% This script will plot the output of a half-wave rectified sine wave
% applied to a high-pass filter with cut-off at .5*(fundamental freq.)
% and another case with cut-off at 4*(fundamental freq.).
% You will need my mfile unit.m to successfully run this. It will also
% Plot the TF and Fourier spectra.

w0 = 200*pi;      % Define fundamental frequency
ne=[-10:2:10];    % Create integer vectors for even numbers
no = [-1,+1];     % Create integer vectors for odd numbers
fs = 100*10*5;    % Set sampling frequency 5 times the highest harmonic
t = [0:round(fs/50)]/fs;  % Time axis to plot 2 periods
vi = sin(w0*t).*unit(sin(w0*t));  % create input

figure(1)
plot(t,vi)
axis([0 .02 -.6 1.1]) % Fix axis on all plots to make for easier
comparison
title('Input - Half-Wave Rectified Sine wave')
xlabel('Seconds')
ylabel('volts');

dvie=(1/pi)*[(1 ./((1-ne.^2)))];   % Even harmonic FS coefs for input
dvio =(-j/4)*no;                % Odd harmonic FS coefs for input

% Input reconstructed from first 10 harmonics
nesize = length(ne);  %number of times to loop through even harmonics
nosize = length(no);  %number of times to loop through odd harmonics
vir = zeros(1,length(t));  % Initial initialize place to accumulate output
for k=1:nesize
    vir = vir + dvie(k)*exp(j*w0*t*ne(k));  % Sum up even harmonics
end
for k=1:nosize
    vir = vir + dvio(k)*exp(j*w0*t*no(k));  % Sum up odd harmonics
end

figure(2)
plot(t,real(vir))
axis([0 .02 -.6 1.1]) % Fix axis on all plots to make for easier
comparison
title('Input - Reconstructed with 10 harmonics')
xlabel('Seconds')
ylabel('volts');

% For 1/(RC) = .5w0
hjwe = (j*2*ne)./ (1 + j*2*ne);   % TF at even harmonics
hjwo = (j*2*no)./ (1 + j*2*no);   % TF at odd harmonics
dv0e = hjwe.*dvie;       % Even harmonic FS coefs for output
dv0o = hjwo.*dvio;       % Odd harmonic FS coefs for output
nesize = length(ne);  %number of times to loop through even harmonics
nosize = length(no);  %number of times to loop through odd harmonics
v0 = zeros(1,length(t));  % Initialize place to accumulate output
for k=1:nesize
    v0 = v0 + dv0e(k)*exp(j*w0*t*ne(k));  % Sum up even harmonics
end
for k=1:nosize
    v0 = v0 + dv0o(k)*exp(j*w0*t*no(k));  % Sum up odd harmonics
end
figure(3)
plot(t, real(v0))
axis([0 .02 -.6 1.1])
title('Output1 - Filtered at wc = .5*w0')
xlabel('Seconds')
ylabel('volts');

% For 1/(RC) = 4*w0

hjwe = (j*.25*ne)./(1 + j*.25*ne);   % TF at even harmonics
hjwo = (j*.25*no)./(1 + j*.25*no);   % TF at odd harmonics

dv0e = hjwe.*dvie;       % Even harmonic FS coefs for output
dv0o = hjwo.*dvio;       % Odd harmonic FS coefs for output
nesize = length(ne);  %number of times to loop through even harmonics
nosize = length(no);  %number of times to loop through odd harmonics
v0 = zeros(1,length(t));  % Initialize place to accumulate output

for k=1:nesize
   v0 = v0 + dv0e(k)*exp(j*w0*t*ne(k));  % Sum up even harmonics
end
for k=1:nosize
   v0 = v0 + dv0o(k)*exp(j*w0*t*no(k));  % Sum up odd harmonics
end

figure(4)
plot(t, real(v0))
axis([0 .02 -.6 1.1])  % Fix axis on all plots to make for easier comparison
title('Output1 - Filtered at wc = 4*w0')
xlabel('Seconds')
ylabel('volts');

% Plot case 2 TF magnitude and input FS coefs on same plot;

figure(5)
stem(w0*ne/(2*pi), abs(dvie));
hold on
stem(w0*no/(2*pi), abs(dvio));
hold on
w = [-10*w0:10*w0];  % Create frequency axis for finer evaluation
tf = (j*w/(4*w0)) ./ (1 + (j*w/(4*w0)));
plot(w/(2*pi), abs(tf));
hold off
title('TF and Input magnitudes in Frequency Domain')
xlabel('Hertz')
ylabel('volts')

% Do the same for the phase:

figure(6)
stem(w0*ne/(2*pi), -sign(ne).*angle(dvie)*(180/pi));
hold on
stem(w0*no/(2*pi), angle(dvio)*(180/pi));
hold on
plot(w/(2*pi), angle(tf)*(180/pi));
hold off
title('TF and Input phases in Frequency Domain')
xlabel('Hertz')
ylabel('Degrees')
Figure 1

Input - Half-Wave Rectified Sine wave

Figure 2

Input - Reconstructed with 10 harmonics
Figure 3
Output1 - Filtered at \( wc = 0.5w_0 \)

Figure 4
Output1 - Filtered at \( wc = 4w_0 \)
Figure 5
TF and Input magnitudes in Frequency Domain

Figure 6
TF and Input phases in Frequency Domain