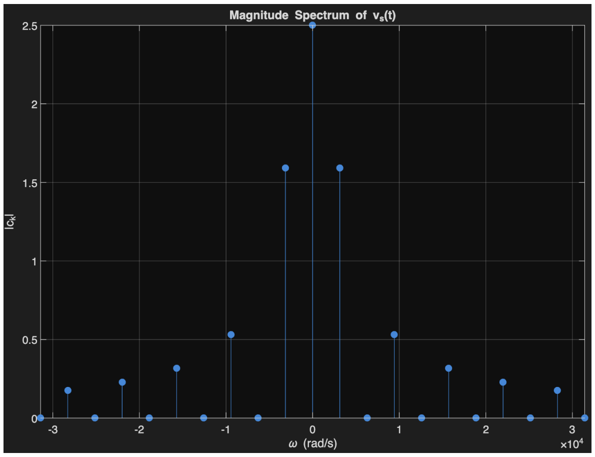
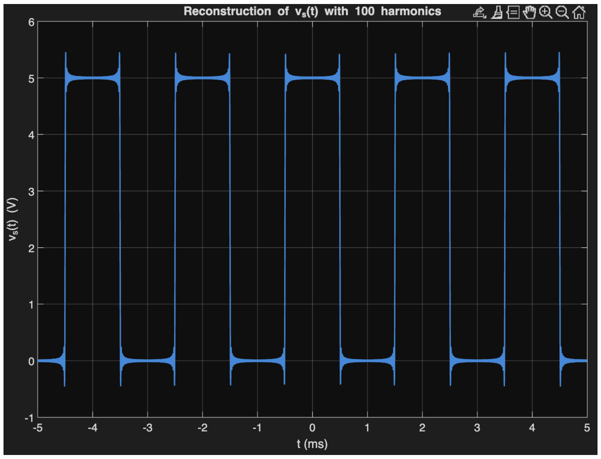
Luis Kligman Computer Assignment 8

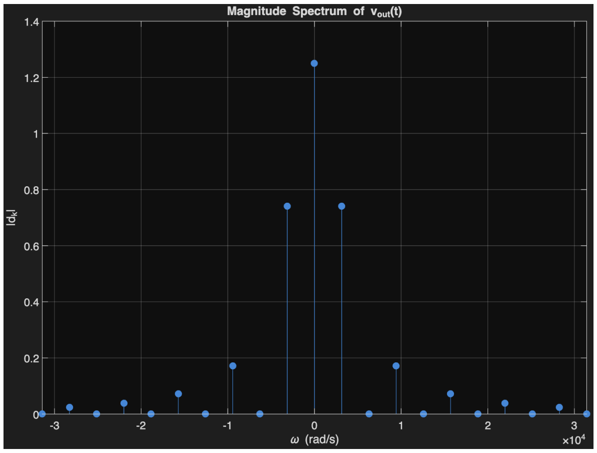
Problem 1:

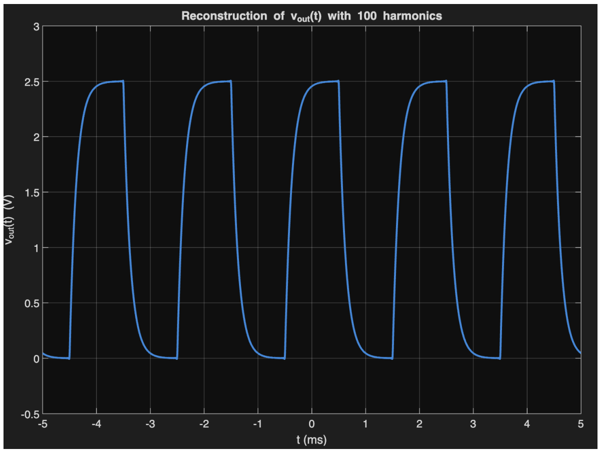




For problem 1a, the problem is solved by first creating variables to store the given information, which will be used in this problem, as well as all the others. Next, we create a coefficient vector to store all of the coefficient values ck by computing their complex Fourier series coefficients. This is done by using the 100 harmonics on each side, in the range (-100 to 100), the code then reconstructs $v_s(t)$ by taking the summation. Once we have this, we then plot the waveform over several periods. A second figure then plots the magnitude spectrum $|c_k|$ for the harmonic frequencies between -10000π and +10000π, showing the discrete line spectrum of the input signal.

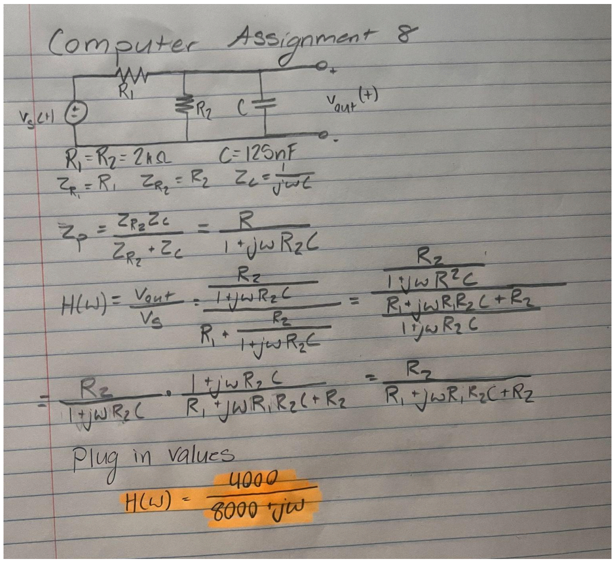
Problem 2:





In this problem, the output coefficients are computed using the LTI-frequency domain relationship $d_k = H(kw_0)c_{k,}$ where H(w) is the transfer function that I computed on paper. The code then applies this transfer function to each harmonic of the input signal. Using these output coefficients, the code then reconstructs $v_{out}(t)$ with the same Fourier series method and plots the output, which is a smoothed waveform. The magnitude spectrum $|d_k|$ is also plotted over the range from -10,000π to +10,000π, showing how the high frequency components are attenuated.

