EE 3TR4
Lab 2
-
Stefan Tosti - Tostis - 400367761 - L05
Rawan Sajid - Sajidr4 - 400393373 - L05
2024 - 03 - 03

Part 1 - Message Signal

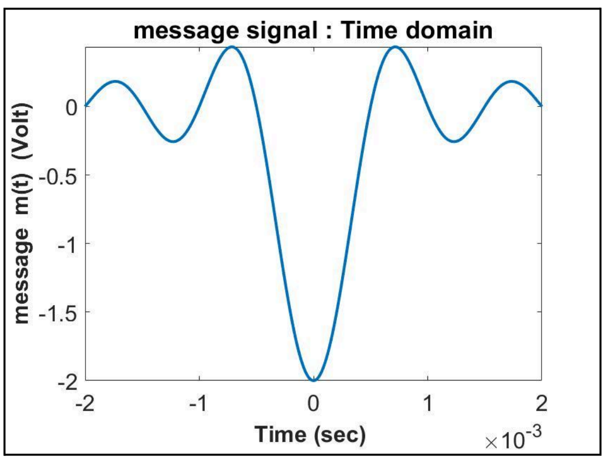
Given the message signal $m(t) - 2sinc\left(\frac{t}{T_m}\right)$

We can plot the message signal in time and frequency using the MATLAB code found below…

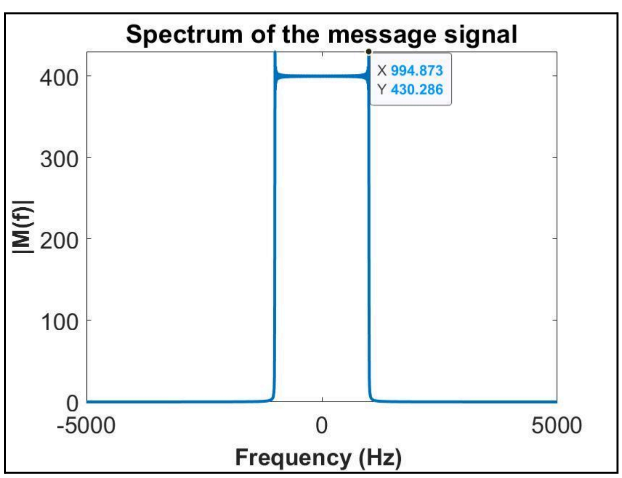
```
% Stefan Tosti 400367761
% Rawan Sajid 400393373

% time samples
N = 2^16; %No. of FFT samples
sampling_rate = 40e4; %unit Hz
tstep = 1/sampling_rate;
tmax = N*tstep/2;
tmin = -tmax;
tt = tmin:tstep:tmax-tstep;

% time samples
N = 2^16; %No. of FFT samples
sampling_rate = 40e4; %unit Hz
tstep = 1/sampling_rate;
tmax = N*tstep/2;
tmin = -tmax;
tt = tmin:tstep:tmax-tstep;

%message signal
Am=1;
fm = 1e3;
Tm = 0.0005;
mt = -2*sinc(tt/Tm);
%mt = Am*cos(2*pi*fm*tt);

% Message Signal Time domain
figure(2)
Hp1 = plot(tt,mt);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('message  m(t)  (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('message signal : Time domain');
```

```
axis([-2e-3 2e-3 min(mt) max(mt)])
pause(1)

% Spectrum of Message signal
Mf1 = fft(fftshift(mt));
Mf = fftshift(Mf1);
figure(4)
Hp1=plot(freq,abs(Mf));
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Frequency (Hz) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('|M(f)|');
set(Hx,'FontWeight','bold','Fontsize',16)
title('Spectrum of the message signal');
axis ([-5e3 5e3 0 max(abs(Mf))])
%pause(5)
```

The message signal in the time is as follows…



The message signal in frequency is as follows…



As seen above the Fourier transform of the Sinc function is the Rectangular function in the frequency domain. We can prove this below…

$$m(t) = -2sinc\left(\frac{t}{T_m}\right)$$

$$Duality\ Theorem:\ sinc(t) \Leftrightarrow rect(-f) = rect(f)$$
$$Scaling\ Theorem:\ g(at) = \frac{1}{|a|}G\left(\frac{f}{a}\right)$$

$$M(f) \Leftrightarrow -2\left|T_m\right|rect\left(fT_m\right)$$

As can be seen in the frequency graph above, the highest frequency component is around $995Hz$. When we compare the magnitude spectrum to the frequency response, we see that the rectangular function is highest between $\pm\ 1000Hz$

