BIRZEIT UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING

Communication Systems

ENEE 3309

Course Project

Student's Name: Tariq Odeh

Student's No: 1190699

Sec: 1

Instructor: Dr. Wael Hashlamoun

Date: 5th December 2021

Table of Contents

List Of Figures

1. Introduction

In this project we will produce a normal AM waveform using a simple switching modulator circuit. The resultant AM waveform is then modulated using an envelope detector circuit. We will cover amplitude modulation and demodulation. We will implement it using MATLAB software, so we want to write MATLAB code for plotting modulated, and demodulated signals in time domain, and evaluating the optimum value of the time constant that minimizes the mean square error between modulated signal and the output signal of the envelope detector.

2. Procedure and Discussion

2.1. Part I

Use MATLAB (m-file commands) to plot $s(t)$ assuming μ = 0.25, Ac = 1, fm = 1Hz, fc = 25Hz over two cycles of the message m(t) = cos(2πfmt).

Explanation:

In the code, we cleared all variables for acquiring accurate results and closed all workspaces. To generate s(t), first we defined the variables (μ = 0.25, Ac = 1, fc = 25Hz, fm = 1Hz, Tc = 1/fc, Tm = 1/fm), and created time vector (starts from 0 to 2*Tm (two cycles of the message) increases by 0.0001 each time), then created a tau vector (starts from Tc to Tm increases by 0.0001 each time). And then generated message signal m(t) = cos(2πfmt) and plot it, then generated AM signal $s(t)$ = Ac [1 + μ cos(2πfmt)] cos(2πfct) and plot it.

Code:

```
clear all
close all
clc

%AM signal s(t) = Ac[1 + u cos(2pifmt)]cos(2pifct)
u = 0.25;               %modulation index
Ac = 1;                 %amplitude of the carrier signal
fc = 25;                %frequency of the the carrier signal
fm = 1;                 %frequency of the modulating signal
Tc = 1/fc;
Tm = 1/fm;
t = 0: 0.0001 :2*Tm;    %time vector
tau = Tc: 0.0001 :Tm;   %1/fc << tau << 1/fm

%---------------------------------------------------------%
%plot modulating signal Message m(t) = cos(2pifmt)
mt = cos(2.*pi.*fm.*t);
subplot (3,1,1)
plot (t,mt)
axis([0 2 -2 2]);
grid on
title('Modulating Signal Message m(t)')
xlabel('Time (s)')
ylabel('Message m(t)')
```

Figure 1: code of defined the variables and message signal m(t)

```
%------------------------------------------------------------%
%plot AM signal s(t) = Ac[1 + u cos(2pifmt)]cos(2pifct)
st = (Ac.*(1 + u.*mt).*cos(2.*pi.*fc.*t));
subplot (3,1,2)
plot (t,st)
axis([0 2 -2 2]);
grid on
title('AM Signal s(t)')
xlabel('Time (s)')
ylabel('Signal Amplitude (volts)')

%------------------------------------------------------------%
```

Figure 2: code of AM signal s(t)

Sketch:

