



Final Project Signals

Anas Emad Ahmed 7048

Mohab Ayman Abd-Elhamid 7127

Group 2 section 1

B)Code:

```
% question(2a)
fs=4000;
sincx = @(x) sin(pi*x+1E-8)./(pi*x + 1E-8);
x = linspace(-0.005,0.005,0.01*fs);
y = sincx(1000*x).^2;
subplot(1,2,1);
plot(x,y)
title('m(t)=Sinc(10^3*t)^2')

y1=fftshift(fft(y));
y1mag=abs(y1);
fvec=linspace(-fs/2,fs/2,0.01*fs);
subplot(1,2,2);
plot(fvec,y1mag)
title('M(w)=pi/1000*tri(w/(4*10^3))')

%question(2b)
fs=300000;
sincx = @(x) sin(pi*x+1E-8)./(pi*x + 1E-8);
x = linspace(-0.005,0.005,0.01*fs);
y = sincx(1000*x).^2;
r=y.*cos(2*pi*100000*x);
subplot(1,2,1);
plot(x,r)
title('r(t)=(Sinc(10^3*t)^2)*cos(2*pi*10^5*t')

y1=fftshift(fft(r));
y1mag=abs(y1);
fvec=linspace(-fs/2,fs/2,0.01*fs);
subplot(1,2,2);
plot(fvec,y1mag)
title('R(w)=pi/2000*[tri((w-
wo)/(4*10^3))+tri((w+wo)/(4*10^3))]')
```

PART II:

1-CODE

```

clc;
clear all;
fs = str2double(input('ENTER SAMPLING FREQUENCY:', 's'));
while(fs<=0)
    fs=str2double(input('INVALID INPUT, PLEASE ENTER POSITIVE SAMPLING FREQUENCY:', 's'));
end
start_time = str2double(input('ENTER START TIME:', 's'));
end_time = str2double(input('ENTER END TIME:', 's'));
while(end_time<=start_time)
    end_time = str2double(input('INVALID INPUT!, Enter valid end time:', 's'));
end
breaknum = str2double(input('ENTER NUMBER OF BREAKPOINTS:', 's'));
breakpoints = zeros(1, breaknum); % 1d array size of breaknum
t_begin=start_time; % will help in breakpoints
sign=zeros(1,breaknum+1); % array of signals to be drawn
for(i=1:breaknum)
    breakposition = str2double(input(['ENTER POSITION OF BREAKPOINT ' num2str(i)], 's'));
    while(breakposition<=start_time || breakposition>=end_time)
        breakposition= str2double(input(['INVALID BREAKPOINT, PLEASE ENTER POSITION OF BREAKPOINT ' num2str(i)], 's'));
    end
    breakpoints(i)=breakposition; % position of breakpoints
end
breakpoints=sort(breakpoints); % after sorting breakpoints in order

for(i=1:breaknum+1)

    fprintf("Signal Generator Menu:\n1-DC signal\n2-Ramp signal\n3-Polynomial Signal\n4-Exponential Signal\n5-Sinusoidal Signal\n");
    signal=input("Enter signal name",'s');

    while 1

if(strcmpi(signal,'DC')==1||strcmpi(signal,'Ramp')==1||strcmpi(signal,'Polynomial') ==1||strcmpi(signal,'Exponential')==1||strcmpi(signal,'Sinusoidal')==1)
    break;
else
    signal=input("INVALID SIGNAL!,please enter correct signal name",'s');

end
end

if(strcmpi(signal,'DC')==1)
    sign(i)=1;
else if(strcmpi(signal,'Ramp')==1)
    sign(i)=2;
else if(strcmpi(signal,'Polynomial')==1)
    sign(i)=3;

```

```
else if(strcmpi(signal,'Exponential')==1)
    sign(i)=4;
else if(strcmpi(signal,'Sinusoidal')==1)
    sign(i)=5;

end
end
end
end
end
end

SignalArray=[];
flagBreak=0; % to know if there is more breakpoints or not
t_total=linspace(start_time,end_time,(end_time - start_time)*fs);
PolyEq=0;
for(i=1:length.breakpoints)+1)

if(length.breakpoints)==0) % if no breakingpoint
    flagBreak=1; % no more breakpoints
    t_special1=start_time;
    t_special2=end_time;

else if(i==length.breakpoints)+1) % if last breakingpoint
    flagBreak=1;
    t_special1=t_begin; % last breakpoint
    t_special2=end_time;

else % general case in between signals
    flagBreak=0;
    t_special1=t_begin; %last breakpoint 0
    t_special2=breakpoints(i); % next breakpoint 2

end
end

if(sign(i)==1) % if DC
    DCAmplitude=str2double(input('Enter DC Amplitude','s'));
    temp=DCAmplitude*ones(1,(t_special2-t_special1)*fs);
    SignalArray=[SignalArray temp];
    if(flagBreak==0)
        t_begin=breakpoints(i);
    end

else if(sign(i)==2)% if Ramp
    RampSlope=str2double(input('Enter Ramp slope','s'));
    Ramp_Yinter=str2double(input('Enter Y intercept','s'));
    t_ramp=linspace(t_special1,t_special2,(t_special2-t_special1)*fs);
    temp=RampSlope*t_ramp +Ramp_Yinter;
    SignalArray=[SignalArray temp];
```

```

if(flagBreak==0)
    t_begin=breakpoints(i);
end

else if(sign(i)==3)%if poly
    t_Poly=linspace(t_special1,t_special2,(t_special2-t_special1)*fs);
    PolyPower=str2double(input('Enter Polynomial Power','s'));
    for(j=PolyPower:-1:1)
        PolyCoeff=str2double(input(['Enter Polynomial coefficient of '
num2str(j) ' power : ','s']));
        PolyEq=(PolyEq+PolyCoeff.*(t_Poly).^j);
    end
    PolyAmplitude = str2double(input('Enter polynomial amplitude:
','s'));
    PolyInter=str2double(input('Enter Polynomial Intercept','s'));
    PolyEq=PolyAmplitude.*(PolyEq+PolyInter);
    temp=PolyEq;
    SignalArray=[SignalArray temp];
if(flagBreak==0)
    t_begin=breakpoints(i);
end

else if(sign(i)==4)%if Exponential
    ExpAmp=str2double(input('Enter Exponential Amplitude','s'));
    ExpExpo=str2double(input('Enter Exponential Exponent','s'));
    t_Expo=linspace(t_special1,t_special2,(t_special2-t_special1)*fs);
    temp = ExpAmp*exp(ExpExpo*t_Expo);
    SignalArray=[SignalArray temp];
    if(flagBreak==0)
        t_begin=breakpoints(i);
    end

else if(sign(i)==5)%if sinusoidal

    while 1
        sinorcos=str2double(input('ENTER 1 for SIN AND 2 FOR COSINE:',
's'));
        if(sinorcos==1 || sinorcos==2)
            break;
        end
    end
    SinuAmp=str2double(input('Enter Sinusoidal Amplitude','s'));
    SinuFreq=str2double(input('Enter Sinusoidal Freq ','s'));
    SinuPhase=str2double(input('Enter Sinusoidal Phase','s'));
    t_Sinu=linspace(t_special1,t_special2,(t_special2-t_special1)*fs);
    if(sinorcos==1)
        temp=SinuAmp*sin(2*pi*SinuFreq*t_Sinu+SinuPhase);
        SignalArray=[SignalArray temp];
        if(flagBreak==0)
            t_begin=breakpoints(i);
        end
    else if(sinorcos==2)
        temp=SinuAmp*cos(2*pi*SinuFreq*t_Sinu+SinuPhase);
        SignalArray=[SignalArray temp];
    if(flagBreak==0)
        t_begin=breakpoints(i);
    end
end

