LAB 3 - TASK 2 (1/1 point)

In Task 1, you have estimated the effect of the amplitude response of the IR channel by visual inspection. In this task, we will perform a more systematic study of the amplitude response of the IR channel by considering more frequencies. We do this automatically in MATLAB by using a **for** loop to sweep through different frequencies, and writing code to measure the peak-to-peak amplitude of the received signal.

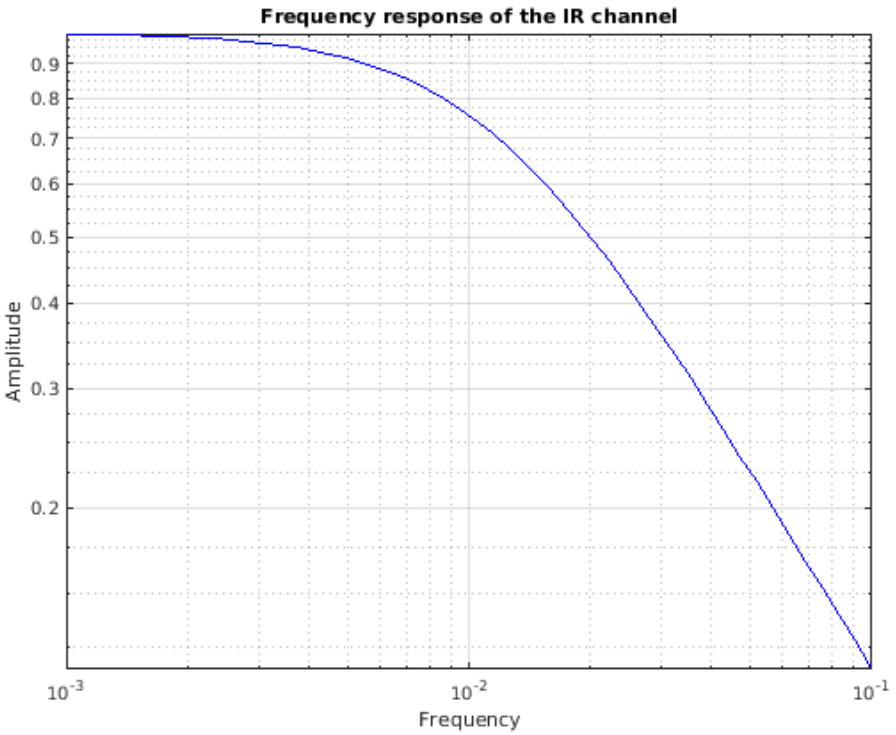
```
30      rx_peak2peak(i)= mx + mn;
31
32      % % % % Do not change the code below % % % %
33  end
34  figure(1);
35  % Estimate the amplitude response by dividing the peak-to-peak
36  % amplitude of the output by the peak-to-peak amplitude of the i
37  S = rx_peak2peak/2;
38  % Plot the amplitude response using logarithmic scaling on
39  % horizontal and vertical axes
40  loglog(flist,S,'b');
41  grid
42  xlabel('Frequency');
43  ylabel('Amplitude');
44  title('Frequency response of the IR channel');
45
```

Correct

Figure 1

5.5 Lab 3 - Frequency Response
Lab due Nov 16, 2015 at 15:30 UTC

- Topic 6: The Discrete Fourier Transform
- MATLAB download and tutorials
- MATLAB Sandbox



You have used 1 of 10 submissions

INSTRUCTIONS

The MATLAB code above should estimate the amplitude response of the IR channel by transmitting a set of sinusoidal waves with different frequencies through the channel and measuring the peak-to-peak values of the received signals. The amplitude response at each frequency is found by dividing the peak-to-peak amplitude of the received signal by the peak-to-peak amplitude of the input. However, this initial code is incorrect. Your task is to correct this code.

In the initial code, we first initialize the variable **flist**, which contains the list of normalized frequencies that we want to measure the amplitude response at. These frequency responses are logarithmically spaced. In other words, the base-10 logarithms of the frequencies in **flist** are equally spaced. The frequencies range from 10^{-3} and 10^{-1}. The normalized frequencies have units of cycles per sample, and range between 0 and 0.5. Discrete time cosines with normalized frequencies greater than 0.5 are aliased to look like lower frequency cosines.

The code then initializes the variable **rx_peak2peak**, which, at the end of the code, should contain the peak to peak amplitude of the received signals at different frequencies. Then, we start a **for** loop. In each iteration, the code should create a cosinusoidal input that has frequency **freq** picked from **flist** and is **nsamp** samples long. This input should then be sent through the channel. The received signal is stored as the variable **rx_wave**. The code should then compute the peak-to-peak amplitude of the output and store it inside the variable **rx_peak2peak**. Remember that we should skip past the first 200 samples in order to avoid artifacts due to the transient response. Finally, the code computes the amplitude response as the ratio of the peak-to-peak amplitudes of the output and input and plots this as a function of frequency. Since we use cosines with unit amplitude, the peak-to-peak amplitude of the input is two. Given the large ratio between the minimum and maximum values of the frequency and the amplitude response, the plot uses logarithmic scaling along the horizontal and vertical axes for visualization.

Your task is to correct the mistakes inside the for loop and store the peak to peak response of each frequency inside the variable **rx_peak2peak**. Please do not modify the variables **a**, **flist** and **nsamp**. When measuring the peak-to-peak amplitude, remember to ignore the transient response of the channel. For this lab, you can safely assume that the transient response ends before the first 200 samples.