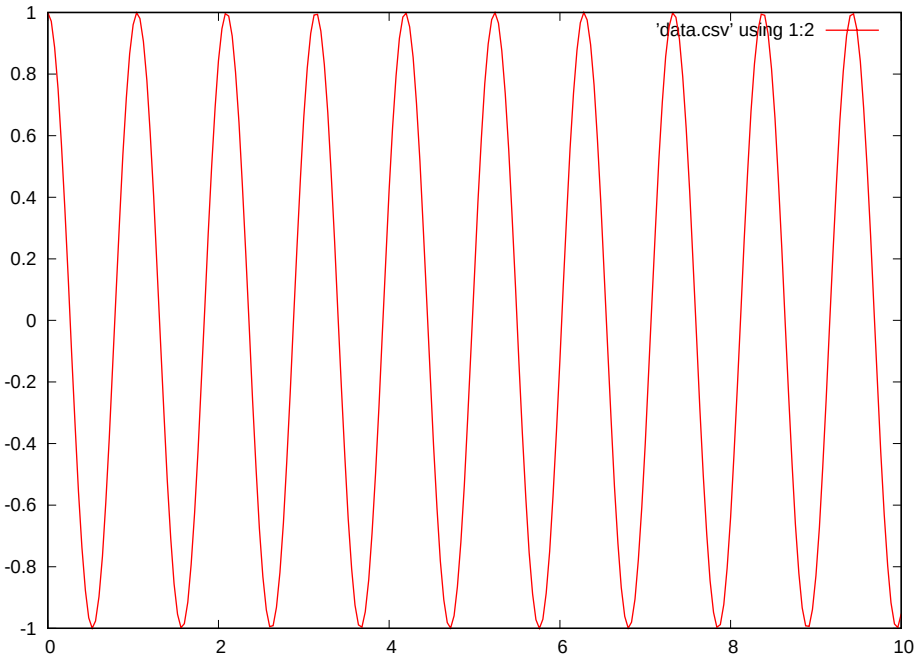
Then, after the comma-separated value file was populated with data from the C++ program's execution, I ran `gnuplot` using the following command:

```
gnuplot -p script.txt
```

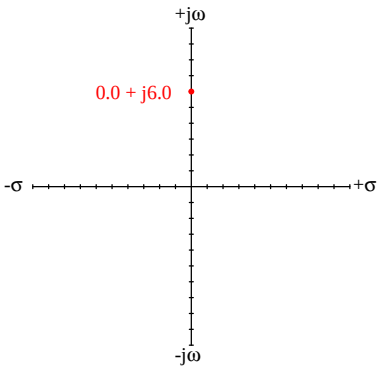
For each of the following outputs, I show the `sigma` and `omega` variable assignments as well as how s appears written as a complex number in rectangular form, followed by the `gnuplot` graphic display of the waveform over a timespan from 0 to 10.0 seconds. The purpose of these examples is to illustrate the relationship between σ, ω, and the resulting sinusoidal waveform.

```
double sigma = 0.0, omega = 6.0;
```

Complex frequency value: $s = 0 + j6$ per second

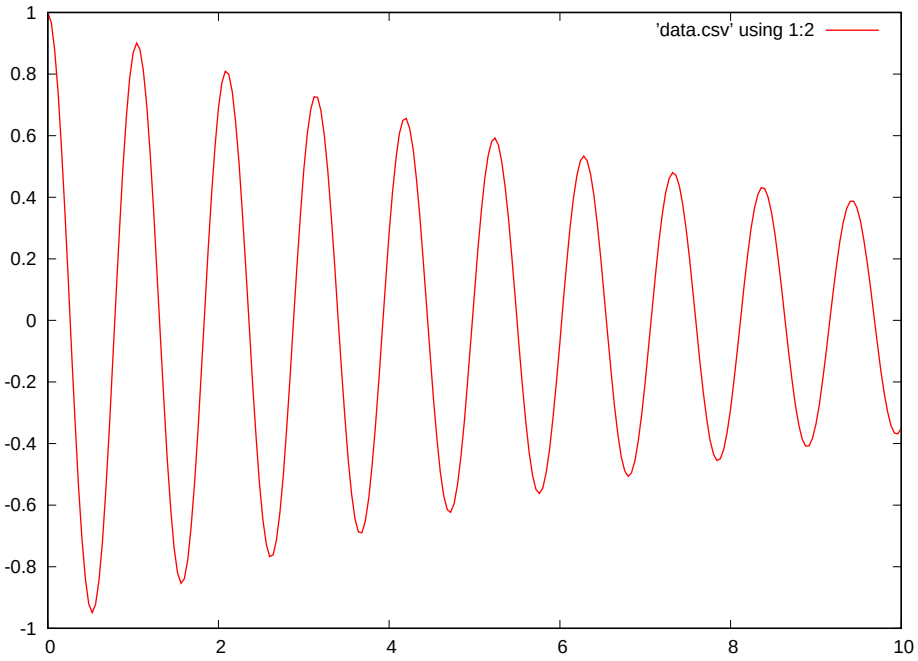


This shows a waveform with constant peak amplitude over time, since σ (the growth/decay rate) is equal to zero. Shown as a point on the s plane:

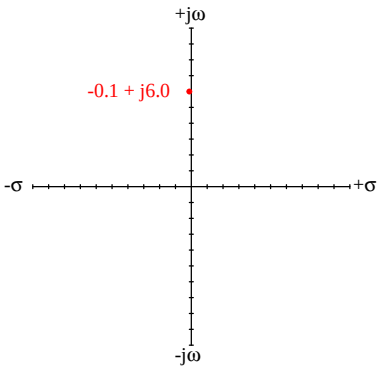


```
double sigma = -0.1, omega = 6.0;
```

Complex frequency value: $s = -0.1 + j6$ per second

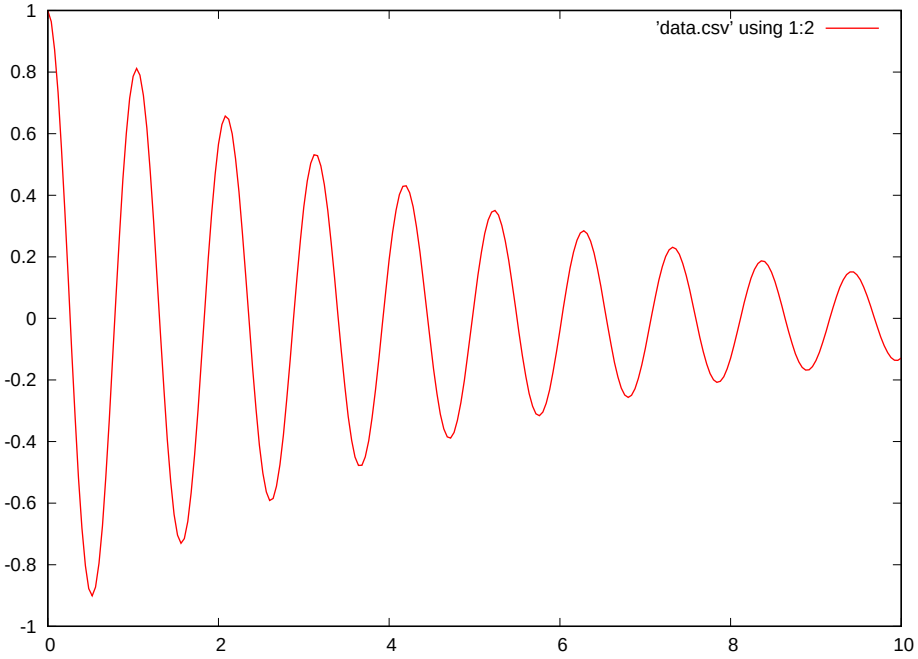


This shows a waveform with decaying amplitude over time, since σ (the growth/decay rate) is negative. Shown as a point on the s plane:

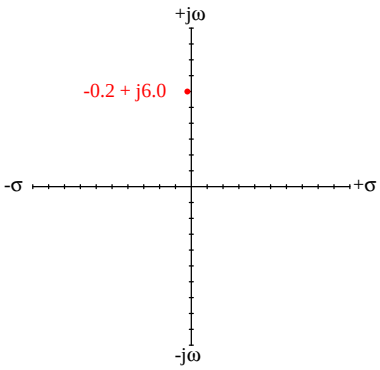


```
double sigma = -0.2, omega = 6.0;
```