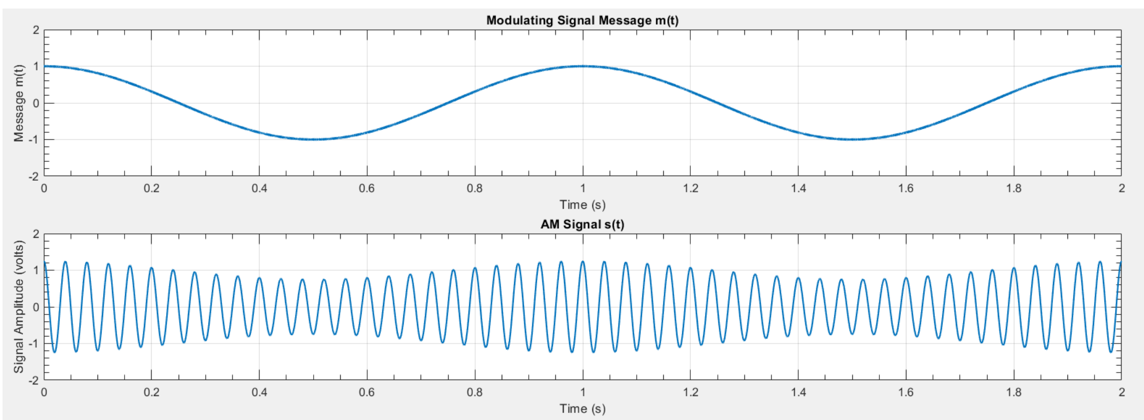


Figure 3: sketch of message signal m(t) and AM signal s(t)

2.2. Part II

If $s(t)$ is passed through an ideal envelope detector, plot the demodulated signal over two cycles of the message m(t).

Explanation:

Based on the previous code in Part I, we want to plot demodulated signal after $s(t)$ is passed through an ideal envelope detector. The ideal envelope detector is a circuit which takes the absolute value of its input, and then passes the result through a lowpass filter. So, we take the absolute value of s(t) and save the output in yi = abs (Ac.*(1 + u.*cos (2. *Pi. *fm.*t))), and then plot the demodulated signal yi.

Code:

```
%----------------------------------------------------------%
%s(t)passed through an ideal envelope detector,
%plot the demodulated signal
yi = abs(Ac.*(1 + u.*cos(2.*pi.*fm.*t)));
subplot (3,1,3)
plot (t,yi,t,st)
axis([0 2 -2 2]);
grid on
title('Demodulated Signal yi(t)')
xlabel('Time (s)')
ylabel('yi(t)')

%----------------------------------------------------------%
```

Figure 4: code of demodulated signal

Sketch:

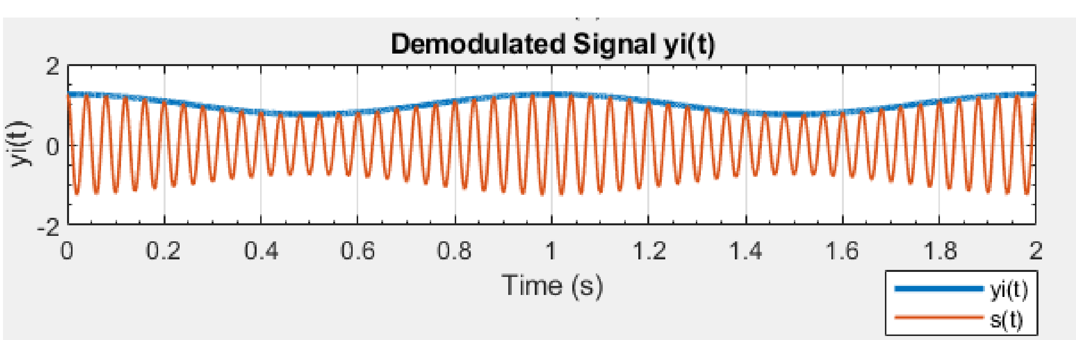


Figure 5: sketch of demodulated signal

2.3. Part III

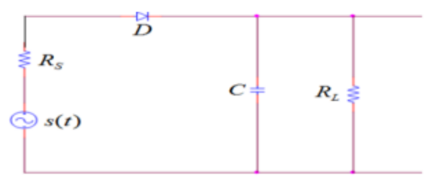
Assume that $s(t)$ is passed through the envelope detector shown in the figure to produce the waveform $y(t)$, where $R_s = 0$ and the diode is ideal ($V_D = 0$). In class, we put the following condition on the time constant of the circuit for best performance

$$\frac{1}{f_c} \ll \tau = R_L C \ll \frac{1}{f_m}$$

Define the mean squared error between $s(t)$ and $y(t)$ as:

$$D = \frac{1}{T_m}\int_0^{T_m} (y(t) - m(t))^2 dt$$

a. Plot D versus $\frac{1}{f_c} \leq \tau \leq \frac{1}{f_m}$
b. From the figure, determine the optimum value of the time constant that minimizes D.
c. Plot $y(t)$ that corresponds to the minimum D



