RC Circuits and Filtering

Matlab Case Study for Signals and Systems (Draft)

# Introduction

In this course (and possibly others), you have learned that signals can be expressed as a combination of different component frequencies. Being able to examine which frequencies are present in a signal and at what strengths is essential for working with nearly any system. Often when working with a signal, we will want to control which frequencies are kept and which are attenuated. (For a practical example, maybe we want to cut the thumping, low-frequency bass on our stereo to avoid annoying our next door neighbors!)

Resistor-Capacitor circuits are common circuits used for analog signal filtering and can be used to construct filters that will reject some frequencies of signals while allowing others through mostly unattenuated. In this lab, we will explore the effect of these types of circuits on various signals, and apply our findings by filtering out unwanted parts of a signal.

By the end of this case study, you will have a firmer understanding of how

# Objectives

In this case study, you will:

1. Simulate the response of three different RC circuits to a variety of inputs.
2. Relate the results of the simulation to your own practical understanding of electronic circuits.
3. Explore properties of linearity and superposition.
4. Study the relationship between input frequency and amplitude gain for each circuit
5. Extend this process to various audio files to connect your conceptual understanding of frequency analysis to your everyday experiences with sound
6. Design an RC circuit to remove unwanted frequencies from a signal.

# Transfer Functions

A transfer function can be thought of as a way of representing a system or process in terms of the effect it has on different frequencies of input. It is a complex function, meaning it can take in a complex number *s = σ+jω* and output another complex number (When considering the transfer function of an RC circuit, you can think of *s* as an AC signal. For a pure AC signal with no phase offset, *σ=0* and the value *ω* represents the AC frequency).

The magnitude of the transfer function output for a particular frequency is called the “gain,” and expresses how the amplitude of that frequency is changed as it passes through the system. Similarly, the phase of the output expresses how the phase of the frequency is changed as it passes through the system.

For instance, imagine we have an RC circuit with the following transfer function:

Say we are interested in knowing what would happen if we applied a 100 rad/s sine wave to the input of this circuit. We can model this input as the complex number *s = σ+jω* = *0 + 100j*. The gain of the transfer function for this input is:

The phase of the transfer function for this input is:

This means that if we put a 100 rad/s sine wave through the RC circuit, the output will be a 100 rad/s wave with an amplitude that has been reduced by a factor of . In addition, the output will lag behind the input by , or 1/8th of a period.

A close up of a mans face

Description automatically generated

# Three Passive RC Filters

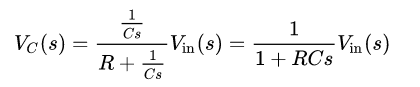
In this case study we will consider three different circuits. Each circuit has different properties that will affect the signals that pass through it.

## Circuit 1:

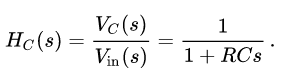
A picture containing clock

Description automatically generated

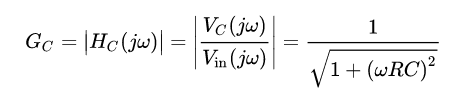
This circuit measures the voltage across the capacitor VC as a function of the voltage across the input Vin. You may notice that it resembles a voltage divider, with one resistor swapped for a capacitor. Using the concept of complex impedance, we can represent the resistance of the capacitor as and write the following expression:



Giving us the transfer function:



Remember that *s* is a complex number  *s = σ+jω,* where *ω* is frequency. Assume no DC offset (*s = jω*). Taking the magnitude of this transfer function:



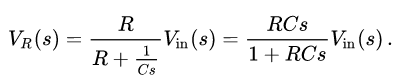
This equation models the gain of the circuit – the ratio between the amplitudes of the output and input. Consider how the gain changes as the frequency becomes larger or smaller. Is this a “high-pass” filter, allowing higher frequencies through while attenuating lower ones, or a “low-pass” filter which allows lower frequencies through?

## Circuit 2

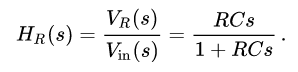
A picture containing object, clock

Description automatically generated

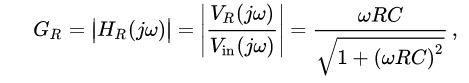
This circuit is similar to the previous one, but the voltage divider measures the voltage across the resistor instead.



This gives us the transfer function:



Which has magnitude:



Consider how the gain changes as the frequency becomes larger or smaller. Is this a “high-pass” filter or a low pass filter?

## Circuit 3

A close up of a antenna

Description automatically generated

This third circuit is effectively a cascaded system: the output of circuit 1 has been attached to the input of circuit 2. The transfer function for this circuit is the product of the previous two:

You will explore the properties of this transfer function during the case study.

# Case Study

Open the RCcasestudyscript.m MATLAB script. This script contains code to simulate three different RC circuits using the lsim() function.