Complex frequency value: $s = -0.2 + j6$ per second

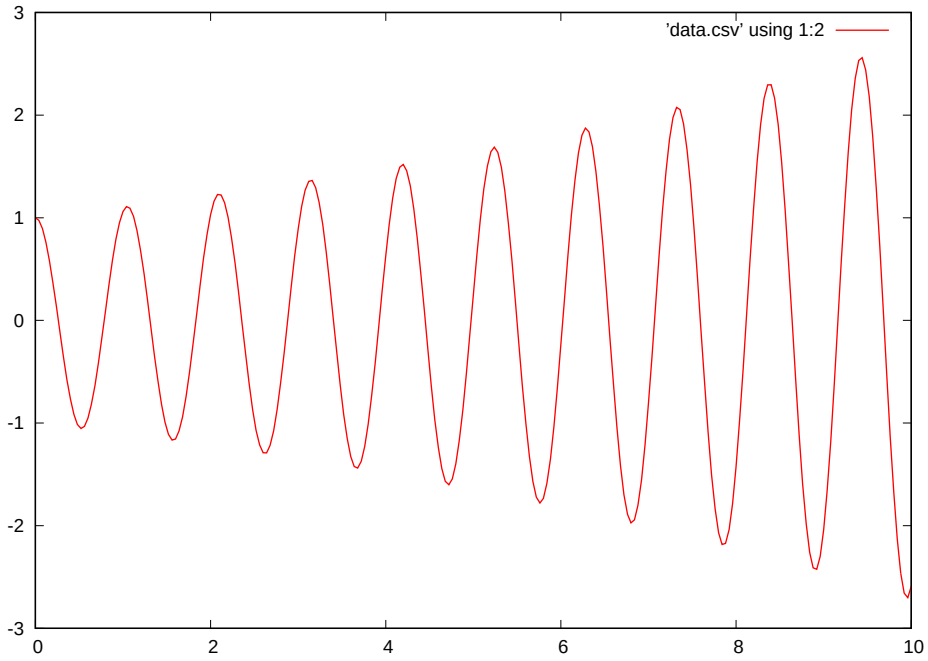


This shows a waveform with a more rapidly decaying amplitude over time, since σ (the growth/decay rate) is a stronger negative number. Shown as a point on the s plane:

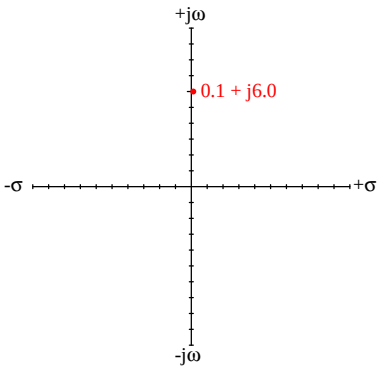


```
double sigma = 0.1, omega = 6.0;
```

Complex frequency value: $s = 0.1 + j6$ per second

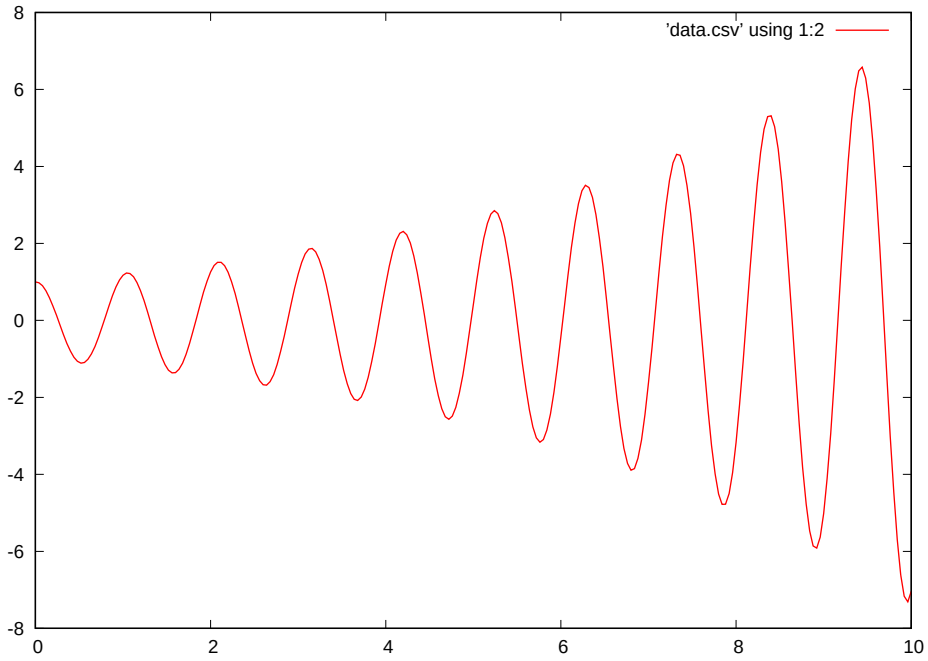


This shows a waveform with a growing amplitude over time, since σ (the growth/decay rate) is a positive number. Shown as a point on the s plane:

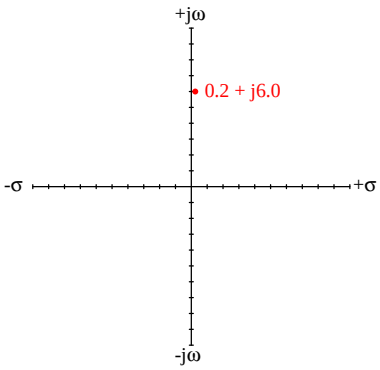


```
double sigma = 0.2, omega = 6.0;
```

Complex frequency value: $s = 0.2 + j6$ per second

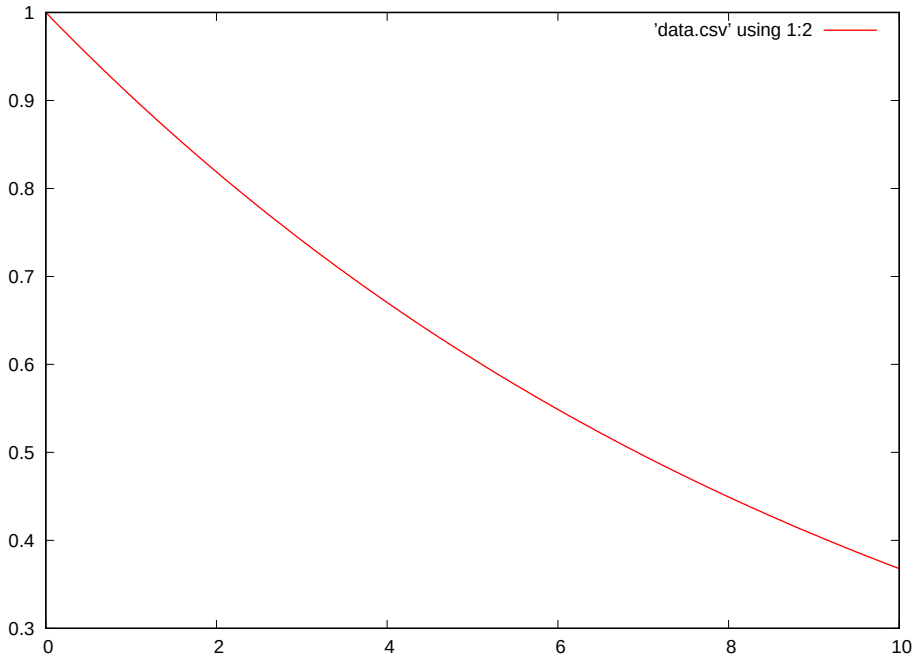


This shows a waveform with a more rapidly growing amplitude over time, since σ (the growth/decay rate) is a stronger positive number. Shown as a point on the s plane:

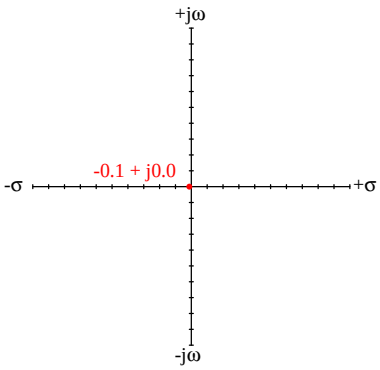


```
double sigma = -0.1, omega = 0.0;
```

