RC Circuits and Filtering

Matlab Case Study for Signals and Systems (Draft)

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 case study, you will use the Simulink software included with MATLAB to simulate the responses of several simulated RC circuits to various sinusoidal inputs. You will then use what you’ve learned to put a custom sound sample through the filter and observe the results.

# 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.

# Impedance, Phasers, and AC Circuits

Not sure how in-depth to go here until I know where this will fit into the curriculum and what I can expect students to know going into this case study, but this section will basically be a TL:DR of impedance, not going super into the theory in favor of just making sure that students understand it’s just what happens when you express “resistance” in the complex plane.

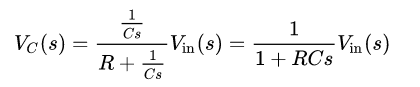
# Three Passive RC Filters

## Circuit 1:

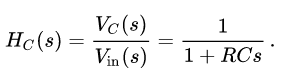
A picture containing clock

Description automatically generated

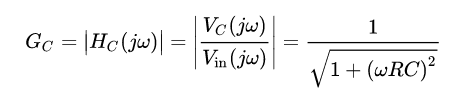
This circuit measures the voltage across the capacitor as a function of the voltage across both elements. You may notice that it resembles a voltage divider, with one resistor swapped for a capacitor. We can combine what we know about the properties of voltage dividers with what we know about the properties of impedance to derive the voltage across the capacitor as a function of the voltage across the entire circuit.



Giving us the transfer function:



Remember that *s* is a complex number  *s = σ+jω,* where *ω* is frequency. Taking the magnitude of this transfer function:



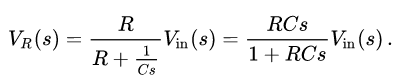
This equation models the gain of the circuit – the ratio between the amplitudes of the output and input. What happens to the gain as frequency increases? What happens when it is very large?

## Circuit 2

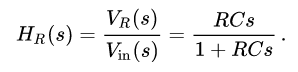
A picture containing object, clock

Description automatically generated

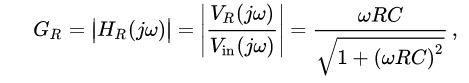
This circuit is similar to the previous one, but the voltage divider measures the voltage across the resistor instead.



This gives us the transfer function:



Which has magnitude:



What happens to the gain as frequency decreases? What happens when it is very small?

## 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 not given here, as you will explore its properties 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 to fix this). Complete the following tasks:

* Examine circuit 3. Based on your knowledge of the previous circuits, and of the properties of cascaded systems and transfer functions, determine the transfer function of this new system. Predict how this circuit will respond to different frequencies. Record your observations in your writeup.
* Experiment with the Simulink model by connecting different input signals to the three RC circuits. Try using superpositions of both high and low frequency signals.
* By experimentation, determine what happens to the magnitude phase of different signals as they are put through the filter. Record your observations in your writeup.
* Check your results by examining the bode plots accompanying each circuit.
* 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 RC filters and comment on the results.
* Examine the values of R, C, R1, C1, R2, and C2 given in init.m. What are the cutoff frequencies of each filter?
* Make a brief recording of your own and load it into the simulation, using chirp\_timeseries as an example. Use the sound() function in matlab to play the sound back before and after putting it through circuit 3. If you don’t hear a difference, try adding some low and high frequency noise to the sample. Plot the power spectral density of your sample. Record your observations in your writeup.
  + The output of circuit 3 is saved to the workspace as the variable “out.simout.data.”