* **Impulse Response and Simulation**
  + Read through the MATLAB script. Using the impulse responses generated for each circuit and the convolution function conv, write some code that will simulate and plot each circuit’s response to a given input signal.
    - Consider using the linkaxis command to plot each circuit’s output separately to avoid overplotting, while keeping the axis dimensions the same for easy comparison.
* **Transient and Steady State Response**
  + Examine the response of each circuit to a step or square wave input signal. Make sure to note the behavior of the circuit both immediately after the step and a short time after.
  + Explain in your own words what is physically happening in each circuit to cause the results you observe.
* **Exponentials and Impulse Response**
  + Compare the responses of each circuit to a decaying exponential. Vary the time constant and comment on how the responses for each circuit change. (τ = 10, 100, and 1000 should be sufficient to make some observations).
  + Between circuits 1 and 2, which is able to keep up with faster changes in input? Which one attenuates slower changes in input?
* **High and Low Pass Filters**
  + Determine what happens to high frequency signals as they pass through each circuit (≈1 kHz). What happens to low frequency signals (≈10 Hz)? What happens to signals that are superpositions of both high and low frequency signals? Are these systems linear? Make some qualitative observations.
  + Which of these circuits is a “high-pass” filter? Which is a low-pass filter? Which is a bandpass filter?
* **Linearity and Complex Exponentials**
  + Pass the complex exponentials and through a circuit (for any frequency *f*) and add the two output complex exponentials together. Plot the result. What do you notice? Explain your results using Euler’s formula and the concept of linearity.
    - Don’t worry about interpreting what it physically means to pass a complex exponential through a real circuit. Just focus on the math.
  + What is the phase of each complex exponential before passing through the circuit? What is the phase of each afterwards? How does this help explain your previous observations?
* **Magnitude Plot**
  + Send a pure sine wave through each of the circuits and estimate the gain in decibels. Do this for several frequencies between 1 Hz and 10kHz. Make a plot with the gain along the Y-axis and the frequency along the X-axis. What patterns do you see for each circuit?
    - You will likely want three or more samples per frequency “decade” – that is, three samples between 10 and 100 Hz, three between 100 and 1000 Hz, etc.
    - The gain in decibels is equal to . It should be zero or less for all frequencies.
    - You may want to automate this process using a for loop.
  + As an optional extension, complete the same steps, this time measuring the phase shift between the input and the output. To automate this process, you’ll need to find a quantitative way to determine the phase shift.
  + These plots of the gain and phase change of a system for a given frequency are called Bode plots.
* **Frequency Sweep**
  + The chirp\_timeseries block contains a sound clip that starts at 1 Hz and increases logarithmically to 10 kHz over the course of three seconds. Put it through each of the circuits. Relate your results to the magnitude plot you created in the previous section.
  + Use the plotPowerSpectrum() function included in the case study folder to examine the frequencies present in the signal before and after it passes through each circuit. Relate your results to the magnitude plot you created in the previous section.
* **Cutoff Frequency**
  + Circuit 1 and Circuit 2 both have a “cutoff frequency” of 1/(2πRC) Hz. Compare the output of both circuits using this frequency as an input and note any observations. Examine the magnitude plots for both circuits. What is the significance of the “cutoff frequency?”
    - For both circuits:
      * R = 1kΩ
      * C = 2µF
  + Circuit 3 is composed of two separate RC circuits with different cutoff frequencies. Calculate both cutoff frequencies and find those frequencies on the magnitude plot for circuit 3. Note your observations.
    - For circuit 3:
      * R1 = 320Ω
      * R2 = 400Ω
      * C1 = 0.2µF
      * C2 = 2µF
* **Custom Audio Samples**
  + Make a brief (1-5 seconds) recording of your own and load it into the simulation using the chirp timeseries as an example. Use the sound() function in MATLAB to play the sound back before and after putting it through circuit 3. Describe qualitatively the difference before and after passing through the circuit. If you don’t hear a difference, try adding some low and high frequency noise to the sample.
  + Plot the power spectrum of your sample before and after passing through the circuit using the plotPowerSpectrum function included in the case study folder. Record your observations in your writeup.
  + You may need to change the simulation duration T in the init.m script to fit your entire sample.
  + For best results use a mono track sampled at 44100 Hz or change the sample rate in the simulation to match.
* **Design an RC Circuit**
  + The blurry\_audio block contains a soundclip that includes some noise. Examine the power spectrum of this audio using the plotPowerSpectrum function included in the case study folder. Devise a new RC circuit consisting of a chain of high-pass and low-pass filters that will “clean up” the audio. Use your best judgement on which signals to keep and which to discard.
  + You can use the code provided in the case study as a template for generating the impulse function of the low and high pass filters.
  + To place two circuits in series or in parallel, you will need to determine the relationship between the impulse functions of the component circuits and the impulse functions of the whole circuit.

# What To Turn In

* Present your results as a writeup in IEEE form that includes:
  + Your observations for each section of the case study
  + Any useful plots you have created.
  + The power spectral density of your audio signal before and after passing it through circuit 3
* Include your modified RCcasestudyscript.m script and any custom functions you created.
* Week 1
  + Sampling and aliasing
  + Discrete vs continuous
  + Transient Response
    - Transient response to both step input and sinusoid
  + Response to sine waves (high frequency vs. low frequency)
  + Play with audio
* Week 2
  + Impulse response & convolution
    - Convolve impulse and step to see transient response from last week
  + Complex exponentials and linearity/superposition
  + Magnitude/phase plot
    - Compare linear and log plot/introduce to dB?
  + Frequency sweep/chirp
* Integrative
  + Cutoff frequencies
  + Cascaded and parallel systems and their impulse responses?
  + DC to AC converter
    - Generate 60 Hz sine wave from square wave
    - Design considerations:
      * Frequency of input square wave
      * How many cascaded filters to use
      * High pass or low pass
  + Audio bandpass filter