```

```
    end
    end
end
end
end
end
end
end

plot(t_total,SignalArray); % draw signals

% part 2
SignalOperation=SignalArray;
t_Operation=t_total;
% acts as a stack to display each signal before modifying it
SignalOperationOld=SignalOperation;
t_OperationOld=t_Operation;
while 1
    operation=str2double(input('Choose any operation to be done on the signal, if not
press 7\n 1-Amplitude scaling\n 2-Time reversal\n 3-Time shift\n 4-Expanding\n 5-
Compressing\n 7-Exit','s')));

while(operation<0 || operation>7)
    operation=str2double(input('INVALID INPUT!, please enter a number between 1 ->
6 ','s')));
end

    subplot(2,2,[1,2]);
    plot(t_total,SignalArray);
    title('Original Signal');

if(operation==1)
    Scalevalue=str2double(input('Enter Scaling value','s'));
    SignalOperation=Scalevalue.*SignalOperation;
    subplot(2,2,3);
    plot(t_OperationOld,SignalOperationOld);
    subplot(2,2,4);
    plot(t_Operation,SignalOperation);
    title('Scaling');

else if(operation==2)

    t_Operation=t_Operation.*-1;
    subplot(2,2,3);
plot(t_OperationOld,SignalOperationOld);
    subplot(2,2,4);
    plot(t_Operation,SignalOperation)
    title('Time reversal');

else if(operation==3)
    Shiftvalue=str2double(input('Enter shift value','s'));
    t_Operation=t_Operation+Shiftvalue;
    subplot(2,2,3);
```

```
plot(t_OperationOld,SignalOperationOld);
    subplot(2,2,4);
        plot(t_Operation,SignalOperation);
        title('Shifting');

else if(operation==4)
    Expansion=str2double(input('Enter Expansion value','s'));
    t_Operation=t_Operation*Expansion;
    subplot(2,2,3);
plot(t_OperationOld,SignalOperationOld);
    subplot(2,2,4);
        plot(t_Operation,SignalOperation);
        title('Expansion');

else if(operation==5)
    Compression=str2double(input('Enter Compression value','s'));
    t_Operation=t_Operation/Compression;
    subplot(2,2,3);
plot(t_OperationOld,SignalOperationOld);
    subplot(2,2,4);
        plot(t_Operation,SignalOperation);
        title('Compression');

else
    close all;
    disp("THANK YOU FOR USING SIGNAL GENERATOR")
    break;

end

end
end
end
SignalOperationOld=SignalOperation; % to store the last signal before next
operation
t_OperationOld=t_Operation;
end
```

TEST CASES:

EXAMPLE 1-One Signal without any breakpoints

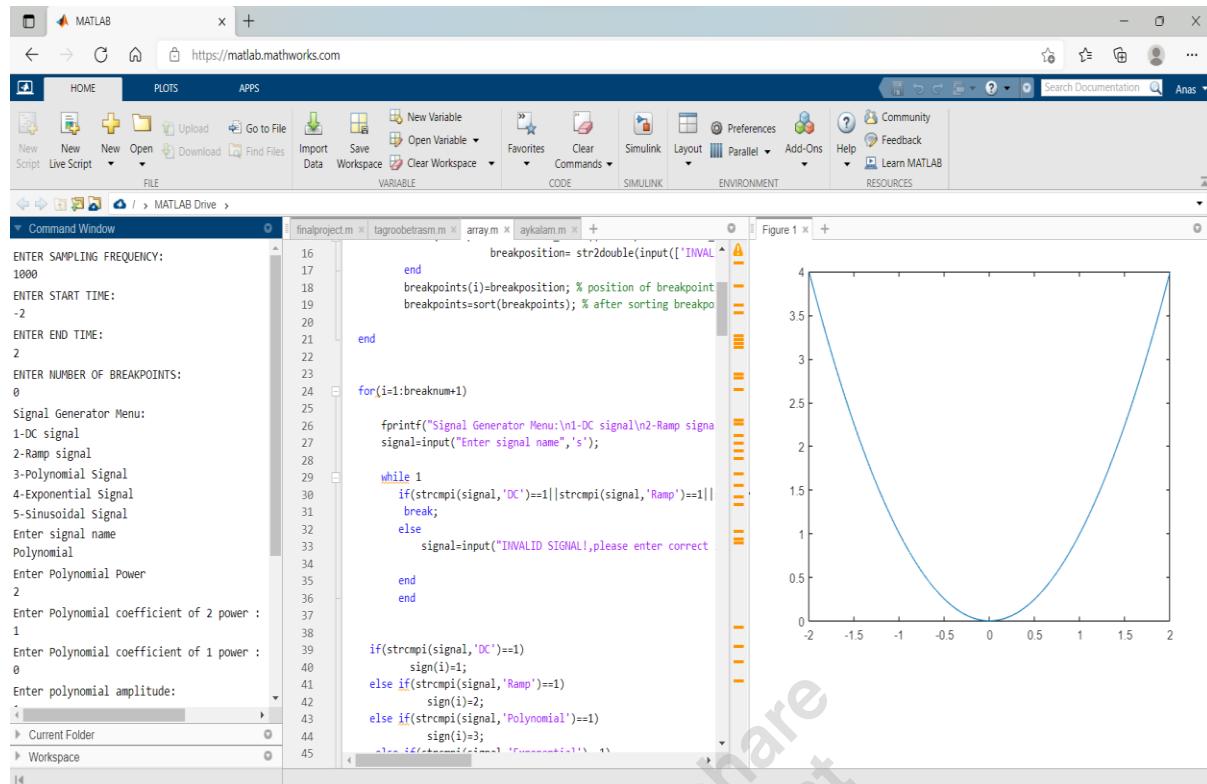
1-Fs=1000

2-Start time=-2

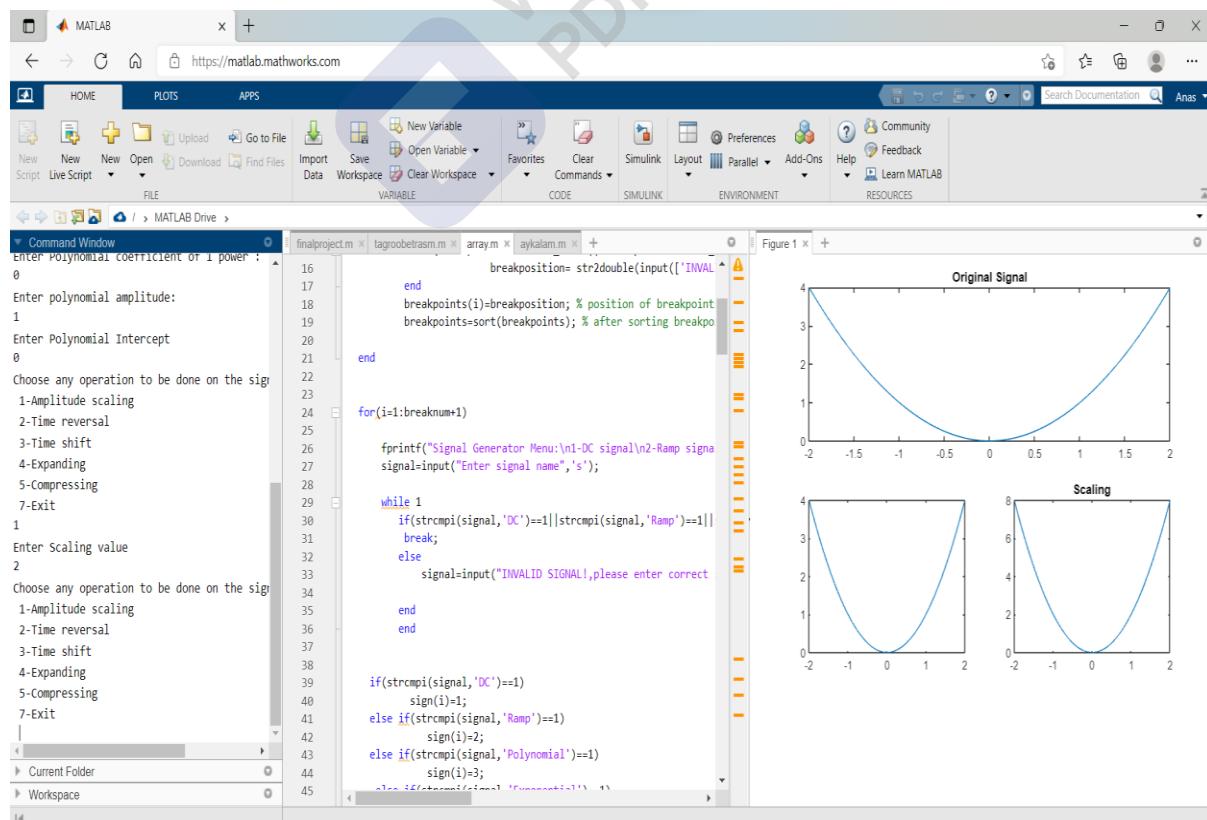
3-End time=2

4-Number of breakpoints=0

5-Signal=Polynomial



OPERATIONS



NOTE: The first plot is the original plot without any operations, the second plot acts as a stack each operation you will make the previous one will appear on the left plot, and finally the last plot is the last operation done by the user.

EXAMPLE 2: Signal with 2 breakpoints

1-Fs=4000

2-Start time=-5

3-End time=5

4-Number of breakpoints=2

5-Position of breakpoints= 0, -2

5-Signal=Polynomial , Sinusoidal , and Exponential

MATLAB

ENTER SAMPLING FREQUENCY:
4000

ENTER START TIME:
-5

ENTER END TIME:
5

ENTER NUMBER OF BREAKPOINTS:
2

ENTER POSITION OF BREAKPOINT 1
0

ENTER POSITION OF BREAKPOINT 2
2

Signal Generator Menu:
1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
Polynomial

Signal Generator Menu:
1-DC signal
2-Ramp signal

Figure 1

```

16    breakposition= str2double(input(['INVALID SIGNAL?']));
17    end
18    breakpoints(i)=breakposition; % position of breakpoint
19    breakpoints=sort.breakpoints); % after sorting breakpoints
20
21
22
23
24    for(i=1:breaknum+1)
25
26        fprintf("Signal Generator Menu:\n1-DC signal\n2-Ramp signal\n");
27        signal=input("Enter signal name",'s');
28
29        while 1
30            if(strcmpi(signal,'DC')==1)||strcmpi(signal,'Ramp')==1|
31            break;
32            else
33                signal=input("INVALID SIGNAL!,please enter correct
34                signal");
35            end
36        end
37
38        if(strcmpi(signal,'DC')==1)
39            sign(i)=1;
40        else if(strcmpi(signal,'Ramp')==1)
41            sign(i)=2;
42        else if(strcmpi(signal,'Polynomial')==1)
43            sign(i)=3;
44        else
45            sign(i)=1;
46        end