Complex frequency value: $s = -0.1 + j0$ per second

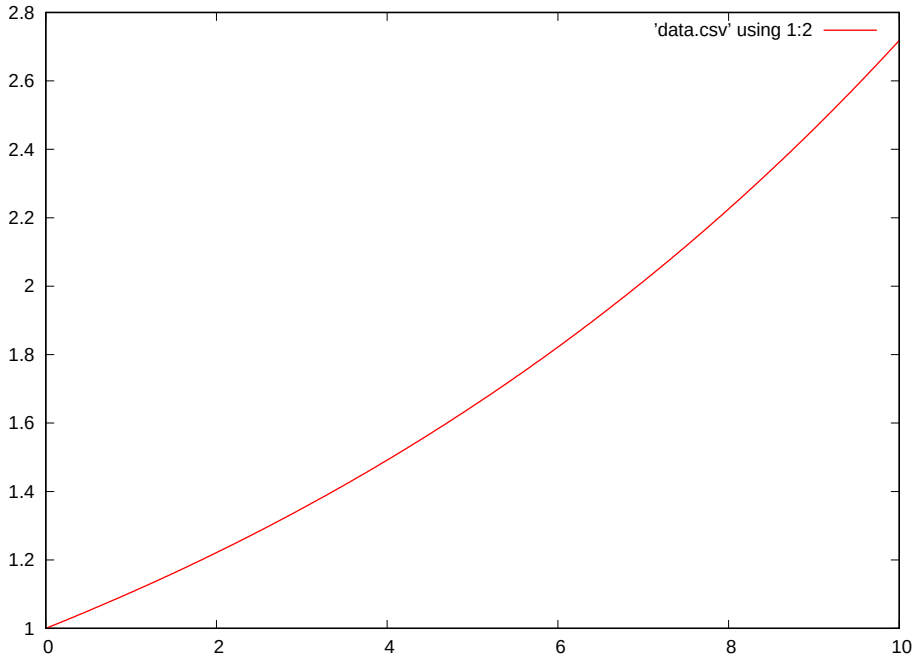


This shows a non-oscillating decay, since σ (the growth/decay rate) is a negative number and ω (the oscillation rate) is zero. Shown as a point on the s plane:

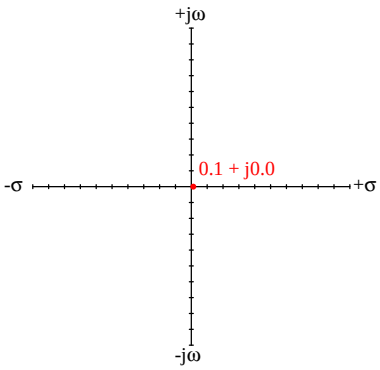


```
double sigma = 0.1, omega = 0.0;
```

Complex frequency value: $s = 0.1 + j0$ per second

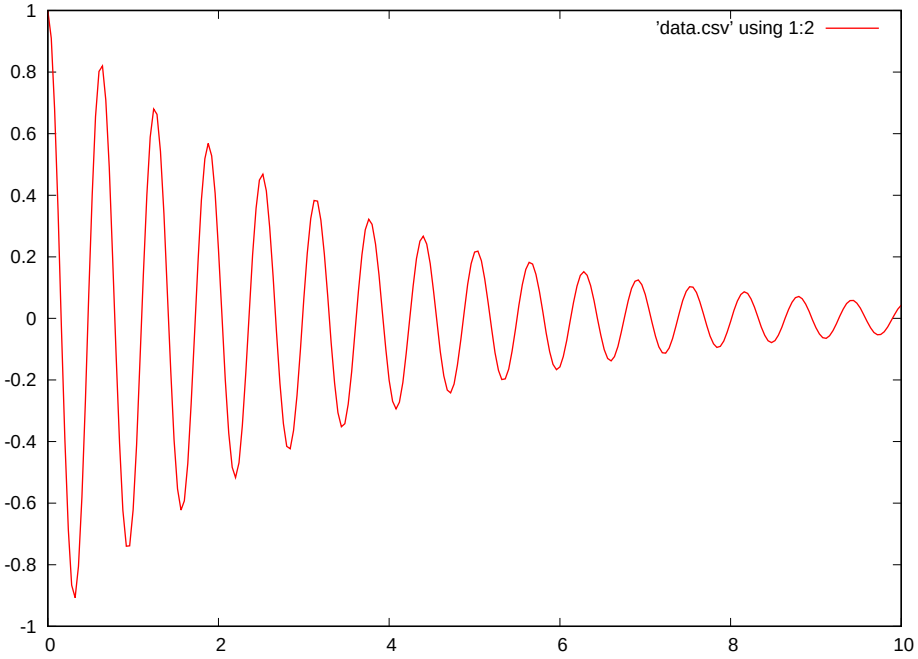


This shows a non-oscillating growth, since σ (the growth/decay rate) is a positive number and ω (the oscillation rate) is zero. Shown as a point on the s plane:

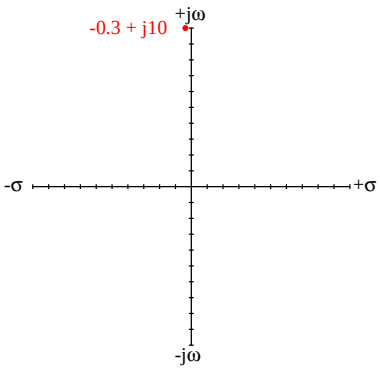


```
double sigma = -0.3, omega = 10.0;
```

Complex frequency value: $s = -0.3 + j10$ per second

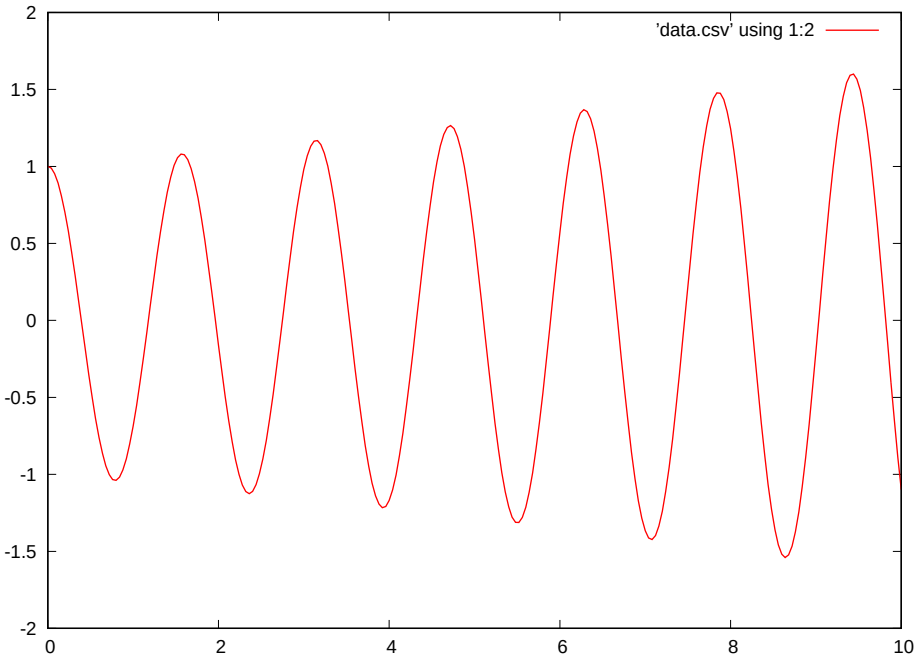


This shows a higher-frequency and faster-decaying waveform than any of the others. Shown as a point on the s plane:

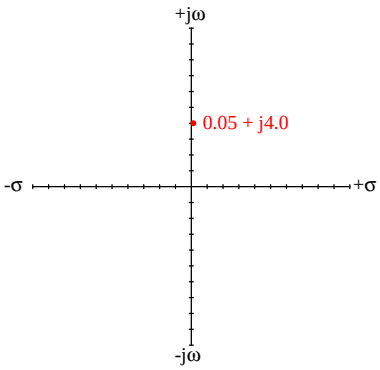


```
double sigma = 0.05, omega = 4.0;
```

Complex frequency value: $s = 0.05 + j4$ per second



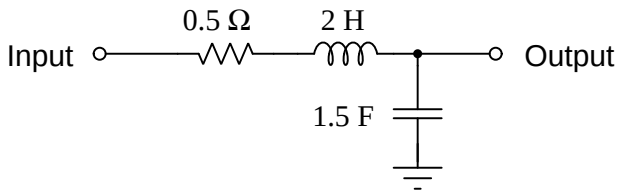
This shows a lower-frequency and slower-growing waveform than any of the others. Shown as a point on the s plane:



5.5 Pole-zero plotter using C++

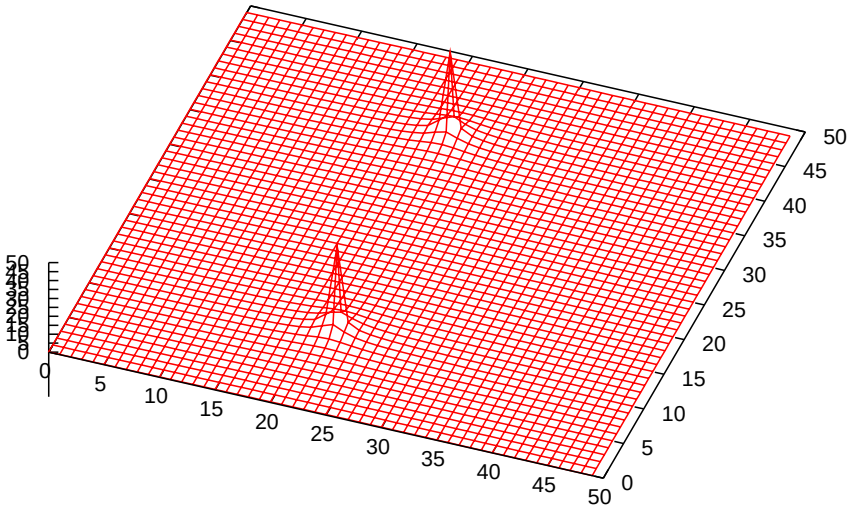
```
#include <iostream>
#include <complex>
#include <cmath>
using namespace std;

double TransferFunction (complex <double>, double);

int main (void)
{
  double sigma, sigma_lo = -1.0, sigma_hi = 1.0;
  double omega, omega_lo = -1.0, omega_hi = 1.0, incr = 0.04;
  complex <double> s;

  for (omega = omega_lo ; omega <= omega_hi ; omega = omega + incr)
  {
    for (sigma = sigma_lo ; sigma <= sigma_hi ; sigma = sigma + incr)
    {
      s = complex <double> (sigma, omega);
      cout << TransferFunction(s, incr / 10.0) << ",";
    }
    cout << endl;
  }

  return 0;
}

double TransferFunction (complex <double> s, double min)
{
  double L = 2.0, C = 1.5, R = 0.5;
  complex <double> Zl, Zc;

  Zl = s * L;

  if (s == 0.0)  // Guards against divide-by-zero error
    s = min;

  Zc = 1.0 / (s * C);

  return abs(Zc / (Zc + Zl + R));  // Double-pole low-pass RLC filter
}
```

This particular program as written generates comma-separated value (CSV) plain-text output which may be imported into a spreadsheet or into mathematical visualization software such as Gnuplot to create a simple three-dimensional pole-zero plot. The circuit being simulated by this example code is the following low-pass RLC filter:



The two nested `for()` loops sweep σ and ω values both from -1.0 per second to 1.0 per second in 0.04 per second increments. This results in a comma-separated value text dataset consisting of 50 rows of text, each with 50 data fields (columns) for numerical values. Each of these values is the gain of the network, calculated as the absolute value of the transfer function, at a particular complex-frequency (s) value. Each row of text data represents a particular natural frequency (ω) value for s, while each comma-separated data fields in a row represents a different growth/decay rate (σ) value for s. Since the program does not include these imaginary and real values for s (but only the network gain at each s value), the plotting software does not have any way to label the x and y axes with actual growth/decay or frequency values. This limitation is not as debilitating as it may seem, though. Since zero is located at the center of each of these axes, it is not difficult to start at the center and then count individual lines as though they are divisions of measurement on an oscilloscope screen or an analog multimeter, the value of each division being the value set by `incr` in the C++ code.

An example of the three-dimensional plot generated from the resulting text file is shown here:



This particular simulation uses increments of 0.04 per second (`incr` in the C++ code), which is the value of each division (space between lines) on the two horizontal axes. The two poles of this filter network are clearly evident in the three-dimensional plot, but the height of these peaks is quite irrelevant to our purposes. We just want to know how many poles there are, and where those poles are located on the s plane.

Looking closely at this plot, we see the peaks of each pole are aligned with the line falling on "22" on the σ axis. This is three divisions to the left of center ($\sigma = 0$), which at 0.04 time constants per second per division means a decay rate of -0.12 time constants per second.

Examining the plot closely again, we see each pole equidistant from center on the ω axis by 14 divisions: one of the poles at the "11" mark and the other on the "39" mark. 14 divisions multiplied by 0.04 radians per second per vision means the natural frequency of our two poles are ± 0.56 radians per second.

If we write the transfer function for this low-pass filter and solve for the pole s values algebraically, we find close agreement with the plotted results:

$$\frac{\frac{1}{sC}}{\frac{1}{sC} + sL + R}$$

$$\frac{\frac{1}{sC}}{\frac{1}{sC} + \frac{s^2LC}{sC} + \frac{sRC}{sC}}$$

$$\frac{\frac{1}{sC}}{\frac{s^2LC+sRC+1}{sC}}$$

$$\frac{1}{s^2LC + sRC + 1}$$

Clearly this function lacks finite zeros with a constant denominator, and for poles we must resort to the quadratic formula to solve for those values of s resulting in $s^2LC + sRC + 1 = 0$. Using the $L = 2$, $C = 1.5$, and $R = 0.5$ values embedded in the C++ code:

$$s^2(2)(1.5) + s(0.5)(1.5) + 1 = 0$$

$$3s^2 + 0.75s + 1 = 0$$

Therefore, a is 3, b is 0.75, and c is 1 for the purpose of applying the quadratic formula:

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

$$s = \frac{-0.75 \pm \sqrt{0.75^2 - (4)(3)(1)}}{(2)(3)}$$

$$s = \frac{-0.75 \pm \sqrt{0.5625 - 12}}{6}$$

$$s = \frac{-0.75 \pm \sqrt{-11.4375}}{6}$$

$$s = \frac{-0.75 \pm j3.3819}{6}$$

$$s = -0.125 \pm j0.5637$$

As we can see here, the actual decay rate of -0.125 time constants per second closely matches what we read from the pole-zero plot (-0.12 time constants per second), the error being far less than the resolution for that plot. The actual natural frequency of ± 0.5637 radians per second is also a close match to what the plot showed as 0.56 radians per second, once again the error being well within the limits of resolution. This demonstrates the accuracy of our simple C++ pole-zero plotting program.

Chapter 6

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?

6.1 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.

6.1.1 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.

6.1.2 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

Electrical source

Electrical load

Limiting cases as a problem-solving strategy

Ohm's Law

Inductance

Capacitance

Reactance

Voltage divider

Frequency

Impedance

Phasor

Phase angle

Frequency domain

Complex numbers

Properties of series AC circuits

Properties of parallel AC circuits

Transfer function

Zero

Pole

Order

Angular velocity

Neper velocity

Complex frequency

“Ohm’s Law” for capacitance

"Ohm's Law" for inductance

Resonance

6.1.3 Explain this joke

Explain why this might be a funny joke to a control systems or aerospace flight controls engineer:

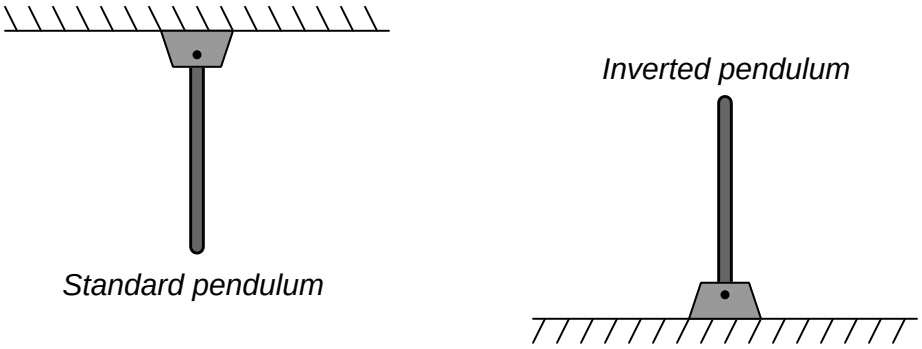
"May all your poles lie within the left half of the s plane"

Challenges

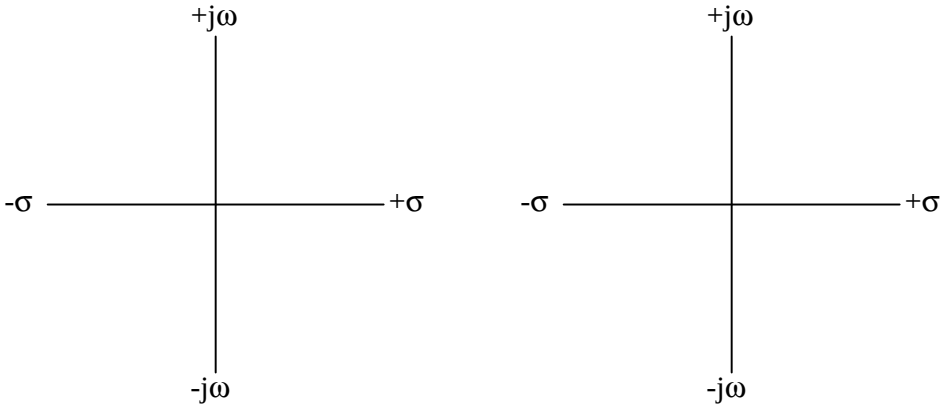
- Might there be some advantage to a system having poles in the right half of the s plane?