Explanation:

Based on the previous code in Part I and Part II, we made 2 loops, the first for the values of tau and the second for the value of time, and based on the data, we calculated the value of the output according to the case of diode and then we use the given law to calculate the values of D, then we found the lowest value of D and the value of tau that gives this value. Finally, we plot the demodulated D versus tau, plot y(t), and print the minimum value of D, value of the time constant that minimizes D.

```
Command Window

The minimum value of D:
    0.0014


value of the time constant that minimizes D:
    0.7463
```

Code:

```
%-------------------------------------------------------%
time_length = length(t);        %length of time vector
tau_length = length(tau);       %length of tau vector

y = zeros(1,time_length);             %output signal
D_matrix = zeros(1,tau_length);       %mean squared error

for i = 1: 1 :tau_length          %first loop for tau
    max = st(1);          %max = 1*[1+u*1*cos(0)]cos(0) --> 1+u --> 1.25
    delay = 0;
    D = 0;

    for j = 1: 1 :time_length   %second loop for tau
        %Diode is off
        if st(j) <  max*exp(-1*(t(j)-delay)/tau(i))
            y(j) =  max*exp(-1*(t(j)-delay)/tau(i));

        %Diode is on
        elseif st(j) >=  max*exp(-1*(t(j)-delay)/tau(i))
            y(j) = st(j);

            %update the max and delay
            if st(j) == yi(j)
                delay = t(j);
                max = st(j);
            end
        end
        D = D + ((yi(j) - y(j)).^2);  %find the integration value
    end
     D_matrix(i)=(1/time_length)*D;    %find the mean squared error value
end

%find minimum value of D
minimum_D = min(D_matrix);
disp('The minimum value of D:')
disp(minimum_D)

%find minimum value of time constant that minimizes D
[~,x] = min(D_matrix);
disp('value of the time constant that minimizes D:')
disp(tau(1,x))

figure
subplot (2,1,1)
plot(tau,D_matrix)
grid on
subplot (2,1,2)
plot(t,y)
grid on
```

Figure 6: code of part III

Sketch:

The minimum value of D is 0.014 when tau is 0.7463 s.