```

Enter Polynomial Power
2

Enter Polynomial coefficient of 2 power :
3

Enter Polynomial coefficient of 1 power :
1

Enter polynomial amplitude:
1

Enter Polynomial Intercept
4

ENTER 1 for SIN AND 2 FOR COSINE:
1

Enter Sinusoidal Amplitude
1

Enter Sinusoidal Freq
4

Enter Sinusoidal Phase
0

Enter Exponential Amplitude
2

Enter Exponential Exponent
1

Choose any operation to be done on the signal
1

Figure 1

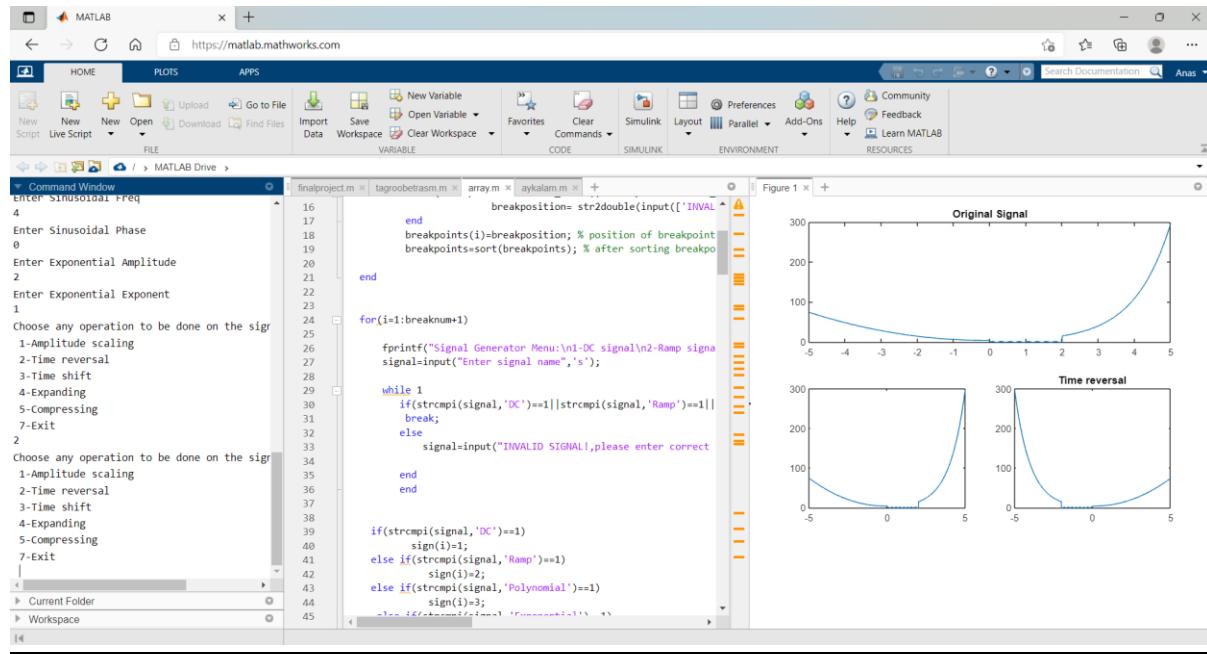
```

16    breakposition= str2double(input(['INVALID SIGNAL?']));
17    end
18    breakpoints(i)=breakposition; % position of breakpoint
19    breakpoints=sort.breakpoints); % after sorting breakpoints
20
21
22
23
24    for(i=1:breaknum+1)
25
26        fprintf("Signal Generator Menu:\n1-DC signal\n2-Ramp signal
27        ");
28        signal=input("Enter signal name",'s');
29
30        while 1
31            if(strcmpi(signal,'DC')==1)||strcmpi(signal,'Ramp')==1|
32            break;
33            else
34                signal=input("INVALID SIGNAL!,please enter correct
35                signal");
36            end
37
38            if(strcmpi(signal,'DC')==1)
39                sign(i)=1;
40            else if(strcmpi(signal,'Ramp')==1)
41                sign(i)=2;
42            else if(strcmpi(signal,'Polynomial')==1)
43                sign(i)=3;
44            else
45                sign(i)=1;
46            end

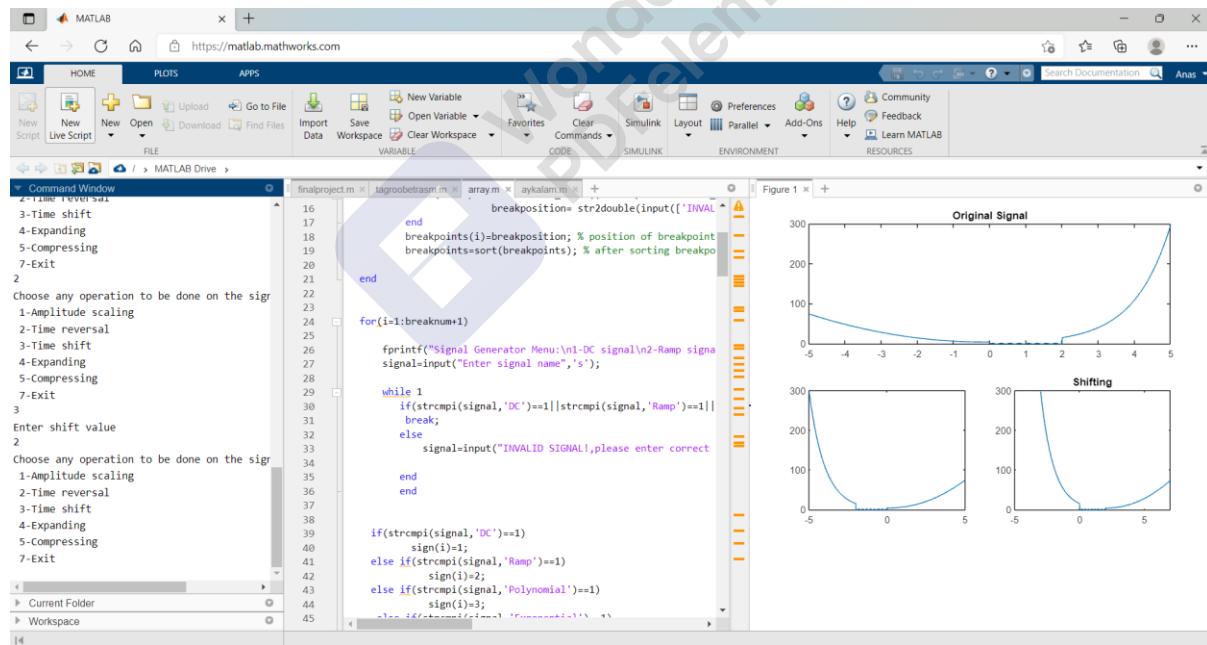
```

OPERATIONS:

1-Time Reversal



2-Time Shift(after Time Reversal)



EXAMPLE 3: Signal with 3 breakpoints

1-Fs=1000

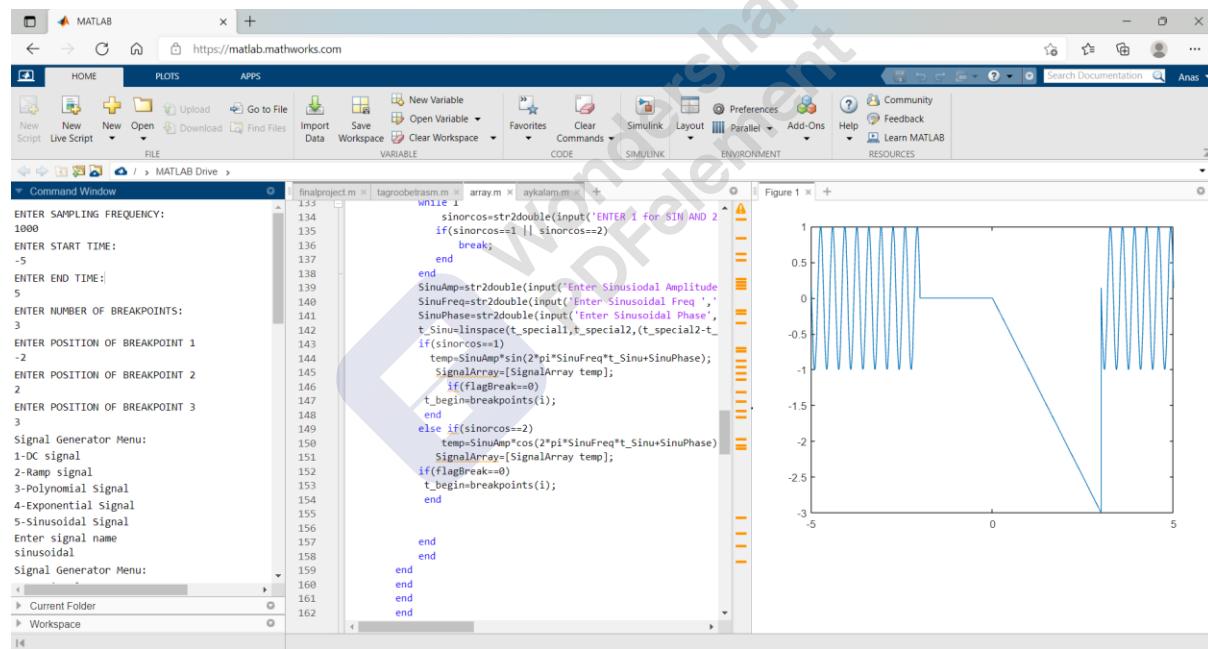
2-Start Time=-5

3-End Time=5

4-Num of breakpoints=3

5-Position of breakpoints=-2,2,3

6-Signals:Sin,DC,Ramp,Sin



INPUT: Signal Generator Menu:

1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name

sinusoidal

ENTER 1 for SIN AND 2 FOR COSINE:

1

Enter Sinusoidal Amplitude

1

Enter Sinusoidal Freq

3

Enter Sinusoidal Phase

3

Enter DC Amplitude

0

Enter Ramp slope

-1

Enter Y intercept

0

ENTER 1 for SIN AND 2 FOR COSINE:

1

Enter Sinusoidal Amplitude

1

Enter Sinusoidal Freq

3

Enter Sinusoidal Phase

3

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

2

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

4

Enter Expansion value

5

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

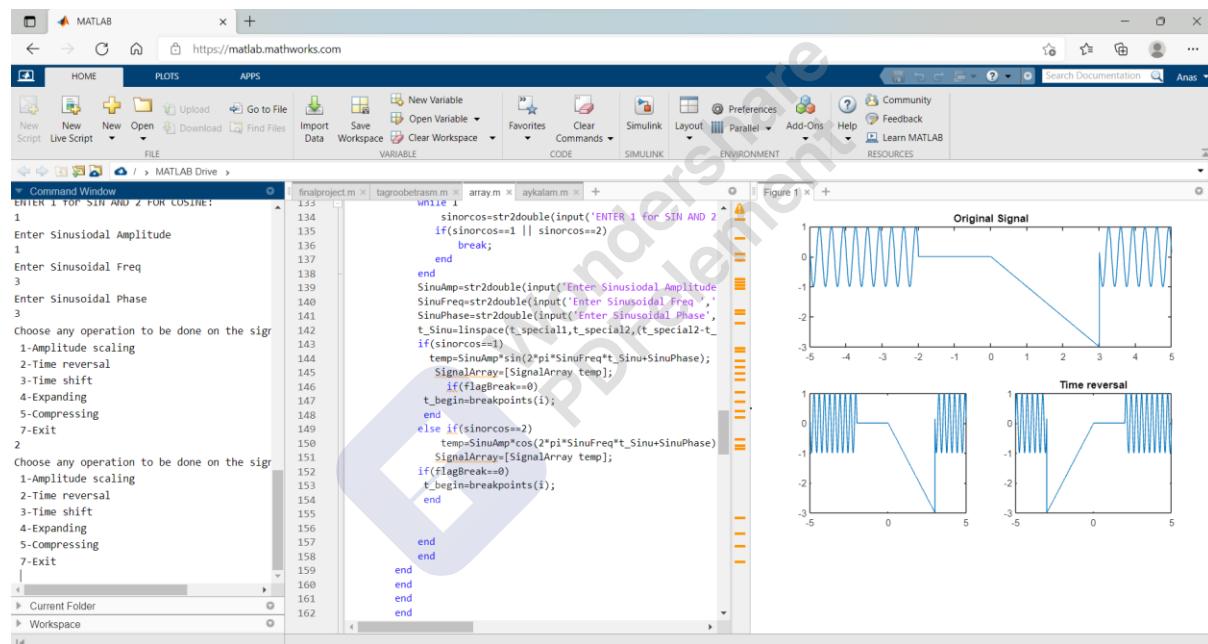
7-Exit

7

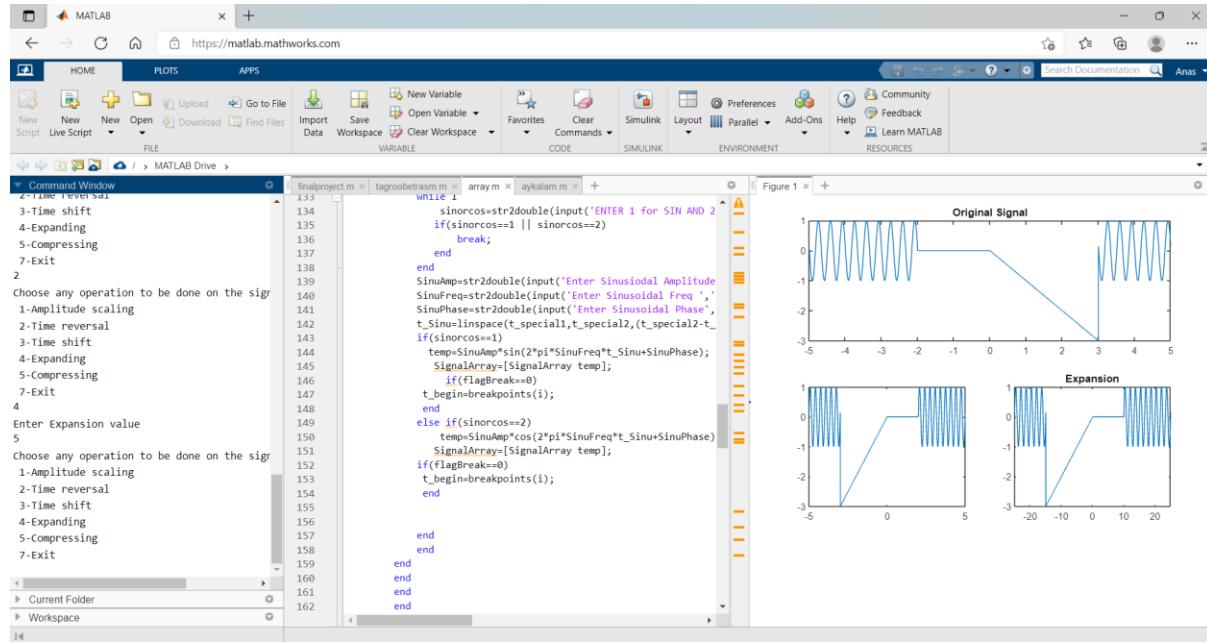
THANK YOU FOR USING SIGNAL GENERATOR

Operations

1-Time Reversal



2-Expanding by 5



EXAMPLE 4: Signal with 4 Breakpoints

1-Fs=1000

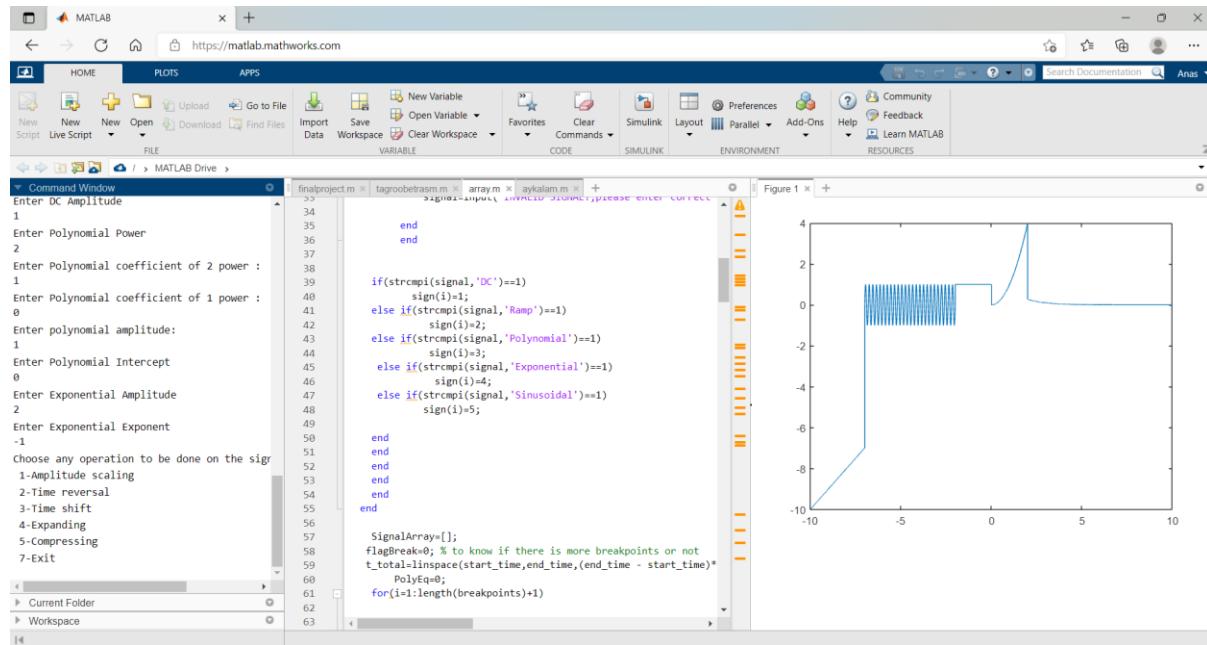
2-Start Time=-10

3-End Time=10

4-Num of breakpoints=4

5-Position of breakpoints=-7,-2,0,2,

6-Signals:Ramp,Sin,DC,Polynomial,Exponential



INPUT:

ENTER SAMPLING FREQUENCY:

1000

ENTER START TIME:

-10

ENTER END TIME:

10

ENTER NUMBER OF BREAKPOINTS:

4

ENTER POSITION OF BREAKPOINT 1

-7

ENTER POSITION OF BREAKPOINT 2

-2

ENTER POSITION OF BREAKPOINT 3

0

ENTER POSITION OF BREAKPOINT 4

2

Signal Generator Menu:

1-DC signal

2-Ramp signal

3-Polynomial Signal

4-Exponential Signal

5-Sinusoidal Signal

Enter signal name

ramp

Signal Generator Menu:

1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
sinusoidal

Signal Generator Menu:

1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
dc

Signal Generator Menu:

1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
polynomial

Signal Generator Menu:

1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
exponential

Enter Ramp slope

1

Enter Y intercept

0

ENTER 1 for SIN AND 2 FOR COSINE:

1

Enter Sinusoidal Amplitude

1

Enter Sinusoidal Freq

5

Enter Sinusoidal Phase

0

Enter DC Amplitude

1

```
Enter Polynomial Power
2
Enter Polynomial coefficient of 2 power :
1
Enter Polynomial coefficient of 1 power :
0
Enter polynomial amplitude:
1
Enter Polynomial Intercept
0
Enter Exponential Amplitude
2
Enter Exponential Exponent
-1
Choose any operation to be done on the signal,if not press 7
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
