EE3TR4 – Lab 2 Report

Yiming Chen, 400230266

Ruiyi Deng, 400240387

cheny466@mcmaster.ca

dengr6@mcmaster.ca

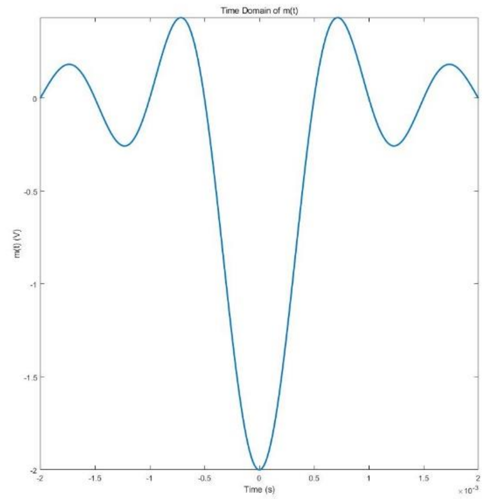
Mar 5th, 2022

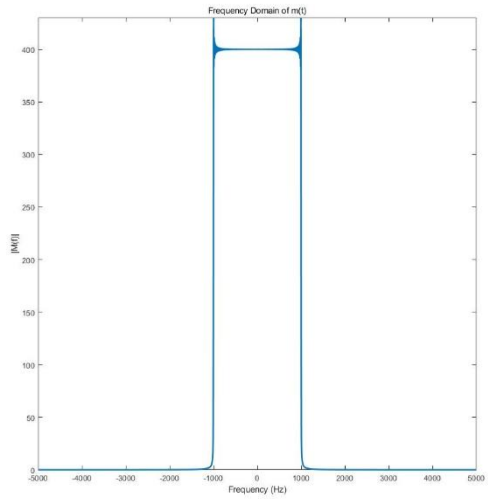
Generating and Plot Message Signal

The message signal is given by $m(t) = -2\backslash sinc(t/T_m)$ in which T_m= 0.0005s. Below is the plot of time domain and frequency domain of the message signal. We can see the magnitude spectrum of the message signal in the frequency domain. Also some part of the code used to generate and plot the message signal are shown below along with the plot.

```
%% Modulation

%carrier
fc = 20e3;
Ac = 1;
ct = Ac*cos(2*pi*fc*tt);

Tm = 0.0005;
mt = -2*sinc(tt/Tm); % message signal
```

```
%% Q1: Plotting the message signal

tiledlayout(1,2);

% time domain
nexttile;
plot(tt, mt, 'LineWidth', 2);
tim_dom_ax = gca;
set(tim_dom_ax);
xlabel('Time (s)');
ylabel('m(t) (V)');
title('Time Domain of m(t)');
axis([-2e-3 2e-3 min(mt) max(mt)]);

% frequency domain
Mf1 = fft(fftshift(mt));
Mf = fftshift(Mf1);
abs_Mf = abs(Mf);

nexttile;
plot(freq, abs_Mf, 'LineWidth', 2);
freq_dom_ax = gca;
set(freq_dom_ax);
xlabel('Frequency (Hz)');
ylabel('|M(f)|');
title('Frequency Domain of m(t)');
axis ([-5e3 5e3 0 max(abs(Mf))]);


% Measure highest frequency component from positive side of spectrum
[max_val, max_freq_index] = max(abs_Mf((length(abs_Mf)/2+1):end));
highest_freq = freq(max_freq_index + length(freq)/2);

fprintf('highest freqeuncy = %f\n', highest_freq)
```





According to our calculation, the highest frequency component occurs at 995Hz, which means it is the frequency bin closest to 1000Hz. Below is the calculation procedure which we performed to calculate the magnitude spectrum of the message signal. When T_m = 0.0005, the rect function representing the magnitude spectrum of the message signal ranges from -1000 to 1000, which matches the magnitude spectrum we plotted in MATLAB.

$$m(t) \rightleftharpoons M(f)$$
$$-2sinc(t/T_m) \rightleftharpoons M(f)$$
$$sinc(t) \rightleftharpoons rect(f)$$
$$sinc((1/T_m)\, t) \rightleftharpoons |T_m| rect(T_m\, f)$$
$$-2sinc(t/T_m) \rightleftharpoons -2|T_m|\backslash rect(T_m\, f)$$
$$M(f) \ = \ -2|T_m| rect(T_m\, f)$$

