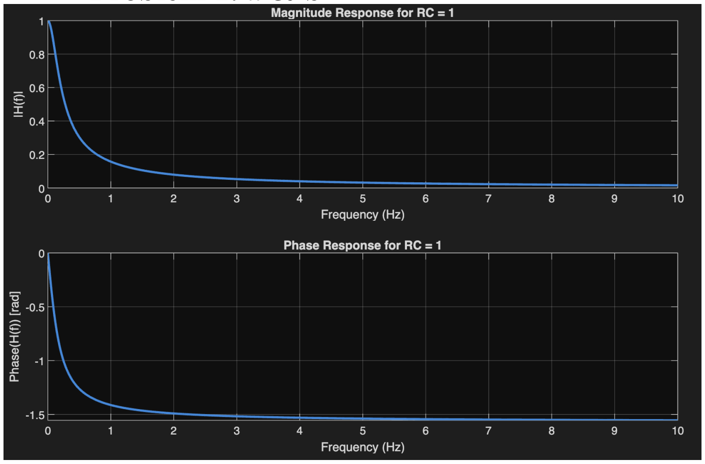
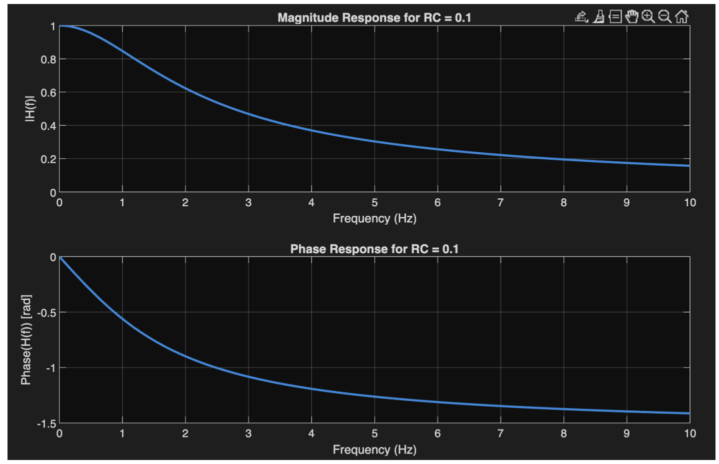
Luis Kligman Computer Assignment 2

Problem 1. a & b



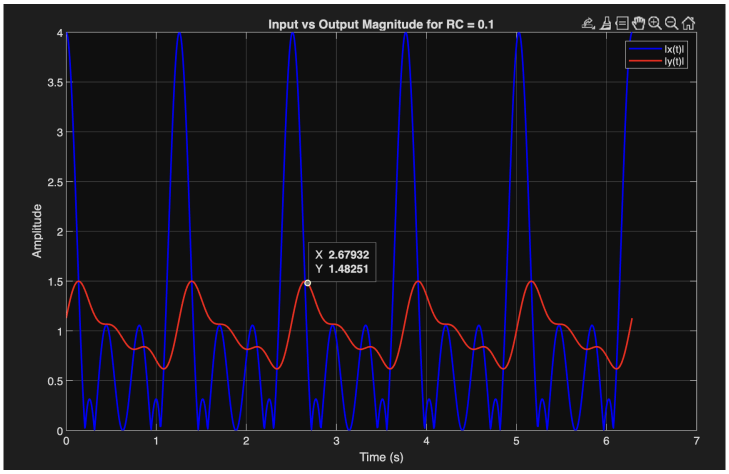


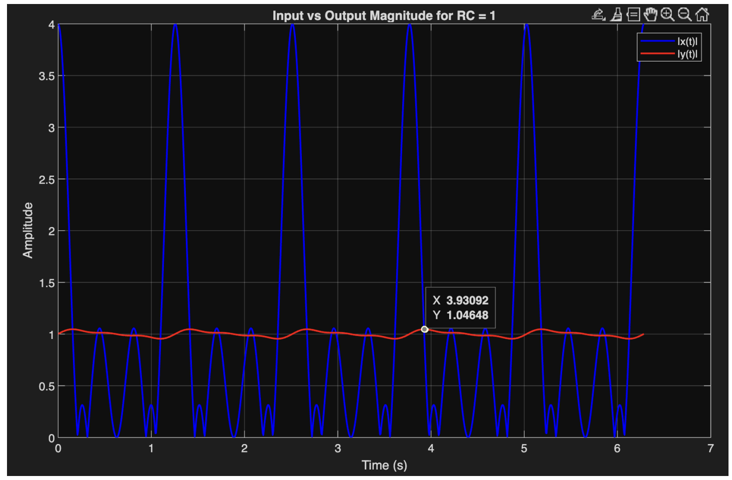
The frequency response of the RC circuit was analytically calculated using the formula H(f) = 1/(1+j2pifRC) for two cases: RC = 0.1 and RC = 1. This function is used to characterize how the circuit attenuates and phase-shifts input frequencies. Using MATLAB, we are able to evaluate the magnitude and phase of H(f) across the range 0 to 10 Hz and plot both. At 5 Hz and 10 Hz, the magnitude for RC = 0.1 was approximately .305 and .16, respectively, while for RC = 1, those values dropped significantly to around .032 and .016. The corresponding phase shifts at 10 Hz were approximately -1.4 radians for RC = 0.1 and -1.65 radians for RC = 1. The plots demonstrate that the larger RC value results in stronger attenuation and greater phase lag for higher frequencies.