Choose any operation to be done on the signal,if not press 7
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
5
Enter Compression value
5
Choose any operation to be done on the signal,if not press 7
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
2
Choose any operation to be done on the signal,if not press 7
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
```

Operations:

1-Compressing by 5

Screenshot of MATLAB interface showing code execution and signal plots.

Command Window Output:

```

Enter Polynomial Intercept
0
Enter Exponential Amplitude
2
Enter Exponential Exponent
-1
Choose any operation to be done on the sig
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
5
Enter Compression value
5
Choose any operation to be done on the sig
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit

```

Figure 1: Displays three plots. The top plot is titled "Original Signal" and shows a signal with a sharp rise from -10 to 0 at t ≈ -7, followed by a series of small oscillations between 0 and 5 until t ≈ 0, then a larger pulse from 0 to 5 at t ≈ 2, and finally a steady state. The bottom-left plot is titled "Compression" and shows the same signal compressed horizontally, appearing as a single sharp rise from -10 to 5 at t ≈ -1. The bottom-right plot is also titled "Compression" and shows the compressed signal again.

2-Time Reversal

Screenshot of MATLAB interface showing code execution and signal plots.

Command Window Output:

```

2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
5
Enter Compression value
5
Choose any operation to be done on the sig
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit
2
Choose any operation to be done on the sig
1-Amplitude scaling
2-Time reversal
3-Time shift
4-Expanding
5-Compressing
7-Exit

```

Figure 1: Displays three plots. The top plot is titled "Original Signal" and shows a signal with a sharp rise from -10 to 0 at t ≈ -7, followed by a series of small oscillations between 0 and 5 until t ≈ 0, then a larger pulse from 0 to 5 at t ≈ 2, and finally a steady state. The bottom-left plot is titled "Time reversal" and shows the signal reversed in time, starting with a sharp rise from -10 to 5 at t ≈ -1, followed by a steady state. The bottom-right plot is also titled "Time reversal" and shows the reversed signal again.