Problem 3:


R_1
R_2
C
$V_s(t)$
$V_{out}(t)$
+
-

Computer Assignment 8

$R_1 = R_2 = 2k\Omega \qquad C = 125nF$

$Z_{R_1} = R_1 \quad Z_{R_2} = R_2 \quad Z_C = \frac{1}{j\omega C}$

$$Z_p = \frac{Z_{R_2} Z_C}{Z_{R_2} + Z_C} = \frac{R}{1 + j\omega R_2 C}$$

$$H(\omega) = \frac{V_{out}}{V_s} = \frac{\frac{R_2}{1 + j\omega R_2 C}}{R_1 + \frac{R_2}{1 + j\omega R_2 C}} = \frac{\frac{R_2}{1 + j\omega R^2 C}}{\frac{R_1 + j\omega R_1 R_2 C + R_2}{1 + j\omega R_2 C}}$$

$$= \frac{R_2}{1 + j\omega R_2 C} \cdot \frac{1 + j\omega R_2 C}{R_1 + j\omega R_1 R_2 C + R_2} = \frac{R_2}{R_1 + j\omega R_1 R_2 C + R_2}$$

Plug in values

$$H(\omega) = \frac{4000}{8000 + j\omega}$$

The transfer function was solved on paper due to the added complexity of solving it in MATLAB. The transfer function my calculations result in was H(w) = 4000 / (8000 + jw). The code uses this expression when forming the output Fourier coefficients in problem 2.

Problem 4:

By comparing the magnitude spectra of $v_s(t)$ and $v_{out}(t)$, the MATLAB results show that high-frequency harmonics in the input are significantly reduced in the output results. Through this, low-frequency components appear to remain unchanged. Furthermore, the reconstructed output waveform is smoother with rounded edges, indicating the suppression of rapid changes. This behavior matches the magnitude response of our transfer function, which decreases as frequency increases. Therefore, the circuit functions as a low-pass filter, passing slow components while attenuating high-frequency components.

Appendix:

```
% Computer Assignment 8 --- Luis Kligman
clear all;
close all;
clc;
```

```
%% Problem 1a: Plot v_s(t) using 100 harmonics
T = 2e-3;  % Period in seconds
w0 = 2*pi/T;  % 1000*pi rad/s
K = 100;  % Number of harmonics on each side
k = -K:K;
% Coefficient vector
ck = zeros(size(k));
% Fill ck
for i = 1:length(k)
   ki = k(i);
   if ki == 0
       ck(i) = 5/2;  % c0
   else
       ck(i) = (5./(pi*ki)) * sin(ki*pi/2);
   end
end
% Time axis for plotting several periods
t = linspace(-5e-3, 5e-3, 4000);
% Reconstruct v_s(t) using complex exponential series
vs = zeros(size(t));
for i = 1:length(k)
   vs = vs + ck(i) .* exp(1j * k(i) * w0 .* t);
end
figure;
plot(t*1e3, real(vs), 'LineWidth', 1.2);
grid on;
xlabel('t (ms)');
ylabel('v_s(t) (V)');
title('Reconstruction of v_s(t) with 100 harmonics');
%% Problem 1b: Magnitude Spectrum of v_s(t)
% Frequency range
w_min = -10000*pi;
w_max = 10000*pi;
% Harmonics that fall in this range
k_spec = -10:10;
w_spec = k_spec * w0;
% Compute |c_k| for these k values:
ck_spec = zeros(size(k_spec));
for i = 1:length(k_spec)
   ki = k_spec(i);
   if ki == 0
       ck_spec(i) = 5/2;
   else
       ck_spec(i) = (5/(pi*ki)) * sin(ki*pi/2);
   end
end
% Plot the magnitude spectrum
figure;
stem(w_spec, abs(ck_spec), 'filled');
```

```
grid on;
xlabel('\omega (rad/s)');
ylabel('|c_k|');
title('Magnitude Spectrum of v_s(t)');
xlim([w_min, w_max]);
%% Problem 2a: Plot v_out(t) using 100 harmonics
% Solved transfer function on paper; H(jw)
H = @(w) 4000 ./ (8000 + 1j*w);
% Output Fourier coefficients d_k
dk = ck .* H(k * w0);
% Reconstruct v_out(t)
vout = zeros(size(t));
for i = 1:length(k)
   vout = vout + dk(i) .* exp(1j * k(i) *w0 .* t);
end
figure;
plot(t*1e3, real(vout), 'LineWidth', 1.8);
grid on;
xlabel('t (ms)');
ylabel('v_{out}(t) (V)');
title('Reconstruction of v_{out}(t) with 100 harmonics');
%% Problem 2b: Magnitude spectrum of v_out(t)
k_spec = -10:10;
w_spec = k_spec * w0;
dk_spec = zeros(size(k_spec));
for i = 1:length(k_spec)
   ki = k_spec(i);
   if ki == 0
       dk_spec(i) = H(0) * (5/2);
   else
       ck_i = (5/(pi*ki)) * sin(ki*pi/2);
       dk_spec(i) = H(ki*w0) * ck_i;
   end
end
figure;
stem(w_spec, abs(dk_spec), 'filled');
grid on;
xlabel('\omega (rad/s)');
ylabel('|d_k|');
title('Magnitude Spectrum of v_{out}(t)');
xlim([-10000*pi 10000*pi]);
```