6.1.4 Pole plots for pendula

A standard pendulum starting in a perfectly vertical orientation, if nudged or struck, will swing back and forth until the oscillations gradually die out due to friction within the pivot as well as air resistance. An *inverted* pendulum starting in a perfectly vertical orientation, if nudged or struck, will simply fall over until it lands on the ground:



Qualitatively place "X" marker(s) for the pole locations for each of these physical systems:

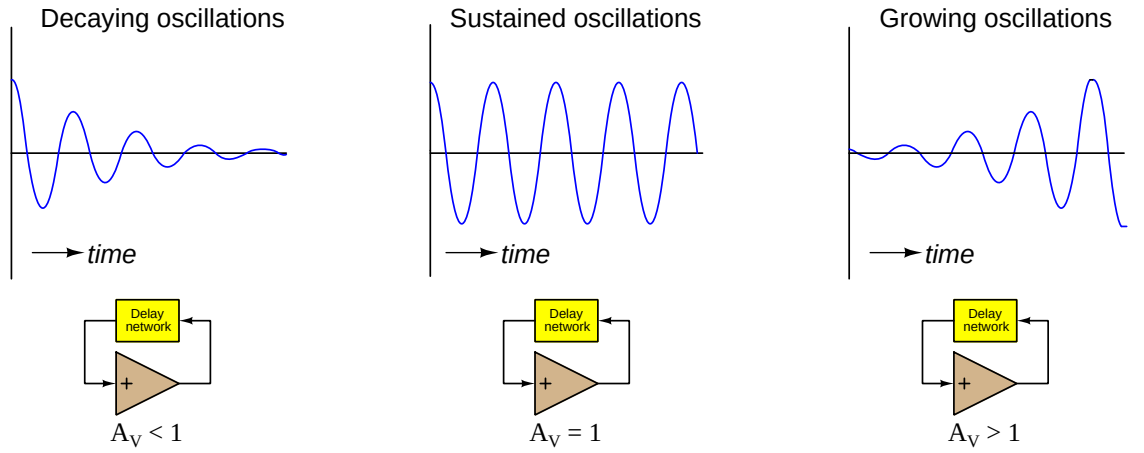


Challenges

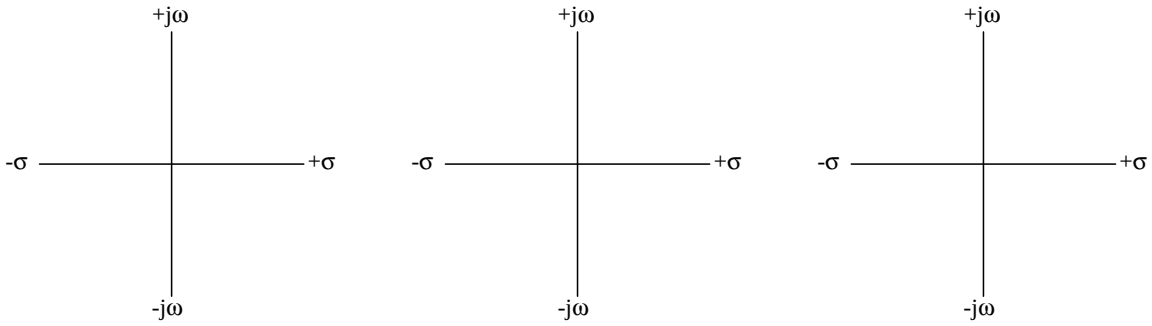
- How would the the poles be affected for both pendula if the pivots were lubricated to absolutely minimize friction there?
- How would the the poles be affected for both pendula if the pendulum arms were altered to be much longer than the originals?
- How would the the poles be affected for both pendula if the initial nudge were made more intense?
- What would a "zero" mean in a physical sense for either mechanism?

6.1.5 Pole plots for a feedback oscillator

Some electronic oscillators utilize an RC delay network along with *feedback* from a voltage-amplification circuit to create and sustain oscillations. The over-all gain of the feedback loop (including the delay network) must be carefully adjusted to equal unity (1), or else the oscillations will either die out or grow until the amplifier clips and distorts the waveform:



Sketch locations of poles for each of these oscillator loop-gain conditions, qualitatively placing "X" marker(s) for those locations on the s plane:



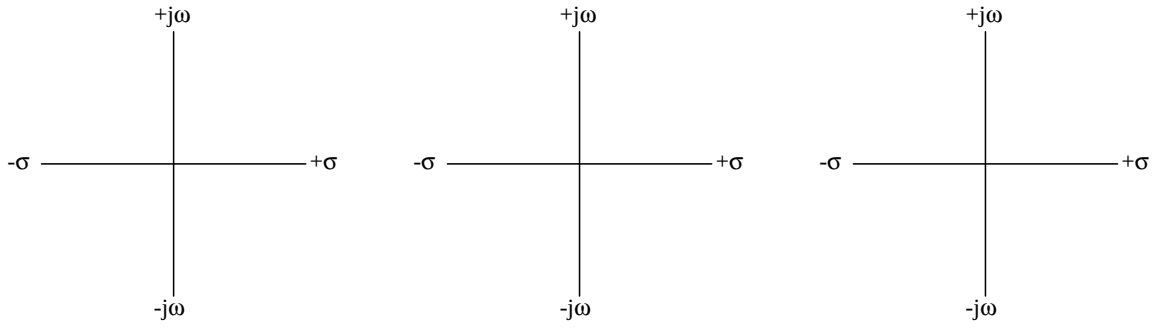
Challenges

- Explain why maintaining a perfect unity-gain condition for a feedback oscillator is so challenging.
- How may such an oscillator circuit be stabilized so that its over-all loop gain never falls below unity while also never distorting the waveform from unrestrained growth?

6.1.6 Pole plots for a bell

Bells, made from cast or forged metal, will "ring" with an audible tone when struck. Both the pitch of that ringing tone and the rate of decay for that ringing tone may be adjusted by varying physical dimensions of the bell.

On the three s plane plots shown below, mark plausible locations for poles describing a bell's acoustic behavior given (left) a moderate pitch with a moderate decay rate, (center) a higher pitch with the same moderate decay rate, and (right) a lower pitch with a very slow decay rate (i.e. to make the bell ring for a long time after being struck):

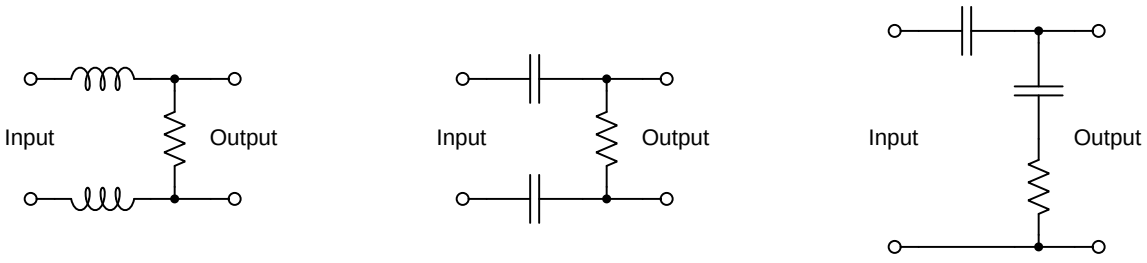


Challenges

- What physical parameters generally dictate a bell's pitch?
- What physical parameters generally dictate a bell's decay rate?
- Is it possible to have a bell with at least one pole on the *other* half of the s plane's real axis?

6.1.7 Network zero conditions

For each of these electrical networks, explain in qualitative terms only (i.e. no mathematical calculations) what input signal(s) would constitute a "zero" condition:



Challenges

- Will any of these networks exhibit zero conditions equally well with *either* high σ values or high ω values?

6.2 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.

6.2.1 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.