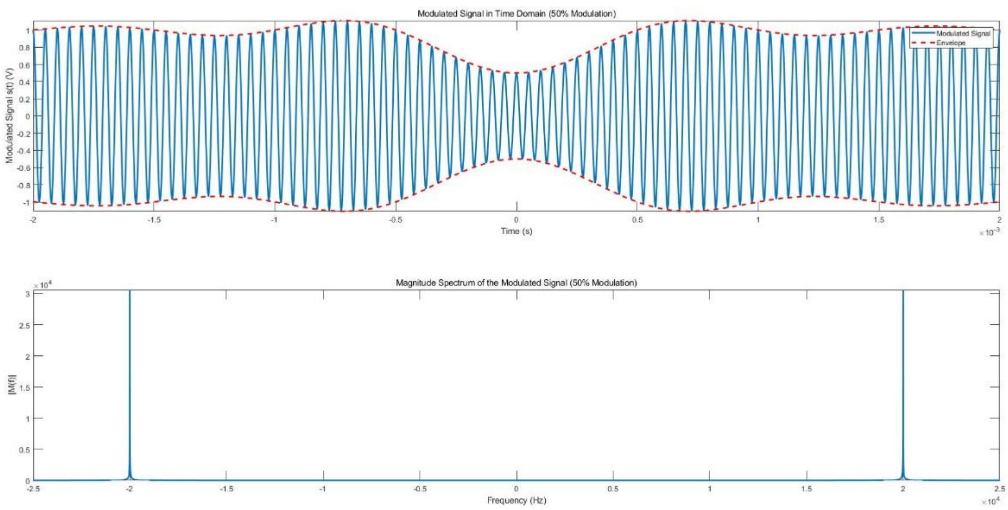
Amplitude Modulation

The modulated signal is given by $s(t) = A_c[1 + k_a m(t)]\cos(2\pi f_c t)$. And the carrier is given by $c(t) = A_c \cos(2\pi f_c t)$ where A_c = 1V and f_c = 20KHz. Attached is the code to generate and plot the modulated signal.

```
%% Q2: Mod and Demod
mt_max = max(abs(mt));

ka = 0.5/mt_max; % 50% modulation
% AM modulation
st = (1+ka*mt).*ct;

% Plot the modulated signal
modulated_sig(st, mt, ka, tt, freq, 2, ka);
```

```
function modulated_sig(signal, message, ka, tt, freq_vector, fig_num, percent_modulation)
    figure(fig_num)
    % Plotting the modulated signal
    tiledlayout(2,1);

    % time domain plotting s(t)
    nexttile;
    plot(tt, signal, 'LineWidth', 2);

    % plotting envelope
    hold on
    plot(tt, ka*message+1,'Color', 'r', 'LineStyle', '--', 'LineWidth', 2); % +1 for DC
    plot(tt, -ka*message-1,'Color', 'r', 'LineStyle', '--', 'LineWidth', 2);
    hold off


    legend('Modulated Signal', 'Envelope');
    tim_dom_ax = gca;
    set(tim_dom_ax);
    xlabel('Time (s)');
    ylabel('Modulated Signal s(t) (V)');

    title_name = "Modulated Signal in Time Domain (" + percent_modulation*100 + "% Modulation)";
    title(title_name);

    axis([-2e-3 2e-3 min(signal) max(signal)]);

    % frequency domain
    Sf1 = fft(fftshift(signal));
    Sf = fftshift(Sf1);

    nexttile;
    plot(freq_vector, abs(Sf), 'LineWidth', 2);
    freq_dom_ax = gca;
    set(freq_dom_ax);
    xlabel('Frequency (Hz)');
    ylabel('|M(f)|');
    title_name = "Magnitude Spectrum of the Modulated Signal (" + percent_modulation*100 + "% Modulation)";
    title(title_name);
    axis([-25e3 25e3 0 max(abs(Sf))]);

end
```

Below is the plot of the time and frequency of the modulated signal with 50% modulation. The frequency domain plot features the magnitude spectrum of the modulated signal.