Example 5: Signal with 5 breakpoints

1-Fs=1000

2-Start Time=-10

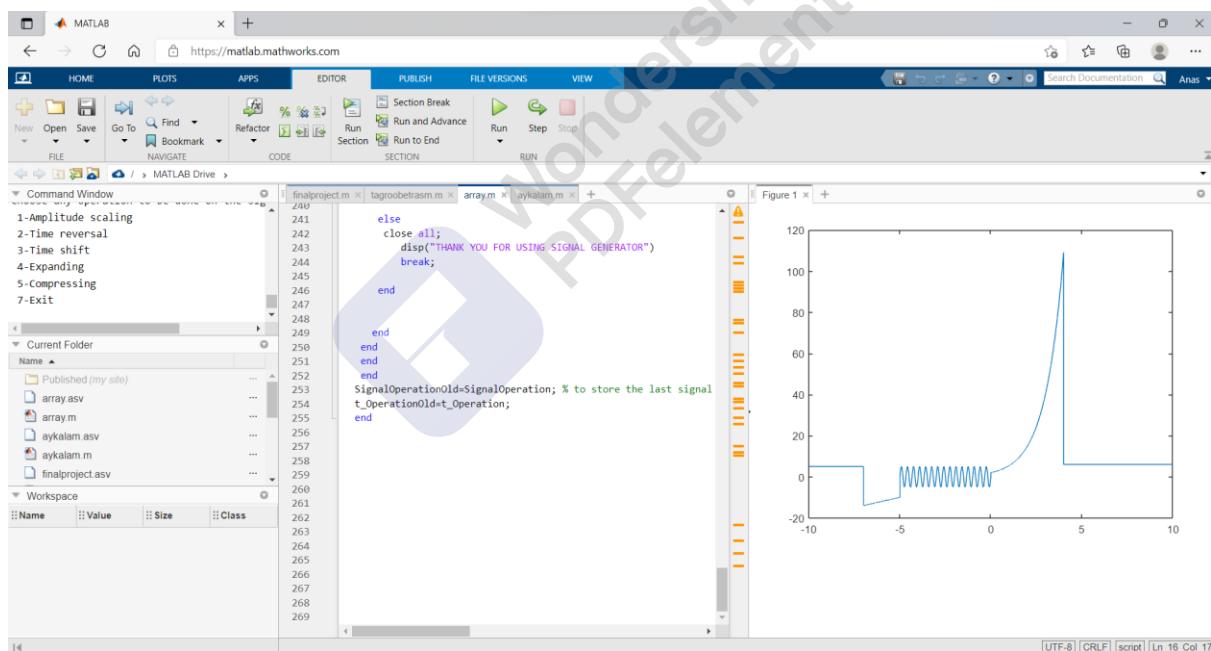
3-End Time=10

4-Num of breakpoints=5

5-Position of breakpoints=-7,-5,0,2,4

6-

Signals,DC,Ramp,Sinusoidal,Polynomial,Exponential,D
C





INPUT: ENTER SAMPLING FREQUENCY:

1000
ENTER START TIME:
-10
ENTER END TIME:
10
ENTER NUMBER OF BREAKPOINTS:
5
ENTER POSITION OF BREAKPOINT 1
-7
ENTER POSITION OF BREAKPOINT 2
-5
ENTER POSITION OF BREAKPOINT 3
0
ENTER POSITION OF BREAKPOINT 4
2
ENTER POSITION OF BREAKPOINT 5
4
Signal Generator Menu:
1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
dc
Signal Generator Menu:
1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
ramp
Signal Generator Menu:
1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
sinusoidal
Signal Generator Menu:
1-DC signal
2-Ramp signal

```
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
polynomial
Signal Generator Menu:
1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
exponential
Signal Generator Menu:
1-DC signal
2-Ramp signal
3-Polynomial Signal
4-Exponential Signal
5-Sinusoidal Signal
Enter signal name
dc
Enter DC Amplitude
5
Enter Ramp slope
2
Enter Y intercept
0
ENTER 1 FOR SIN AND 2 FOR COSINE:
1
Enter Sinusiodal Amplitude
5
Enter Sinusoidal Freq
3
Enter Sinusoidal Phase
0
Enter Polynomial Power
2
Enter Polynomial coefficient of 2 power :
1
Enter Polynomial coefficient of 1 power :
0
Enter polynomial amplitude:
1
Enter Polynomial Intercept
0
Enter Exponential Amplitude
2
```