6.2.2 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.*

6.2.3 Exploring Euler's Relation

Euler's Relation shows the equivalence between imaginary-exponential functions and trigonometric (sine and cosine) functions:

$$e^{jx} = \cos x + j \sin x$$

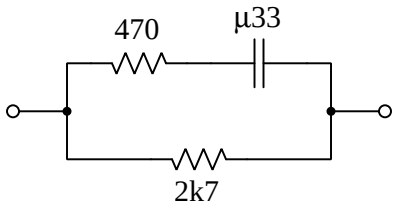
Use a scientific calculator to demonstrate the truth of Euler's Relation by way of specific numerical examples of your own choosing.

Challenges

- Explain why this is a useful tool for learning new mathematical concepts.

6.2.4 RC network impedances

Calculate the total impedance of this network for an applied sinusoidal signal frequency of 750 Hertz with a constant magnitude:



Now calculate the impedance of this same network for an applied signal of 900 Hertz growing exponentially in magnitude at a rate of 2200 time constants per second.

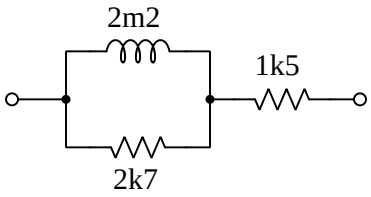
Now calculate the impedance of this same network for an applied signal of 500 Hertz decaying exponentially in magnitude at a rate of -7000 time constants per second.

Challenges

- What effect does the signal's growth or decay have on the capacitor's stored energy?

6.2.5 LR network impedances

Calculate the total impedance of this network for an applied sinusoidal signal frequency of 85 kiloHertz with a constant magnitude:



Now calculate the impedance of this same network for an applied signal of 70 kiloHertz growing exponentially in magnitude at a rate of 58000 time constants per second.

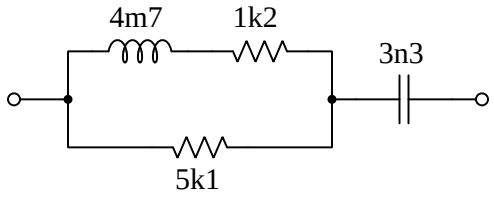
Now calculate the impedance of this same network for an applied signal of 90 kiloHertz decaying exponentially in magnitude at a rate of -88000 time constants per second.

Challenges

- What effect does the signal's growth or decay have on the inductor's stored energy?

6.2.6 RLC network impedances

Calculate the total impedance of this network for an applied sinusoidal signal frequency of 60 kiloHertz with a constant magnitude:



Now calculate the impedance of this same network for an applied signal of 55 kiloHertz growing exponentially in magnitude at a rate of 40000 time constants per second.

Now calculate the impedance of this same network for an applied signal of 40 kiloHertz decaying exponentially in magnitude at a rate of -95000 time constants per second.

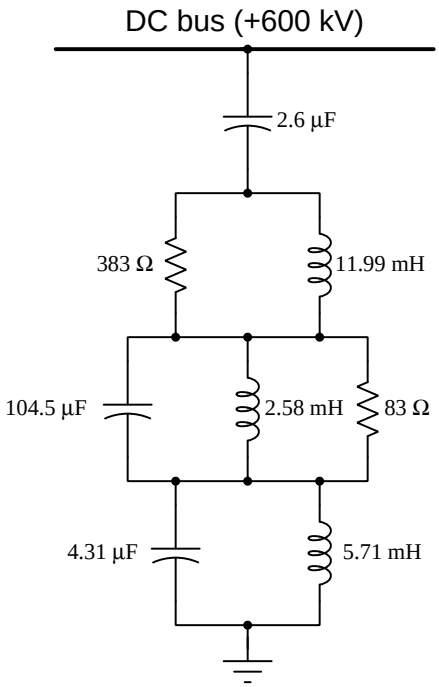
Challenges

- What effect does the signal's growth or decay have on the reactive components' stored energy?

6.2.7 HVDC filter harmonic analysis

High-voltage DC (HVDC) power converters are used for long-distance transmission of electricity at extremely high voltages, using semiconductor switches (e.g. transistors or thyristors) to convert three-phase AC into DC for transmission, then another set of semiconductor switches at the other end to invert DC back into three-phase AC, usually for distances of thousands of kilometers where the savings from reduced power losses offset the expense of the high-voltage switching facilities. The process of converting AC into DC and DC into AC, however, produces unwanted harmonic frequencies on both the AC and DC circuit sides.

RLC filter networks are used to attenuate these harmonics. Below we see a schematic diagram for one of these high-voltage filter networks, this particular one connected between the +600 kiloVolt DC bus line and Earth ground (there is another between the −600 Volt bus line and ground for this "bipolar" ± 600 kV transmission system) for the Porto Valeho to Araraquara transmission line in Brazil:



Identify the type of filter this is (e.g. low-pass, high-pass, band-pass, or band-stop) and also which harmonics of 60 Hz it is designed to attenuate.

Challenges

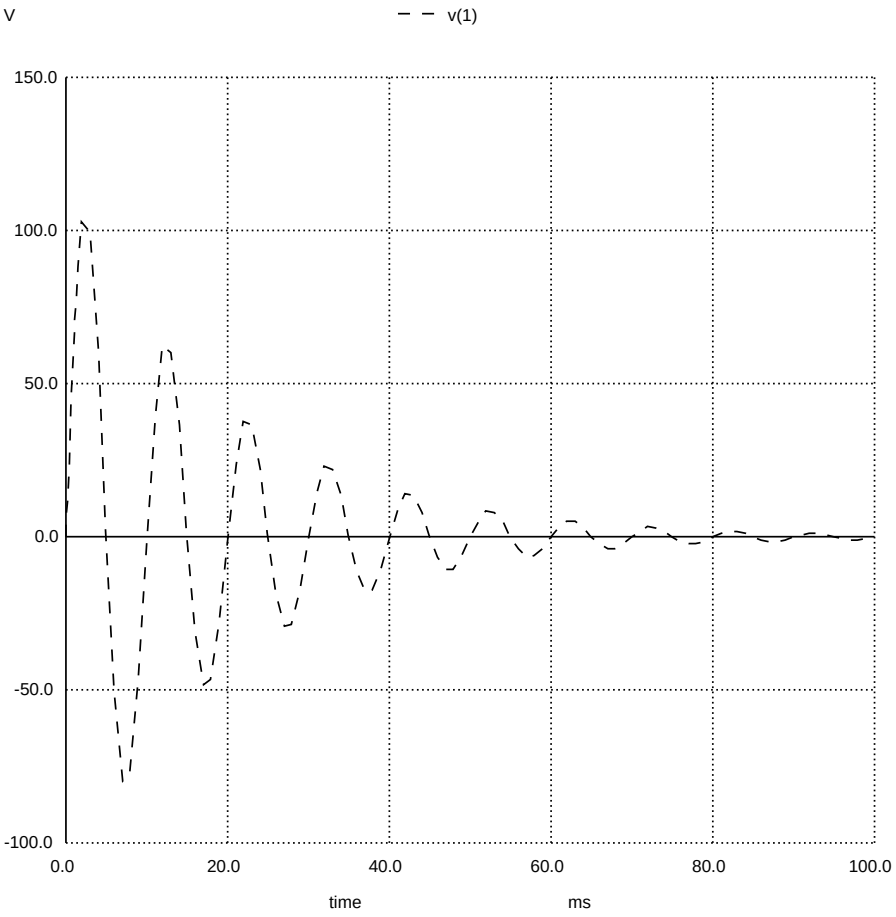
- What are some good ways to manage all of the calculations necessary to analyze this filter network?

- How does this filter respond to the DC that is the "zeroth" harmonic of the power system?