Amplitude Demodulation

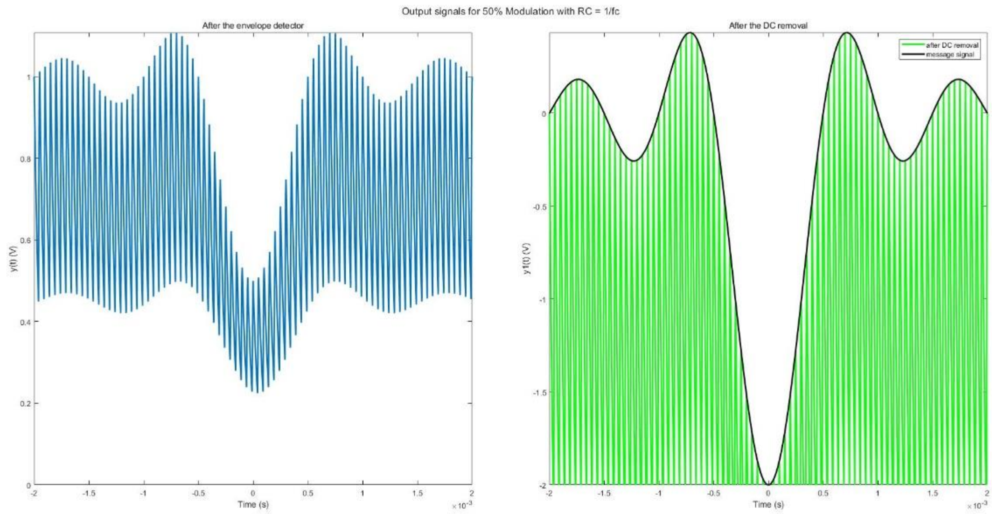
We wrote our own function of envelope detect to remove the DC component of the modulated signal to demodulate it. Below is our envelope detect function and our output fig function.

```
function yt = envelope_detect(st, RC, tt)
    yt = st;
    n = 1;
    for t = tt
        if (n > 1)
            if (yt(n-1) > st(n))
                yt0 = yt(n-1);
                % time when C starts discharging
                tc = tt(n-1);
                yt(n) = yt0*exp(-(t-tc)/RC);
            end
        end
        n = n+1;
    end

end
```

```
function output_fig(signal, original, ka, time_vector, fig_num, percent_modulation, RC_name)
    % Parameter lpf is used to control the Low-Pass Filter
    figure(fig_num);

    % To specify the figure for q(iii)

    tlayout = tiledlayout(1,2);


    title_name = "Output signals for " + percent_modulation*100 + "% Modulation with RC = " + RC_name;
    title(tlayout, title_name);

    % output of envelope detector
    nexttile;
    plot(time_vector, signal,'LineWidth',2);
    envelope_det_ax = gca;
    set(envelope_det_ax);
    xlabel('Time (s)');
    ylabel('y(t) (V)');
    title('After the envelope detector');
    axis([-2e-3 2e-3 0 max(signal)]);

    % dc removal and ka scale removal
    yt1 = (signal - 1) / ka;

    nexttile;
    plot(time_vector, yt1,'g', time_vector, original,'k','LineWidth',2);
    legend('after DC removal','message signal');
    output_signal_ax = gca;
    set(output_signal_ax);
    xlabel('Time (s)');
    ylabel('y1(t) (V)');
    title('After the DC removal');
    axis([-2e-3 2e-3 min(original) max(original)]);


end
```

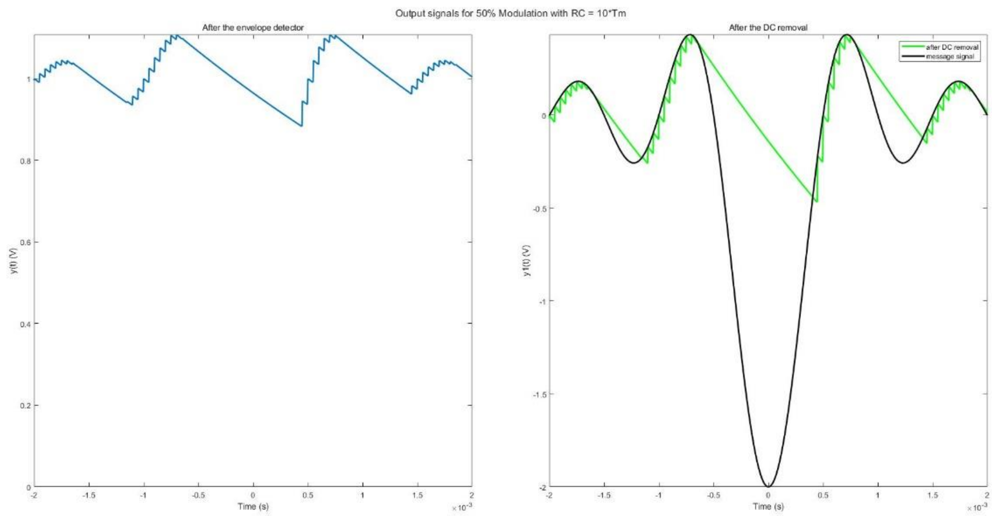
Output Signals for Time Constant $R_L C = 1/f_c$

