AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Course Title: Data Communication[G]

Lab Report-5

Exp. Title: Study of Digital to Analog Conversion using MATLAB

Submitted by: [Group-2]

Name	ID
Shadril Hassan Shifat	20-42451-1
Sadia Sultana Ali	20-42386-1
Abu Shaleh Md. Kaium	20-42475-1
Md. Ali Ahnaf	20-42378-1

Submitted to: Tanjil Amin

Date of Submission: 30 March, 2022

PerformanceTask:

Assume your ID is **AB-CDEFG-H**, and then convert ‘E’, ‘F’ and ‘G’ to 8 bit ASCII characters and together you have a bit stream of 24 bits. Convert this bit stream to analog signal using the following:

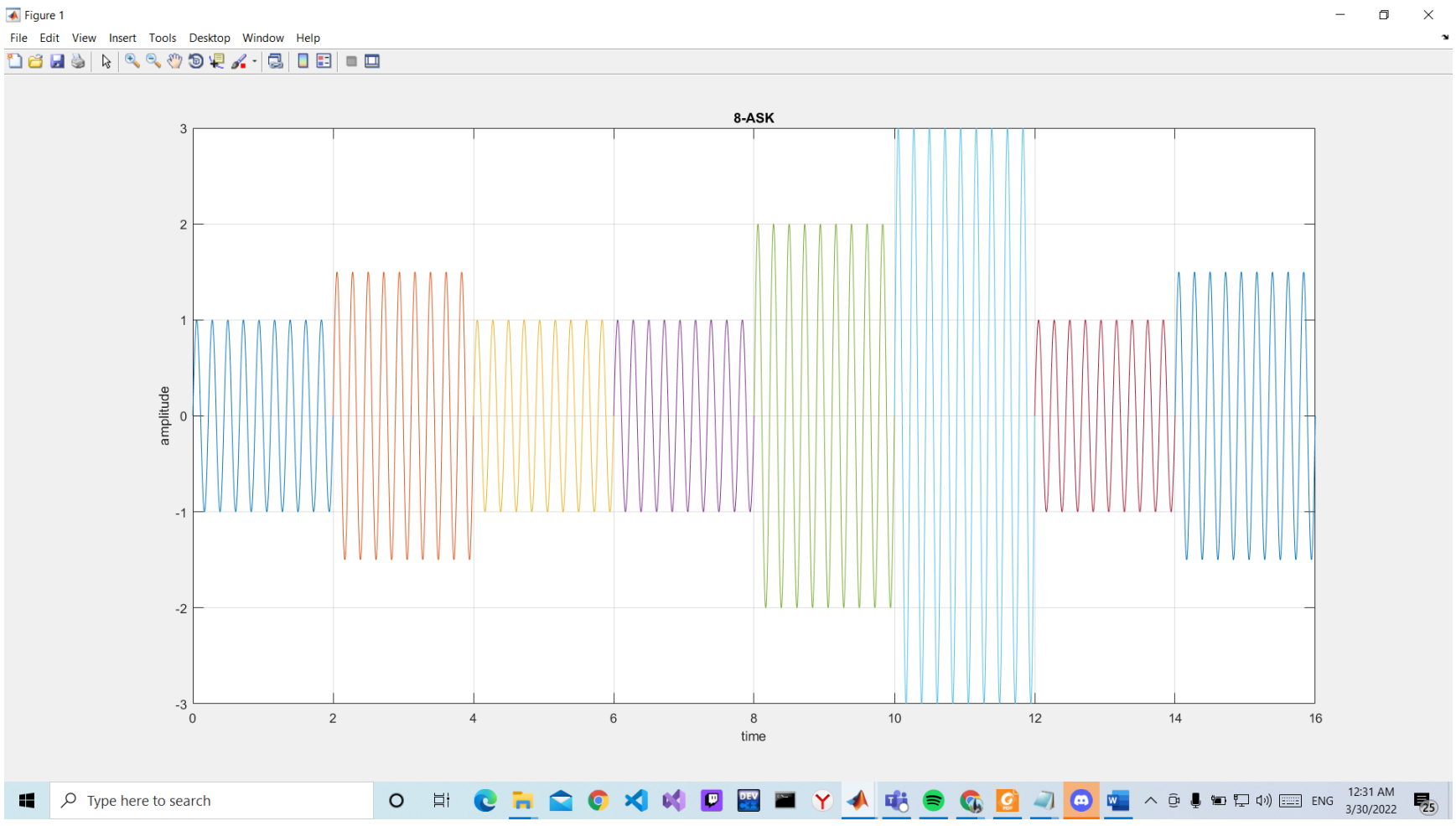
1.8-ASK, different amplitudes in the modulated signal can be 1:0.5:4.5 for 000 to 111 in that order.