6.2.8 Determining s from oscillographs

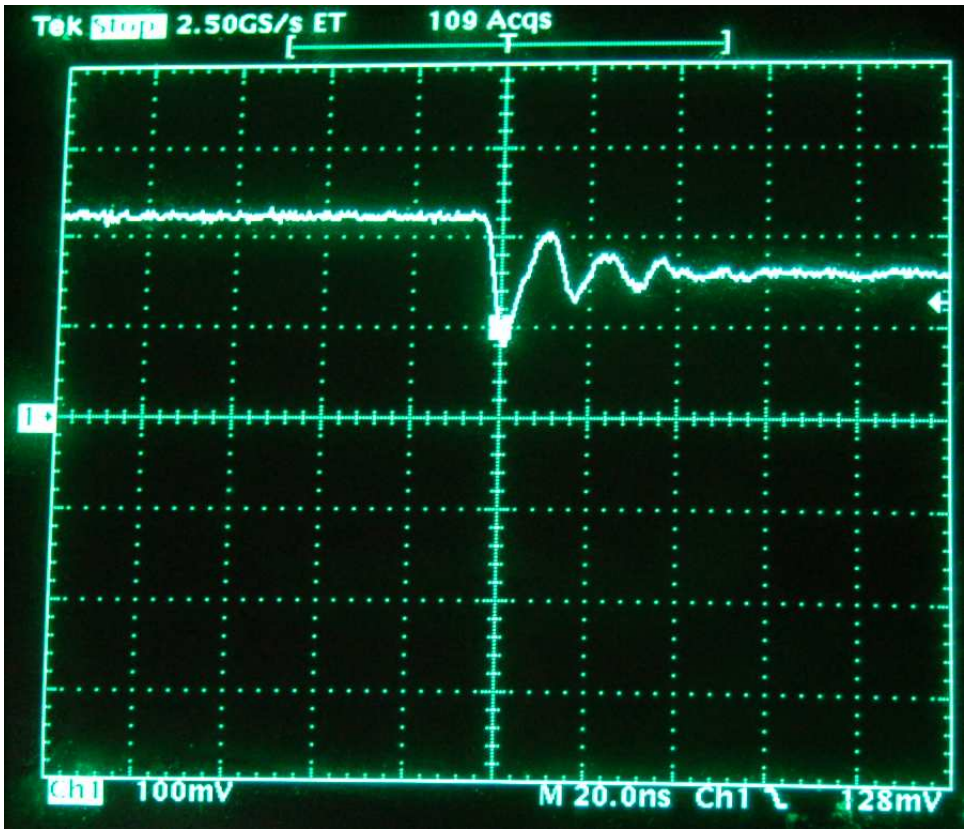
Example #1:

Determine the complex frequency of the following AC voltage signal:



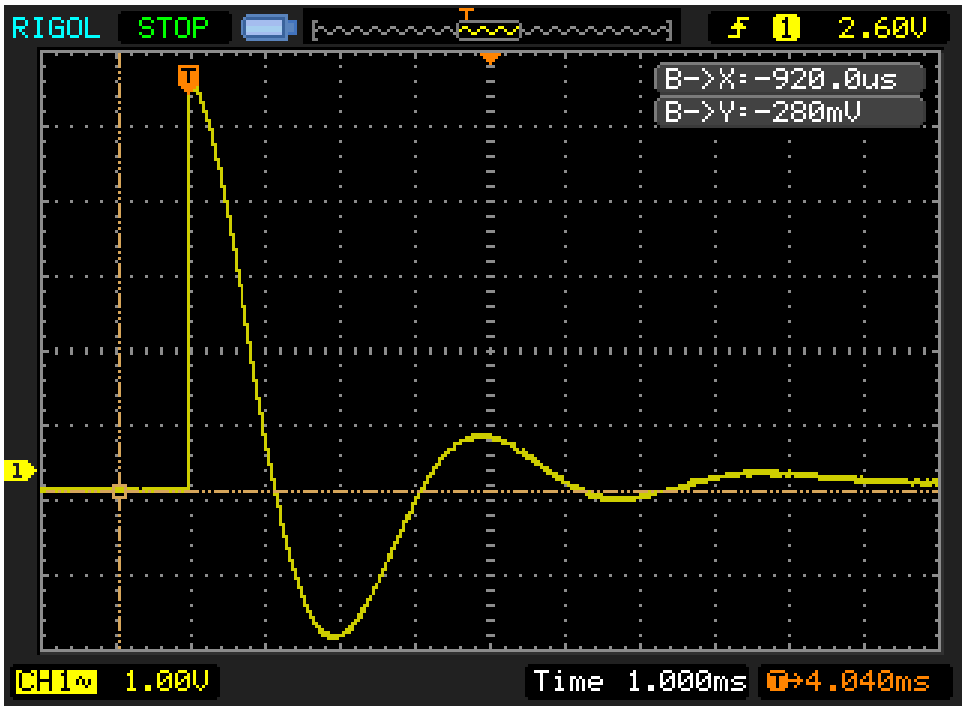
Example #2:

Determine the complex frequency of the "ringing" observed in this AC voltage signal:



Example #3:

Determine the complex frequency of the following AC voltage signal:



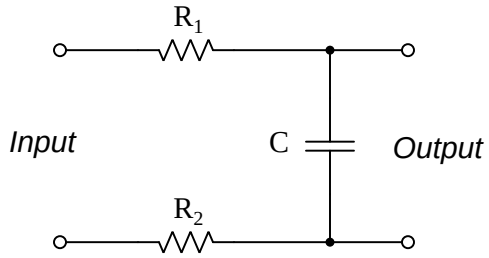
Challenges

- How could we prove that these envelope profiles are actually examples of *exponential* decay rather than some other decay profile?

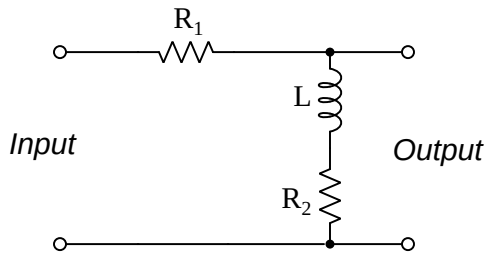
6.2.9 Transfer functions for simple networks

Write the *s*-domain transfer functions for the following networks, eliminating all compound fractions to ensure they are as simple in form as possible. Also identify any poles and zeroes for the transfer functions you write:

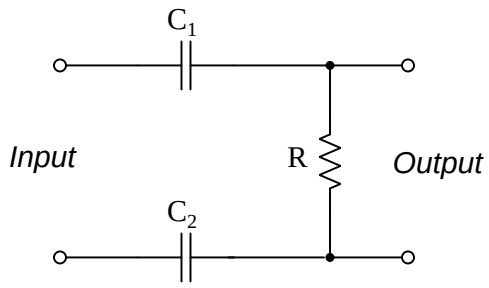
Example #1:



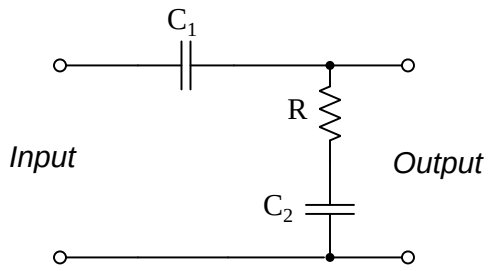
Example #2:



Example #3:



Example #4:

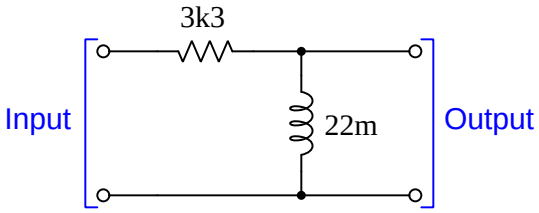


Challenges

- Explain what these poles and zeroes *mean* in practical terms.

6.2.10 Network response to different frequencies

Calculate the value of the transfer function corresponding to the following network at a complex frequency value of $s = -24\text{k} + j100\text{k}$ per second:



What unit(s) of measurement will this transfer function's value have, and why?

Also, explain what this transfer function value means in practical terms.

Now, calculate the transfer function values of the same network at the following frequencies:

- $s = 594 + j90\text{k}$ per second
- Frequency of 30 kHz at a constant magnitude
- DC signal decaying at an exponential rate of -1200 time constants per second

Challenges

- What would happen if we subjected this network to a source having a complex frequency equal to the network's zero value?
- What would happen if we subjected this network to a source having a complex frequency equal to the network's pole value?

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

Boylestad, Robert L., *Introductory Circuit Analysis*, 9th Edition, Prentice Hall, Upper Saddle River, NJ, 2000.

Brandi, Eduardo; Abboud, Ricardo; Calero, Fernando; "Protecting Harmonic Filters in a ±600 kV HVDC Installation", presentation at the 42nd Annual Western Protective Relay Conference, Spokane, WA, 20-22 October 2015.

Harrison, Cecil A., *Transform Methods in Circuit Analysis*, Saunders College Publishing, Philadelphia, PA, 1990.

Kaplan, Wilfred, *Advanced Mathematics for Engineers*, Addison-Wesley Publishing Company, Reading, MA, 1981.

Nilsson, James W., *Electric Circuits*, Addison-Wesley Publishing Company, Reading, MA, 1983.

Palm, William J., *Control Systems Engineering*, John Wiley & Sons, Inc., New York, NY, 1986.

Smith, Steven W., *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, CA, 1997.

Waltman, Gene L., *Black Magic and Gremlins: Analog Flight Simulations at NASA's Flight Research Center*, Monographs in Aerospace History Number 20, NASA History Division, Office of Policy and Plans, Washington DC, 2000.