Below are plot of the output signals for time constant $R_L C = 1/f_c$. We can see that the upper envelope of the output signal does follow the original message signal closely, but there is a significant rippling in the output signal. We may reduce the rippling by passing the signal through a low-pass filter, as rippling in the signal might give us a significant offset. Also, the rippling is possibly because the time constant set in this part is too low.



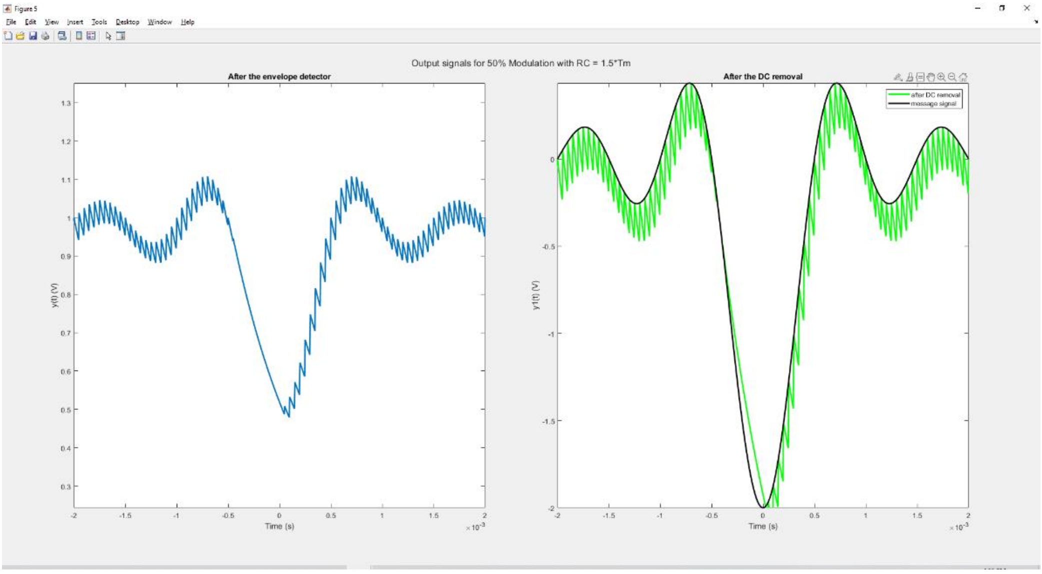
Output Signals for Time Constant $R_L C = 10T_m$

Below are plot of the output signals for time constant $R_L C = 10T_m$ We can see that the output signal lags behind the input signal significantly whenever there is a drop in the input signal, and the curve following the increasing portions of the input signal is very jagged. When there is a drop in the input signal, it means there is a significant portion of input signal is lost, and it would be impossible to accurately retrieve the input signal whenever the signal decreases too rapidly for the envelope to follow. The reason of above phenomenon happened may be caused by the time constant in this part is set too high.



