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 their effects on various frequency inputs, and then apply those observations to some audio recordings.

By the end of this case study, you will have a firmer understanding of the relationship between the time and frequency domain, the relationship between transfer functions and Bode plots, and the relation between sound and frequency.

# Objectives

In this case study, you will:

1. Familiarize yourself with the Simulink software included with MATLAB
2. Simulate the response of three different RC circuits to various sinusoidal inputs in both MATLAB and Simulink
3. Study the relationship between input frequency and amplitude gain for each circuit
4. Examine Bode plots of these circuits and relate them to the effect each circuit has on different input frequencies.
5. Examine power spectral density plots of the input and output of these circuits and relate them to the Bode plots of the circuits.
6. Extend this process to various audio files to connect your conceptual understanding of frequency analysis to your everyday experiences with sound

# Simulink

Simulink is a piece of software packaged with your MATLAB installation that allows you to create simulated environments out of a wide variety of components, including transfer functions, block diagrams, waveform generators, and more. The folder for this case study includes a pre-made model, *RCcasestudy.slx*, which has some basic instructions for how to navigate the Simulink environment.

Simulink models are composed of blocks, each of which have some combination of input and output ports. By connecting the ports together, we can pass different signals through the model and examine the results.

# 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

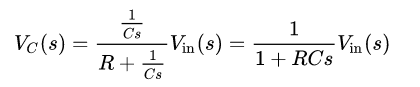
In this case study we will consider three different circuits. Each circuit has different properties that will affect the signals that pass through it.

## Circuit 1:

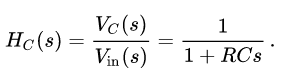
A picture containing clock

Description automatically generated

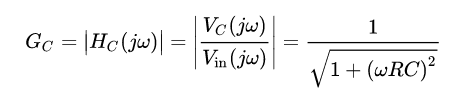
This circuit measures the voltage across the capacitor VC as a function of the voltage across the input Vin. You may notice that it resembles a voltage divider, with one resistor swapped for a capacitor. Using the concept of complex impedance, we can represent the resistance of the capacitor as and write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



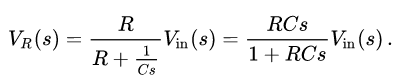
This equation models the gain of the circuit – the ratio between the amplitudes of the output and input. Consider how the gain changes as the frequency becomes larger or smaller. Is this a “high-pass” filter, allowing higher frequencies through while attenuating lower ones, or a “low-pass” filter which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

## Circuit 2

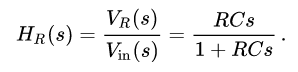
A picture containing object, clock

Description automatically generated

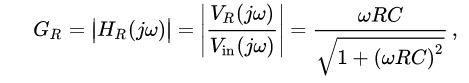
This circuit is similar to the previous one, but the voltage divider measures the voltage across the resistor instead.



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

Navigate to the Circuits Case Study folder in MATLAB and open the RCcasestudy.slx Simulink model. (If you are not in the correct folder, some aspects of the model may not load. Try running init.m in the Circuits Case Study folder and then running the simulation to fix this.).

Additionally, open the RCcasestudyscript.m MATLAB script. This script contains much of the same functionality as the MATLAB model. One aspect of this case study will be exploring both implementations of the same model.

* **Introduction** 
  + Examine the Simulink model and read the annotations for each part of the model. Take some time to play with the model by connecting different input signals to the three RC circuits and observing the output.
  + Similarly, read through the MATLAB script and make sure you understand how it works.
* **Qualitative Observations**
  + Using either model, 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? Make some qualitative observations.
  + Which of these circuits is a “high-pass” filter? Which is a low-pass filter? Which is a bandpass filter?
* **Quantitative Observations** 
  + Using either model, 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 can edit an existing sine wave’s frequency by double-clicking it, or create a new one from the Sources folder in Simulink’s Library Browser. Frequencies are expressed in radians per second, so make sure to convert accordingly.
    - Higher frequency signals can slow down the simulation. This is because when high frequency signals are present, Simulink decreases the step size of its solver to prevent undersampling. Make sure to delete or change high frequency sources once you’re done with them to keep the simulation running quickly!
* **Bode Plots**
  + Follow the instructions near the top of the model to uncomment and run the Bode Plots. Compare the resulting plots with the observations you made in the previous section.
* **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 and plot power spectrums of the chirp signal before and after passing through circuit 3. Relate your results to the Bode plots from the previous section.
* **Test Linearity**
  + On either model, add together two component frequencies and put them through each circuit. Then put the two component frequencies through separately and add the outputs together. Compare the results and comment on their significance.
* **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 bode 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 Bode 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 init.m script 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.
* **Design a Cutoff Frequency**
  + 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 new values for the resistors and capacitors of circuits 3 that will “clean up” the audio. These values can be edited in the init.m script. Use your best judgement on which signals to keep and which to discard.

# 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.
  + Screenshots of simulation output when necessary to aid presentation
  + 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.