MATLAB Code:

```
close all;
clc;
clear all;
%ID=AB-CDEFG-H
A=2;
B=0;
C=4;
D=2;
E=4;
F=5;
G=1;
H=1;

% Converting E=4, F=5 and G=1 to 8 bit ASCII characters array
bitStream_1= dec2bin(E,8)-'0';
bitStream_2= dec2bin(F,8)-'0';
bitStream_3= dec2bin(G,8)-'0';
% genarating bit stream array
bitStream= [bitStream_1 bitStream_2 bitStream_3];
% calculating length of bit stream
n=length(bitStream);
ts=2;
%frequency
fd=4.5;
fs=1000;
i=1;
g=0;

while i:3:n
    %width of any pulse
    t=g*ts:1/fs:(g+1)*ts;
    %condition
    if bitStream(i)==0 && bitStream(i+1)==0 && bitStream(i+2)==0
    ask=1*sin(2*pi*fd*t);

elseif bitStream(i)==0 && bitStream(i+1)==0 && bitStream(i+2)==1
    ask=1.5*sin(2*pi*fd*t);

elseif bitStream(i)==0 && bitStream(i+1)==1 && bitStream(i+2)==0
    ask=2*sin(2*pi*fd*t);

elseif bitStream(i)==0 && bitStream(i+1)==1 && bitStream(i+2)==1
    ask=2.5*sin(2*pi*fd*t);

elseif bitStream(i)==1 && bitStream(i+1)==0 && bitStream(i+2)==0
    ask=3*sin(2*pi*fd*t);

elseif bitStream(i)==1 && bitStream(i+1)==0 && bitStream(i+2)==1
    ask=3.5*sin(2*pi*fd*t);

elseif bitStream(i)==1 && bitStream(i+1)==1 && bitStream(i+2)==0
    ask=4*sin(2*pi*fd*t);

elseif bitStream(i)==1 && bitStream(i+1)==1 && bitStream(i+2)==1
    ask=4.5*sin(2*pi*fd*t);
```

```
    end
    %plotting graph
    plot(t,ask);
    title('8-ASK');
    xlabel('time');
    ylabel('amplitude')
    hold on;
    grid on;

    i=i+3;
    g=g+1;
end;
hold off;
```

Output:



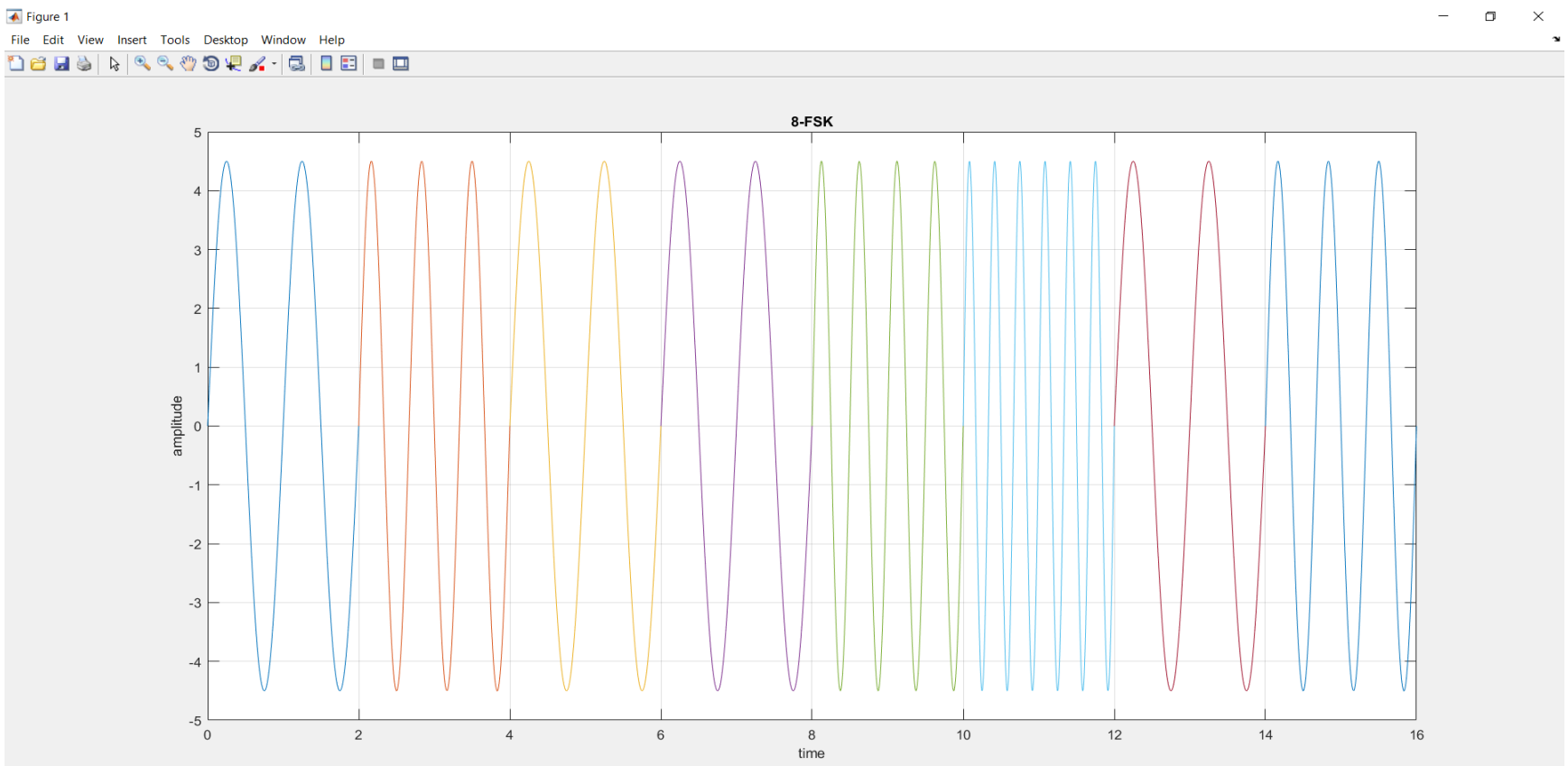
2. 8-FSK, different frequencies in the modulated signal can be 1:0.5:4.5 for 000 to 111 in that order.