Problem 1. c

For this part, we computed the output y(t) by scaling and phase-shifting each cosine term in the input x(t) using the frequency response values at 5 Hz, 10 Hz, and 15 Hz. This approach will properly model how the RC filter alters the amplitude and phase of each frequency component present in the signal.

Problem 1. d





We plotted the magnitudes of both the input and output signals over time for each RC value. The input |x(t)| had a maximum amplitude of 4 due to constructive interference of the cosine components. For RC = 0.1, the output |y(t)| had a reduced peak of approximately 1.5, while for RC = 1, the peak dropped to about 1.05. These results clearly show that the filter attenuates the signal more strongly as RC increases

Problem 1. e

The RC circuit functions as a low-pass filter, attenuating high-frequency components while allowing low-frequency content to pass. Our results show that as the RC value increases, the system suppresses more of the input signal. The output peak dropped from about 1.5 at RC = 0.1 to 1.05 at RC = 1, while also putting an end to the swings the output signal |y(t)| had. This demonstrates that larger time constant shifts the cutoff frequency lower, resulting in a smoother, more filtered signal with reduced high-frequency oscillations.

Appendix

```
% ELCT 222 - Computer Assignment 3
clear;
clc;
close all;
% Constants
RC_values = [0.1, 1];  % Case (i) and (ii)
f = linspace(0, 10, 1000);  % Frequency range (Hz)
t = linspace(0, 2*pi, 1000);  % Time vector for x(t) and y(t)
% Input signal
x = 1 + cos(5*t) + cos(10*t) + cos(15*t);
% Part 1a and 1b: Frequency response plots
for idx = 1:2
   RC = RC_values(idx);
   % Frequency response H(f)
   H_f = 1 ./ (1 + 1j*2*pi*f*RC);
```

```
    mag_H = abs(H_f);
    phase_H = angle(H_f);
    % Plot magnitude
    figure;
    subplot(2, 1, 1);
    plot(f, mag_H, 'LineWidth', 2);
    xlabel('Frequency (Hz)');
    ylabel('|H(f)|');
    title(['Magnitude Response for RC = ', num2str(RC)]);
    grid on;
    % Plot phase
    subplot(2,1,2);
    plot(f, phase_H, 'LineWidth', 2);
    xlabel('Frequency (Hz)');
    ylabel('Phase(H(f)) [rad]');
    title(['Phase Response for RC = ', num2str(RC)]);
    grid on;
end
% Part 1c: Compute output y(t) using frequency response at 0, 5, 10, 15 Hz
frequencies = [0, 5, 10, 15];  % Frequencies at which to compute output
for idx = 1:2
    RC = RC_values(idx);
    % Compute frequency response values
    H = 1 ./ (1 + 1j*2*pi*frequencies*RC);
    % Output signal y(t)
    y = real(H(1)) + ...
        abs(H(2))*cos(5*t + angle(H(2))) + ...
        abs(H(3))*cos(10*t + angle(H(3))) + ...
        abs(H(4))*cos(15*t + angle(H(4)));
    % Store for later plot
    if idx == 1
        y_RC_01 = y;
    else
        y_RC_1 = y;
    end
end
% Part 1d: Plot |x(t)| and |y(t)| for both RC values
for idx = 1:2
    RC = RC_values(idx);
    if idx == 1
        y = y_RC_01;
    else
        y = y_RC_1;
    end
    figure;
    plot(t, abs(x), 'b', 'LineWidth', 1.5);
    hold on;
    plot(t, abs(y), 'r', 'LineWidth', 1.5);
    xlabel('Time (s)');
```

```
    ylabel('Amplitude');
    legend('|x(t)|', '|y(t)|');
    title(['Input vs Output Magnitude for RC = ', num2str(RC)]);
    grid on;
end
```