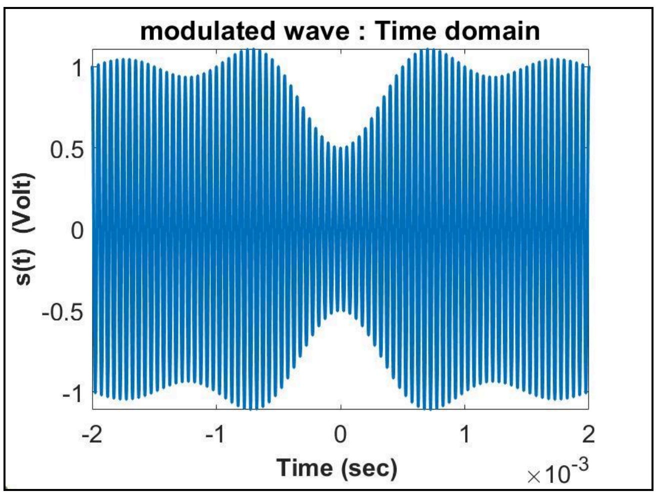
Part 2 - Envelope Detector

The following code was used to plot the modulated signal in frequency and time…

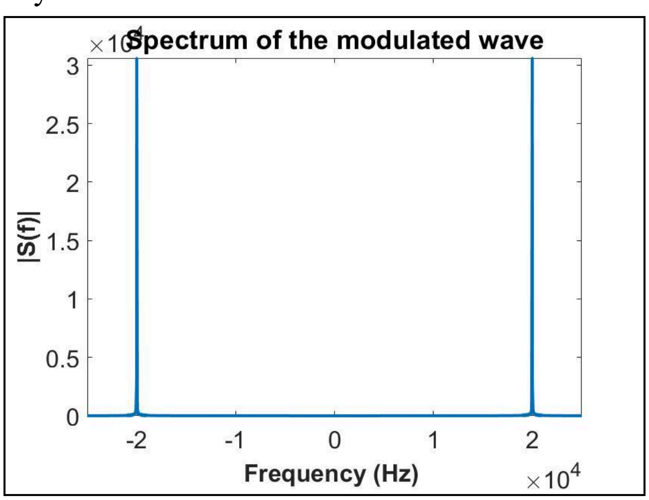
```
% Stefan Tosti 400367761
% Rawan Sajid 400393373

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

%message signal
Am=1;
fm = 1e3;
Tm = 0.0005;
mt = -2*sinc(tt/Tm);
%mt = Am*cos(2*pi*fm*tt);

%max of absolute of m(t)
maxmt = max(abs(mt));

%For 50% modulation
ka=0.5/maxmt;

%AM signal
st = (1+ka*mt).*ct;


% Modulated Signal, Time domain
figure(3)
Hp1 = plot(tt,st);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('s(t)  (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('modulated wave : Time domain');
axis([-2e-3 2e-3 min(st) max(st)])
pause(1)
```

```
% Spectrum of Modulated signal
Sf1 = fft(fftshift(st));
Sf = fftshift(Sf1);
figure(5)
Hp1=plot(freq,abs(Sf));
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Frequency (Hz) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('|S(f)|');
set(Hx,'FontWeight','bold','Fontsize',16)
title('Spectrum of the modulated wave');
axis ([-25e3 25e3 0 max(abs(Sf))])
%pause(5)
```

The modulated signal in time is as follows…



The modulated signal in frequency is as follows…



As we can see above, the maximum magnitude of the spectrum occurs at a frequency of $2 \times 10^4\ Hz$