Output Signals for Optimal Time Constant $R_L C = 1.5T_m$

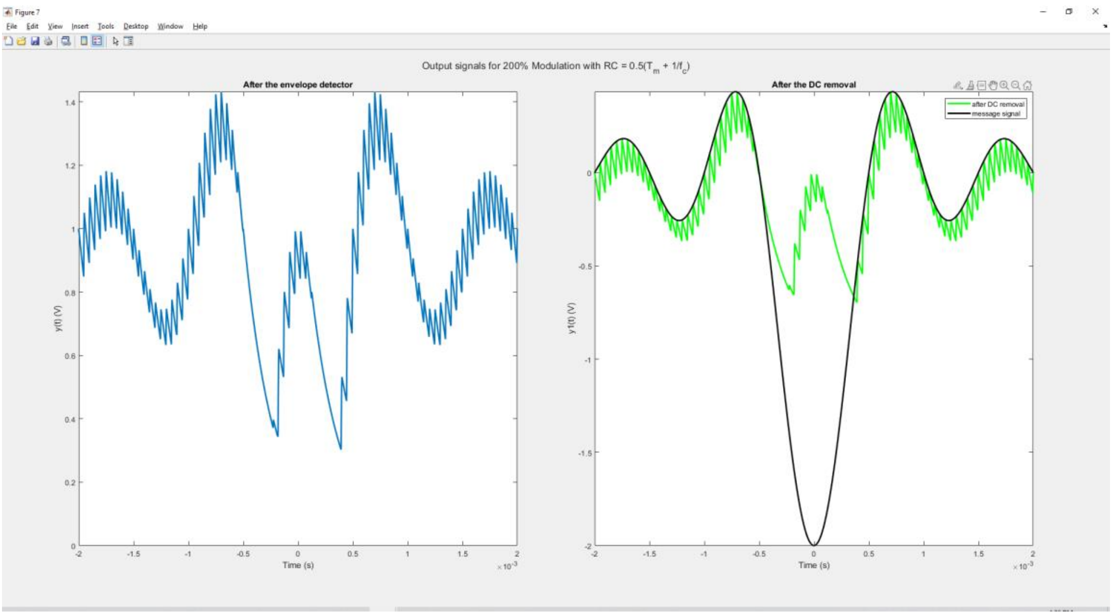
After testing for several times we finally determined a approximate range for the optimal time constant which is from 1*Tm – 2*Tm. In this range, it could avoid generating dense ripples by two sides and the middle part of the signal could follow the message signal instead of cases for time constant = 10*Tm like what is shown above.

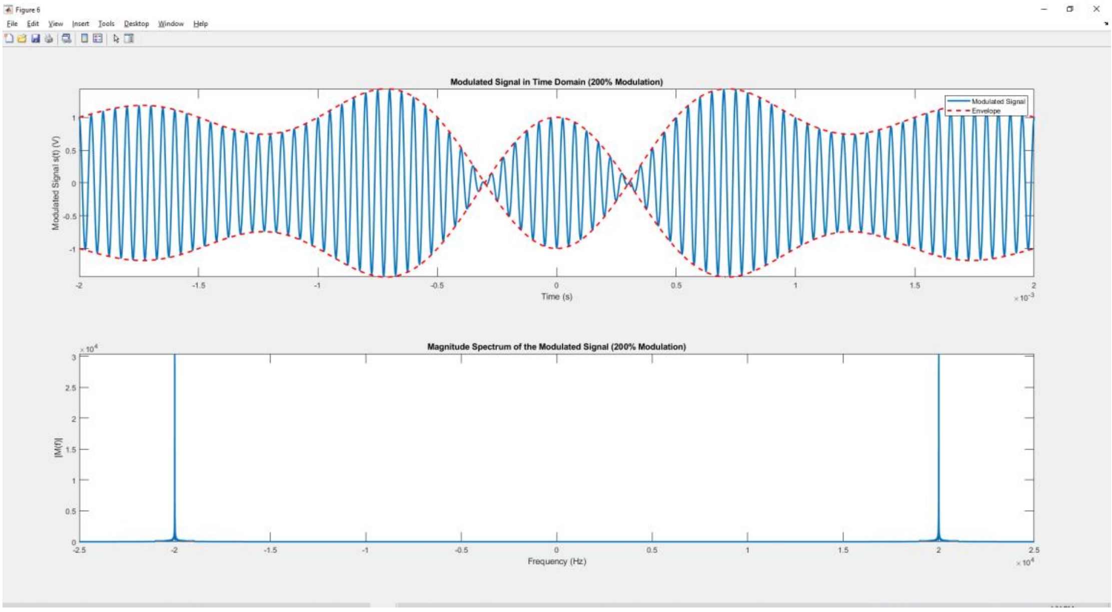


Apart from that, if we want to obtain a cleaner signal, we could apply a low pass filter to modify the ripples.

Output Signals for Amplitude Modulation of 200%

In this part, we changed the % of modulation to 200%. Intuitively, this will lead to the two envelopes crossing with each other. Below is the simulated result and from which the assumption is proved.





From the signal shown above it is obvious that in the crossing part, the signal fails to pass from the message signal. In other word, the envelopes ruined the demodulated signal. Thus, we deem that due to the crossing section when % of modulation over 100%, we cannot retrieve the signal from the message signal from demodulation.