Enter Exponential Exponent

1

Enter DC Amplitude

6

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

1

Enter Scaling value

2

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

2

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

3

Enter shift value

5

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

7-Exit

4

Enter Expansion value

6

Choose any operation to be done on the signal,if not press 7

1-Amplitude scaling

2-Time reversal

3-Time shift

4-Expanding

5-Compressing

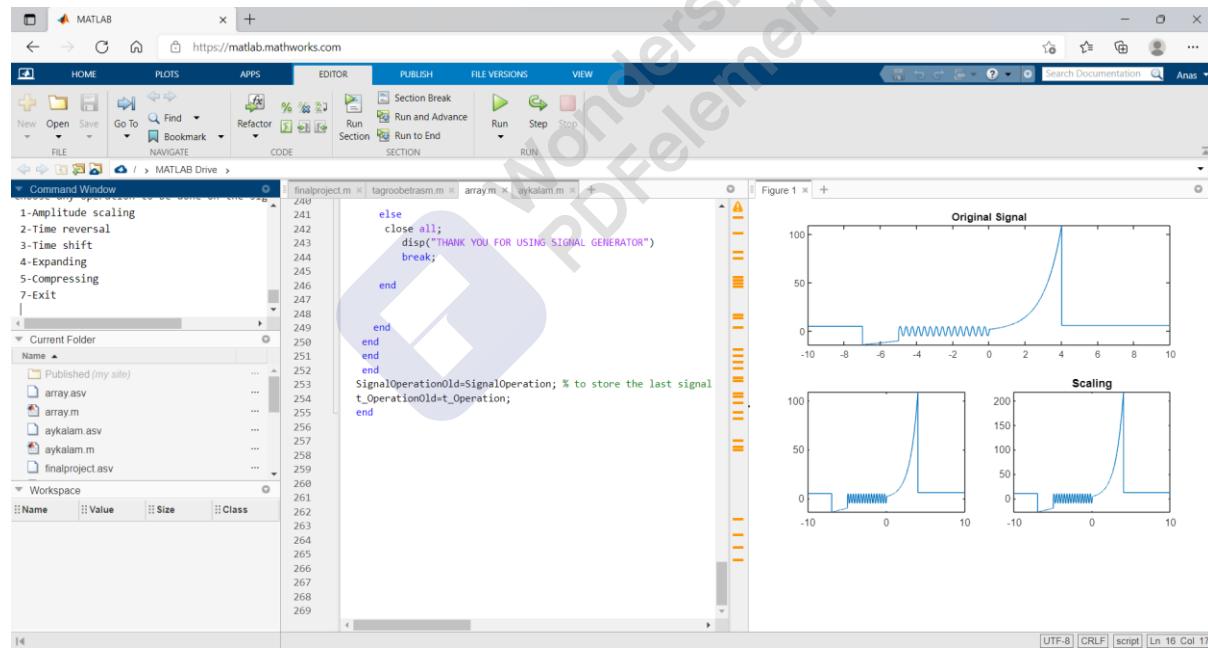
7-Exit

6

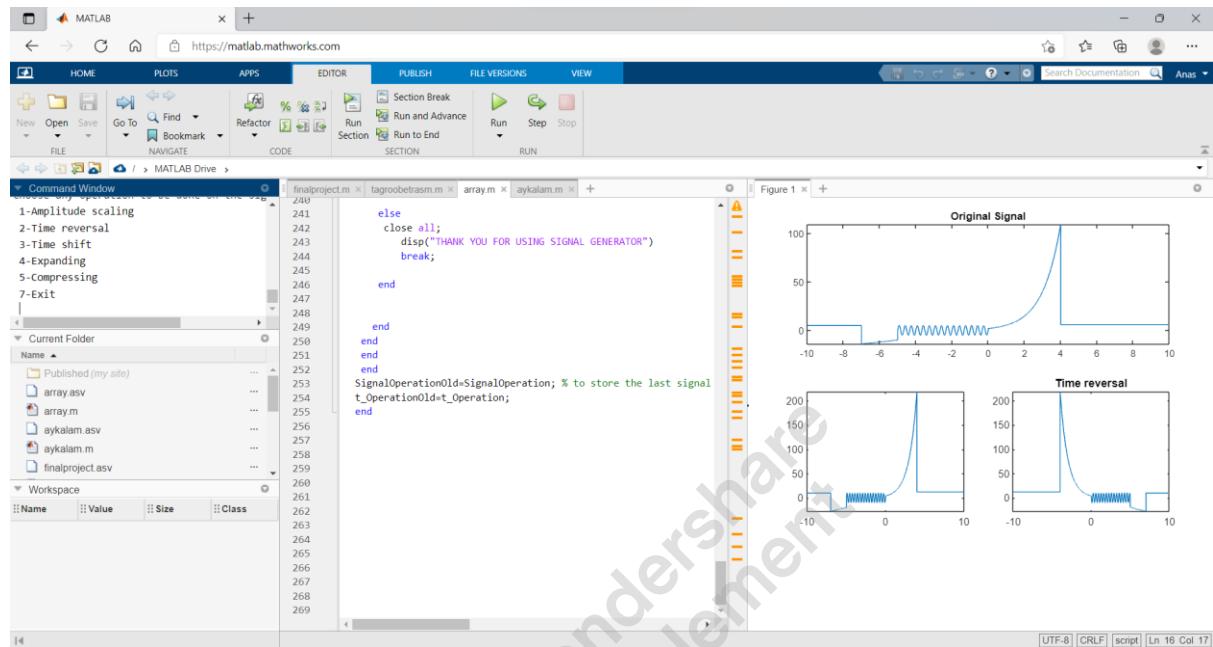
THANK YOU FOR USING SIGNAL GENERATOR

OPERATIONS

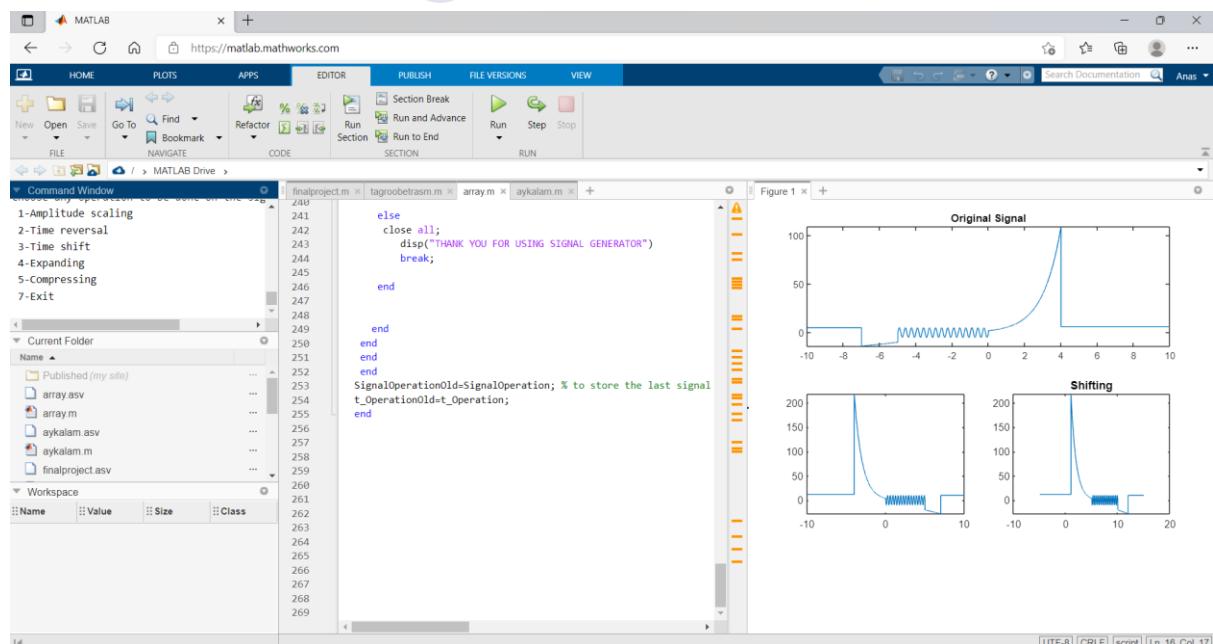
1-Ampliude Scaling by 2



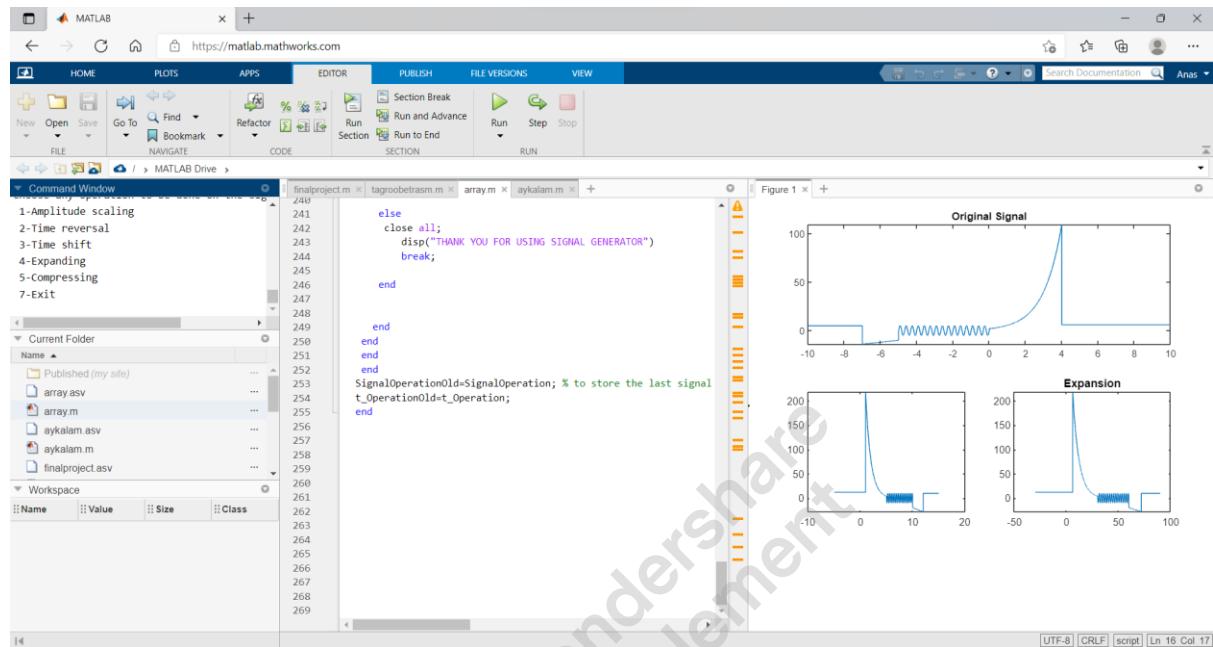
2-Time Reversal



3-Time Shift by 5



4-Expanding by 6



THANK YOU