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.

11-22 March 2026 – minor edits to the Tutorial, as well as to the "Example: poles and zeroes of a series-parallel RC network" section of the Case Tutorial. Also added a new Conceptual Reasoning question.

10 December 2025 – added Sallen-Key low-pass and high-pass filter networks to the Case Tutorial section showing "Gallery of pole-zero analyses for opamp networks", also taking the word "simple" out of the title for this section.

12-15 October 2025 – added some instructor notes, as well as another oscillograph to the "Determining s from oscillographs" Quantitative Reasoning question. Minor edits made to the "Pole plots for pendula" Conceptual Reasoning question.

8 October 2025 – minor updates to the Tutorial.

25 May 2025 – added a new Derivations section on negative frequency.

13 March 2025 – typographical error correction courtesy of Jacob Stormes – "locates shows" was grammatically incorrect.

12 March 2025 – made minor clarifying edits to the Tutorial, and also added a new Programming References section demonstrating a C++ program that plots time-domain graphs of various s complex-frequency values.

11 March 2025 – corrected a typographical error in the Tutorial ("to the original" should have been "to the origin"). Also added a question called "Pole plots for a bell" to the Conceptual Reasoning section. Also added comments to the LC tank circuit example in the Tutorial discussing conceptual implications of pole placement.

10 March 2025 – edited some instructor notes in the Questions chapter, as well as added some footnotes in the Tutorial chapter. Also added a new Case Tutorial section showing examples of series-parallel RLC network impedance calculations using complex frequency.

14-16 September 2024 – minor edits to the Tutorial, and added more instructor notes. One of those instructor-note additions was a program written in C by Gavin Koppel to calculate network impedance for the "RC network impedances" Quantitative Reasoning question.

30 September 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 some concepts to the Introduction chapter.

30 March 2024 – added another example to the "Examples: gallery of pole-zero analyses for simple R, L, C networks" Case Tutorial section, and corrected a couple of typographical errors.

19 March 2024 – corrected a typographical error where I described an over-damped time-domain plot as being under-damped, courtesy of Joe Archer. Also simplified an instructor-note answer courtesy of Daniel Wing.

17-18 March 2024 – added a new Quantitative Reasoning question challenging the reader to analyze a HVDC power system harmonic filter. Also, made a minor edit to the "Transfer function summary" section of the Tutorial, where I changed "Multi-order systems are capable of self-oscillation" to "Only multi-order systems are capable of self-oscillation". Also edited titles of two Qualitative Reasoning questions to mention only poles rather than poles and zeroes.

10-13 March 2024 – corrected some errors in the instructor notes as well as more explicitly described how the "Ohm's Law" equations for inductance and capacitance relate to the phase-shifted voltage and current oscillographs shown in that section. Also corrected a grammatical error where the text said "to perform to apply". Also made all the expressions of radians per second consistent, writing them as *rad/s* rather than *rad/sec*. Other minor edits made to the Tutorial, as well as additions to instructor notes. Also added new Summary section to the Tutorial chapter, re-naming the old one "Transfer function summary" and the new one "s variable summary".

10-19 February 2024 – added Case Tutorial examples of pole-zero analysis for networks, and made other minor edits to the Tutorial text. Also deleted some unnecessary code within the "Pole-zero plotter using C++" Programming References section.

30 December 2023 – added many entries to the Foundational Concepts subsection of the "Conceptual Reasoning" section of the Questions chapter, as well as many entries to the Introduction chapter.

27 December 2023 – added another Programming Reference section showing a pole-zero plotting program written in C++.

21-23 December 2023 – major additions and edits to the Tutorial chapter, to section titles, and also to questions. Also re-named the Full Tutorial as simply "Tutorial" because there is no Simplified

Tutorial to date, and altered some of the section titles in addition to adding new sections. Also added some questions to the module as student exercises as well as a Case Tutorial chapter.

24 October 2021 – minor edit in the RC high-pass filter example section of the Full Tutorial, expanding s to be $\sigma + j\omega$ in the introductory transfer function equation.

21 January 2020 – added a new section to the beginning of the Full Tutorial.

July 2018 – document first created.

Index

Q (quality factor of RLC filter circuit), 75
Q, filter, 75
s variable, 46, 76
"Ohm's Law" for a capacitor, 34, 37
"Ohm's Law" for an inductor, 34, 37

Adding quantities to a qualitative problem, 150
Angular frequency, 40
Angular velocity, 37, 46, 81, 84
Annotating diagrams, 149
Anti-resonance resistor, 69

Bode plot, 54

C++, 92
Checking for exceptions, 150
Checking your work, 150
Code, computer, 159
Compiler, C++, 92
Complex frequency, 40, 82, 85
Complex number, 35
Computer programming, 91
Critical damping, 22, 25, 68

Damping, anti-resonance resistor, 69
Damping, automobile suspension, 69
Damping, critical, 22, 25, 68
Damping, over-, 22, 25, 68
Damping, under-, 22, 25, 68
Dimensional analysis, 149
Discriminant, 22, 67, 88

Edwards, Tim, 160
ePiX C++ library, 55
Euler's Relation, 3, 39, 81, 84, 86, 135

Frequency, 81, 84
Frequency, angular, 40
Frequency, complex, 40, 82, 85
Frequency, natural, 37
Friction, 36
Functions, transfer, 48

GNU Radio Companion software, 89
Graph values to solve a problem, 150
Greenleaf, Cynthia, 117
Growth/decay rate, 39

How to teach with these modules, 157
Hwang, Andrew D., 161

Identify given data, 149
Identify relevant principles, 149
Imaginary, 36
Impedance, 35, 48
Inertia, 36
Intermediate results, 149
Interpreter, Python, 96
Inverted instruction, 157

Java, 93

Kirchhoff's Voltage Law, 52
Knuth, Donald, 160
KVL, 52

Lamport, Leslie, 160
Laplace transform, 48, 55
Limiting cases, 50, 150
Linear time-invariant (LTI) system, 76
LTI system, 76

Metacognition, 122
Moolenaar, Bram, 159
Murphy, Lynn, 117

Natural frequency, 37

Neper velocity, 40, 46

Ohm's Law, 34, 37, 80, 83
Open-source, 159
Over-damping, 22, 25, 68

Pass band, filter, 48
Phasor diagram, 36
Pole, transfer function, 13, 48, 51, 57
Pole-zero plot, 53
Polynomial expression, 67
Problem-solving: annotate diagrams, 149
Problem-solving: check for exceptions, 150
Problem-solving: checking work, 150
Problem-solving: dimensional analysis, 149
Problem-solving: graph values, 150
Problem-solving: identify given data, 149
Problem-solving: identify relevant principles, 149
Problem-solving: interpret intermediate results, 149
Problem-solving: limiting cases, 50, 150
Problem-solving: qualitative to quantitative, 150
Problem-solving: quantitative to qualitative, 150
Problem-solving: reductio ad absurdum, 150
Problem-solving: simplify the system, 149
Problem-solving: thought experiment, 149
Problem-solving: track units of measurement, 149
Problem-solving: visually represent the system, 149
Problem-solving: work in reverse, 150
Programming, computer, 91
Python, 96

Quadratic formula, 67, 70, 72
Qualitatively approaching a quantitative problem, 150
Quality factor, band-pass filter, 75
Quality factor, RLC filter circuit, 75

Rate, growth/decay, 39
Reactance, 35
Reading Apprenticeship, 117
Real, 36
Reductio ad absurdum, 150, 156, 157
Resistance, 35
Resistor, anti-resonance, 69
Resonance, 61
Root, polynomial, 67

Schoenbach, Ruth, 117
Scientific method, 122
Simplifying a system, 149
Socrates, 156
Socratic dialogue, 157
Source code, 92
SPICE, 117
Stallman, Richard, 159
Stop band, filter, 48
Superconductor, 80

Tank circuit, 61
Thought experiment, 149
Time constant, 39, 52, 58
Torvalds, Linus, 159
Transfer function, 76
Transfer function, pole, 13, 48, 51, 57
Transfer function, zero, 13, 48, 51, 57, 62
Transfer functions, 48
Transform, Laplace, 48, 55

Under-damping, 22, 25, 68
Units of measurement, 149

Velocity, angular, 37, 46
Velocity, Neper, 40, 46
Visualizing a system, 149
Voltage divider formula, 50

Whitespace, C++, 92, 93
Whitespace, Python, 99
Work in reverse to solve a problem, 150
WYSIWYG, 159, 160

Zero, transfer function, 13, 48, 51, 57, 62