MATLAB Code:

```
close all;
clc;
clear all;
%ID=AB-CDEFG-H
A=2;
B=0;
C=4;
D=2;
E=4;
F=5;
G=1;
H=1;

% Converting E=4, F=5 and G=1 to 8 bit ASCII characters array
bitStream_1= dec2bin(E,8)-'0';
bitStream_2= dec2bin(F,8)-'0';
bitStream_3= dec2bin(G,8)-'0';
% genarating bit stream array
bitStream= [bitStream_1 bitStream_2 bitStream_3];
% calculating length of bit stream
n=length(bitStream);
```

```
ts=2;
%frequency
fd=4.5;
fs=1000;
i=1;
g=0;
while i:3:n
    %width of any pulse
    t=g*ts:1/fs:(g+1)*ts;
    %condition
    if bitStream(i)==0 && bitStream(i+1)==0 && bitStream(i+2)==0
        fsk=4.5*sin(2*pi*t);

    elseif bitStream(i)==0 && bitStream(i+1)==0 && bitStream(i+2)==1
        fsk=4.5*sin(2*pi*1.5*t);

    elseif bitStream(i)==0 && bitStream(i+1)==1 && bitStream(i+2)==0
        fsk=4.5*sin(2*pi*2*t);

    elseif bitStream(i)==0 && bitStream(i+1)==1 && bitStream(i+2)==1
        fsk=4.5*sin(2*pi*2.5*t);

    elseif bitStream(i)==1 && bitStream(i+1)==0 && bitStream(i+2)==0
        fsk=4.5*sin(2*pi*3*t);

    elseif bitStream(i)==1 && bitStream(i+1)==0 && bitStream(i+2)==1
        fsk=4.5*sin(2*pi*3.5*t);

    elseif bitStream(i)==1 && bitStream(i+1)==1 && bitStream(i+2)==0
        fsk=4.5*sin(2*pi*4*t);

    elseif bitStream(i)==1 && bitStream(i+1)==1 && bitStream(i+2)==1
        fsk=4.5*sin(2*pi*4.5*t);

    end

    %plotting graph
    plot(t,fsk);
    hold on;
    grid on;
    title('8-FSK');
    xlabel('time');
    ylabel('amplitude')

    i=i+3;
    g=g+1;
end;
hold off;
```

Output:



3. 8-PSK, different phases in the modulated signal can be 0◦:45◦:315◦ for 000 to 111 in that order.

MATLAB Code:

```
close all;
clc;
clear all;
%ID=AB-CDEFG-H
A=2;
B=0;
C=4;
D=2;
E=4;
F=5;
G=1;
H=1;

% Converting E=4, F=5 and G=1 to 8 bit ASCII characters array
bitStream_1= dec2bin(E,8)-'0';
bitStream_2= dec2bin(F,8)-'0';
bitStream_3= dec2bin(G,8)-'0';
% genarating bit stream array
bitStream= [bitStream_1 bitStream_2 bitStream_3];
% calculating length of bit stream
n=length(bitStream);
ts=2;
%frequency
fd=4.5;
fs=1000;
i=1;
g=0;
while i:3:n
    %width of any pulse
    t=g*ts:1/fs:(g+1)*ts;
    %condition
    if bitStream(i)==0 && bitStream(i+1)==0 && bitStream(i+2)==0
        psk=4.5*sin(2*pi*fd*t);
    elseif bitStream(i)==0 && bitStream(i+1)==0 && bitStream(i+2)==1
        psk=4.5*sin(2*pi*fd*t+pi/4);
    elseif bitStream(i)==0 && bitStream(i+1)==1 && bitStream(i+2)==0
        psk=4.5*sin(2*pi*fd*t+pi/2);
    elseif bitStream(i)==0 && bitStream(i+1)==1 && bitStream(i+2)==1
        psk=4.5*sin(2*pi*fd*t+(3*pi)/4);
    elseif bitStream(i)==1 && bitStream(i+1)==0 && bitStream(i+2)==0
        psk=4.5*sin(2*pi*fd*t+pi);
    elseif bitStream(i)==1 && bitStream(i+1)==0 && bitStream(i+2)==1
        psk=4.5*sin(2*pi*fd*t+pi+pi/4);
    elseif bitStream(i)==1 && bitStream(i+1)==1 && bitStream(i+2)==0
```

```
        psk=4.5*sin(2*pi*fd*t+pi+pi/2);
    elseif bitStream(i)==1 && bitStream(i+1)==1 && bitStream(i+2)==1
        psk=4.5*sin(2*pi*fd*t+pi+(3*pi)/4);
    end
    %plotting graph
    plot(t,psk);
    title('8-PSK');
    xlabel('time');
    ylabel('amplitude')
    hold on;
    grid on;


    i=i+3;
    g=g+1;
end;
hold off;
```

Output: