EE16B
May 9, 2019
Jordan Chernof

# MYOP Homework 4

## Bode Plotter

This problem will walk you through creating a basic low pass or high pass filter bode plot in iPython.

(a) **Follow the instructions in the iPython notebook to create a lambda function *getOmegap* that takes in a resistance R and a capacitance C and returns the pole frequency (omegap) of the RC tandem.**

See iPython solutions notebook for solution.

(b) Now we will build the *getImpedance* function. Given a circuit component, we know the impedance may be a function of omega. Therefore, we might not know a component's impedance without knowing the frequency. **Fill in the blanks in the skeleton code for the getImpedance function, where the outer function calls a higher order function that calculates the impedance of resistors or capacitors.**

See iPython solutions notebook for solution.

(c) **Fill in the code creating an impedance divider function.** Note this function uses the boolean *RC_circuit* to decide if the numerator is a resistor or capcaitor.

See iPython solutions notebook for solution. *Hint: This function makes several calls to the getImpedance function!*

See iPython solutions notebook for solution.

(d) Fill in the *getMag* function, which takes in a transfer function and a frequency and outputs the magnitude of the transfer function at the specified frequency.

*Hint: Use np.linalg.norm and the given num and den variables!*

See iPython solutions notebook for solution.

(e) Fill in the *getPhase* function, which takes in a transfer function and a frequency and outputs the phase of the transfer function at the specified frequency.

*Hint: Use np.angle to find the arctangent of a complex number!*

**Now you have successfully (hopefully) created a bode plotter!**

See iPython solutions notebook for solution.

(f) Try out some example circuits to confirm your understanding of bode plots and to check your code. Write down any comments you may have. Does your iPython plot match up with a hand-calculated plot?

Answers may vary. As long as the iPython plot matches a hand-calculated plot, the answer should be marked correct.