Subsection 1

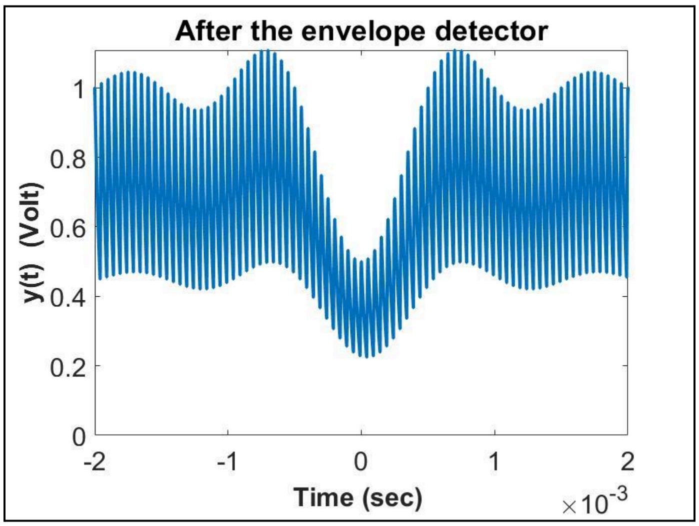
$R_L C = \frac{1}{f_c}$

The following MATLAB code was used to generate the below plots…

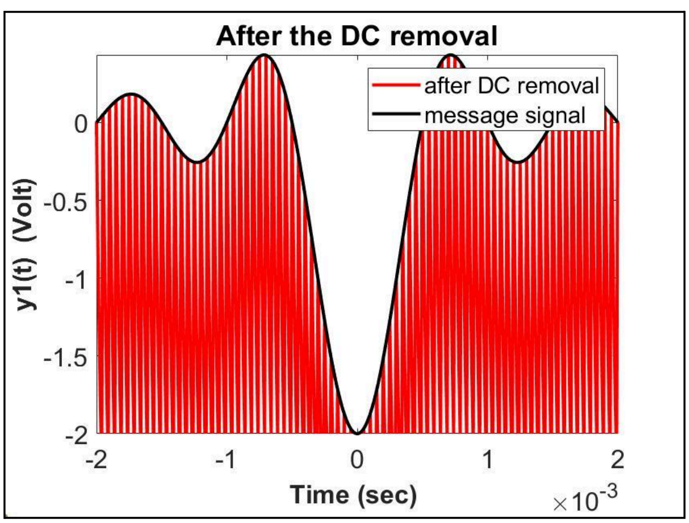
```
% Stefan Tosti 400367761
% Rawan Sajid 400393373

%time constant RC
%This should be optimized to avoid envelope distortion
RC = 1/fc;


%Envelope detector
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
yt(1)=yt(2);


figure(6)
Hp1 = plot(tt,yt);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('y(t)  (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('After the envelope detector');
axis([-2e-3 2e-3 0 max(yt)])
pause(1)
figure(7)
```

```
%DC removal and division by ka
yt1 = (yt - 1)/ka;
Hp1 = plot(tt,yt1,'r',tt,mt,'k');
legend('after DC removal','message signal')
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('y1(t)  (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('After the DC removal');
axis([-2e-3 2e-3 min(mt) max(mt)])
```

After the Envelope Detector…



After the DC removal…



As we can see the desired output and the message signal after being demodulated, are not close to the original message signal. We can first consider the graph showing the signal after the envelope detector. The upper bound of the envelope follows the general shape of the input signal quite closely. The lower bound of the envelope, however, does not follow the original signal very well, especially when in comparison to the original, 50% modulated signal.

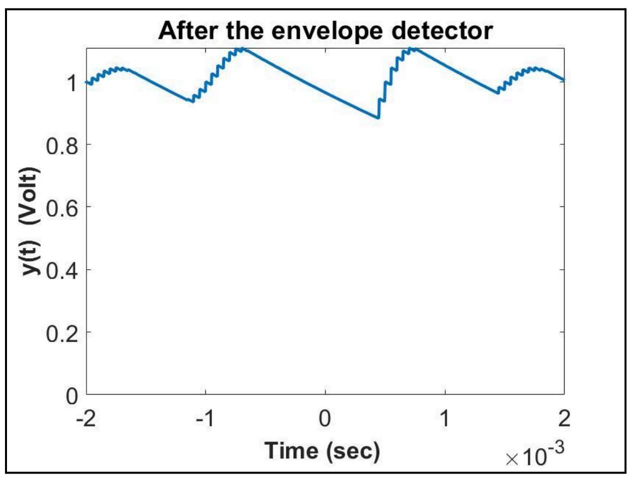
When we examine the output of the DC removal, we can once again see that the output signal does not follow the original input signal. We can see that the output closely follows the modulated signal, and as a result has a lot of oscillation. We believe that the RC value of the system is too small, and the charging and discharging time of the capacitor is too short.

Subsection 2

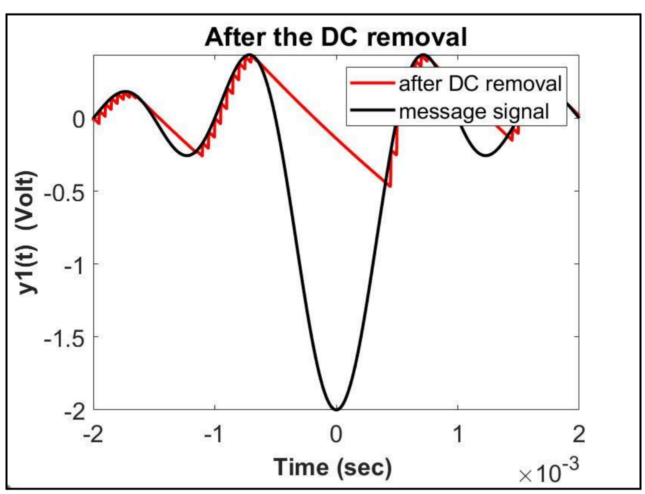
$R_L C = 10T_m$

By changing the value of $R_L C$ in the above plots we were able to generate the following plots

After the envelope detector…



After the DC removal…

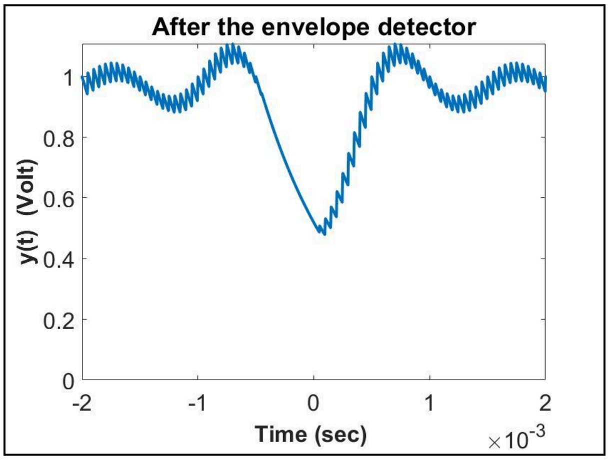


When the output time constant is set as it is above, we can see that the majority of our signal is lost. Looking at the 2 graphs above, we can see that as the message signal goes from maximum to minimum, the detector is sampling far too slowly to catch up, and a substantial amount of undersampling occurs. This is likely happening because the time constant is far too slow in this case, and thus we are retrieving data at too slow of a rate.

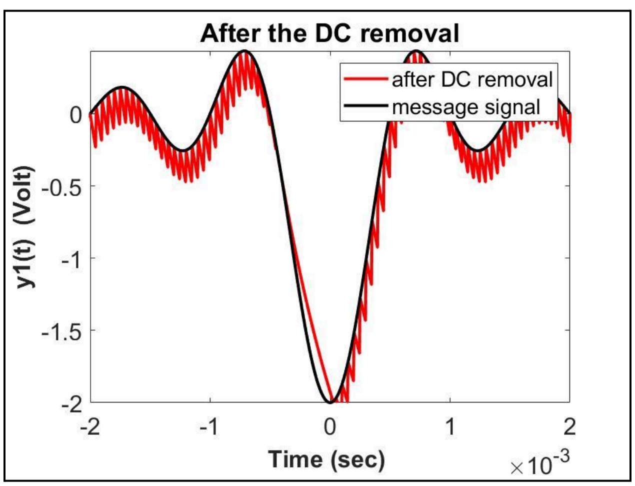
Subsection 3
In order to determine a proper range for the value of the time constant, $R_L C$, we used a mixture of trial and error, lecture notes, and the previous experiments as performed in this lab. As per the lecture notes, we know that $\frac{1}{f_c} \ll R_L C \ll \frac{1}{f_m}$, and we also know, from the 2 previous subsections, that $R_L C = \frac{1}{f_c}$ was too fast, and that $R_L C = 10T_m$ is too fast of a sampling rate.

The below graphs were generated using $R_L C = 1.5T_m$

After the envelope detector…



After DC removal…



Part 3 - 200% Modulation

From our lectures, we know that the envelope detector is only able to detect the positive envelope of a signal. That being, the envelope detector only works as intended when the message signal is entirely above the x-axis. Whenever the message signal crosses beneath the x-axis, we should expect to see significant distortion, as the signal is unreadable in this region.

In the case of 200% modulation, our envelope detector will only follow the portion of the message signal that is in the range of positive x-values. The regions of negative x will cause the message signal to cross over itself, and will thus result in significant distortion.

The above outlined problems can be resolved by implementing Double Sideband Suppressed Carrier Modulation, which is a form of modulation that can adequately process a message signal crossing into the negative region of x-values.

The below MATLAB code was used to generate the plots for the case of 200% modulation…

```
% Stefan Tosti 400367761
% Rawan Sajid 400393373


% time samples
N = 2^16; %No. of FFT samples
sampling_rate = 40e4; %unit Hz
tstep = 1/sampling_rate;
tmax = N*tstep/2;
tmin = -tmax;
tt = tmin:tstep:tmax-tstep;

%freq samples
fmax = sampling_rate/2;
fmin = -fmax;
fstep = (fmax-fmin)/N;
freq = fmin:fstep:fmax-fstep;

%% Modulation

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

%message signal
```

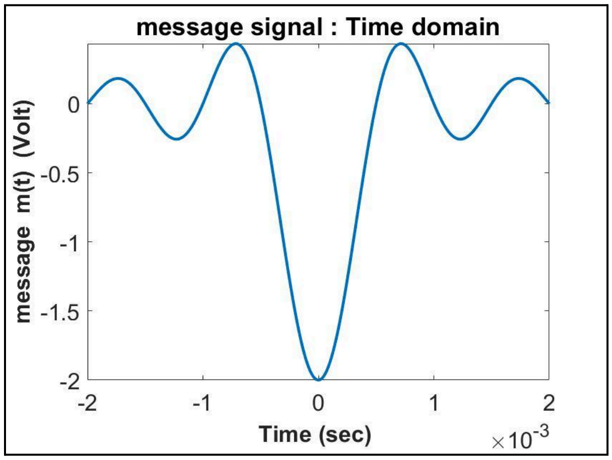
```
Am=1;
fm = 1e3;
Tm = 0.0005;
mt = -2*sinc(tt/Tm);
% mt = Am*cos(2*pi*fm*tt);
%maxmt = max(abs(mt));
ka = 2/maxmt;


%AM signal
st = (1+ka*mt).*ct;


% Carrier Signal, Time Domain
figure(1)
Hp1 = plot(tt,ct);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('Carrier c(t) (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('Carrier : Time domain');
axis([-1e-3 1e-3 -1.1 1.1])
pause(1)

% Message Signal Time domain
figure(2)
Hp1 = plot(tt,mt);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('message m(t) (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('message signal : Time domain');
axis([-2e-3 2e-3 min(mt) max(mt)])
pause(1)

% Modulated Signal, Time domain
figure(3)
```

```matlab
Hp1 = plot(tt,st);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('s(t) (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('modulated wave : Time domain');
axis([-2e-3 2e-3 min(st) max(st)])
pause(1)

% Spectrum of Message signal
Mf1 = fft(fftshift(mt));
Mf = fftshift(Mf1);
figure(4)
Hp1=plot(freq,abs(Mf));
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Frequency (Hz) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('|M(f)|');
set(Hx,'FontWeight','bold','Fontsize',16)
title('Spectrum of the message signal');
axis ([-5e3 5e3 0 max(abs(Mf))])
pause(5)

% Spectrum of Modulated signal
Sf1 = fft(fftshift(st));
Sf = fftshift(Sf1);
figure(5)
Hp1=plot(freq,abs(Sf));
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Frequency (Hz) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('|S(f)|');
set(Hx,'FontWeight','bold','Fontsize',16)
title('Spectrum of the modulated wave');
axis ([-25e3 25e3 0 max(abs(Sf))])
%pause(5)
```

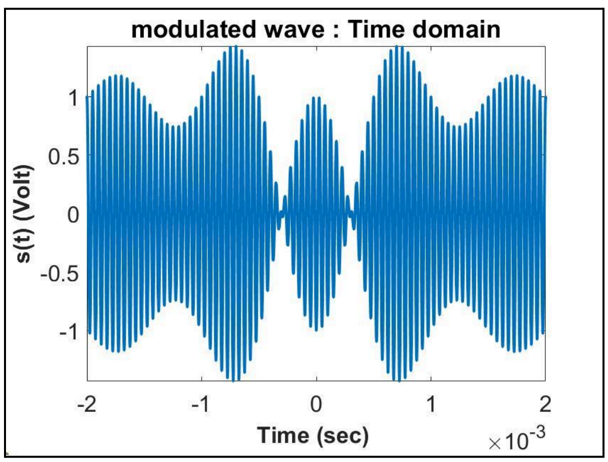
```
%% Demodulation
RC = 0.5*(Tm + 1/fc);
%Envelope detector
yt = st;
n=1;
for t=tt
    if(n > 1)
        if(yt(n-1) > st(n))
            yt0 = yt(n-1);
            %time when C starts discharging
            tc = tt(n-1);
            yt(n) = yt0*exp(-(t-tc)/RC);
        end
    end
    n=n+1;
end
yt(1)=yt(2);


figure(6)
Hp1 = plot(tt,yt);
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('y(t) (Volt)');
set(Hx,'FontWeight','bold','Fontsize',16)
title('After the envelope detector');
axis([-2e-3 2e-3 0 max(yt)])
pause(1)
figure(7)

%DC removal and division by ka
yt1 = (yt - 1)/ka;
Hp1 = plot(tt,yt1,'r',tt,mt,'k');
legend('after DC removal','message signal')
set(Hp1,'LineWidth',2)
Ha = gca;
set(Ha,'Fontsize',16)
Hx=xlabel('Time (sec) ');
set(Hx,'FontWeight','bold','Fontsize',16)
Hx=ylabel('y1(t) (Volt)');
```

```
set(Hx,'FontWeight','bold','Fontsize',16)
title('After the DC removal');
axis([-2e-3 2e-3 min(mt) max(mt)])
```

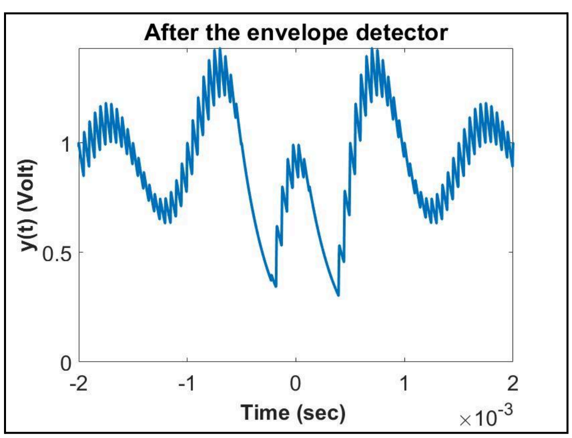
Recall the shape of the message signal…



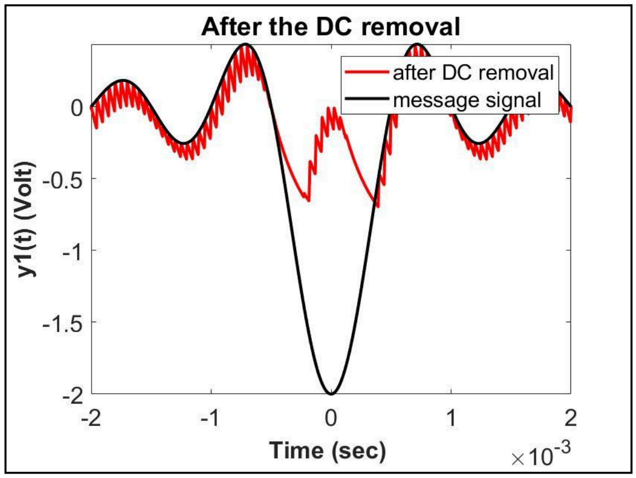
The output of the modulated signal in the time domain…



The output of the envelope detector can be seen below…



The output after the envelope detector can be found below…



As can be seen from the output plots, our original prediction was correct. There is significant distortion in the signal and we are unable to recover the message.