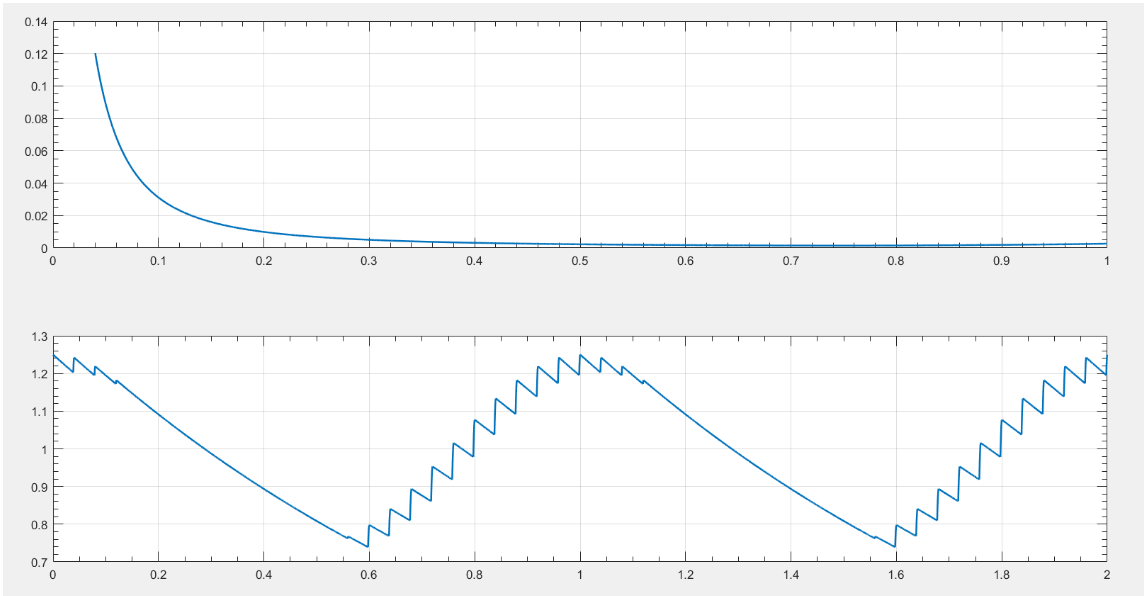


Figure 7: sketch of D versus tau and y(t)

3. Appendix

```
clear all
close all
clc

%AM signal s(t) = Ac[1 + u cos(2pifmt)]cos(2pifct)
u = 0.25;                 %modulation index
Ac = 1;                   %amplitude of the carrier
signal
fc = 25;                  %frequency of the the carrier
signal
fm = 1;                   %frequency of the modulating
signal
Tc = 1/fc;
Tm = 1/fm;
t = 0: 0.0001 :2*Tm;      %time vector
tau = Tc: 0.0001 :Tm;     %1/fc << tau << 1/fm

%-------------------------------------------------------
--%
%plot modulating signal Message m(t) = cos(2pifmt)
mt = cos(2.*pi.*fm.*t);
subplot (3,1,1)
plot (t,mt)
axis([0 2 -2 2]);
grid on
title('Modulating Signal Message m(t)')
xlabel('Time (s)')
ylabel('Message m(t)')

%-------------------------------------------------------
--%
%plot AM signal s(t) = Ac[1 + u cos(2pifmt)]cos(2pifct)
st = (Ac.*(1 + u.*mt).*cos(2.*pi.*fc.*t));
subplot (3,1,2)
plot (t,st)
axis([0 2 -2 2]);
grid on
title('AM Signal s(t)')
xlabel('Time (s)')
ylabel('Signal Amplitude (volts)')

%-------------------------------------------------------
--%
%s(t)passed through an ideal envelope detector,
```

```
%plot the demodulated signal
yi = abs(Ac.*(1 + u.*cos(2.*pi.*fm.*t)));
subplot (3,1,3)
plot (t,yi,t,st)
axis([0 2 -2 2]);
grid on
title('Demodulated Signal yi(t)')
xlabel('Time (s)')
ylabel('yi(t)')

%-------------------------------------------------------
--%
time_length = length(t);        %length of time vector
tau_length = length(tau);       %length of tau vector

y = zeros(1,time_length);             %output signal
D_matrix = zeros(1,tau_length);       %mean squared
error

for i = 1: 1 :tau_length          %first loop for tau
    max = st(1);          %max = 1*[1+u*1*cos(0)]cos(0) -
-> 1+u --> 1.25
    delay = 0;
    D = 0;

    for j = 1: 1 :time_length   %second loop for tau
        %Diode is off
        if st(j) <  max*exp(-1*(t(j)-delay)/tau(i))
            y(j) =  max*exp(-1*(t(j)-delay)/tau(i));

        %Diode is on
        elseif st(j) >=  max*exp(-1*(t(j)-
delay)/tau(i))
            y(j) = st(j);

            %update the max and delay
            if st(j) == yi(j)
                delay = t(j);
                max = st(j);
            end
        end
        D = D + ((yi(j) - y(j)).^2);  %find the
integration value
    end
     D_matrix(i)=(1/time_length)*D;   %find the mean
squared error value
end
```

```
%find minimum value of D
minimum_D = min(D_matrix);
disp('The minimum value of D:')
disp(minimum_D)

%find minimum value of time constant that minimizes D
[~,x] = min(D_matrix);
disp('value of the time constant that minimizes D:')
disp(tau(1,x))

figure
subplot (2,1,1)
plot(tau,D_matrix)
grid on
subplot (2,1,2)
plot(t,y)
grid on
```