



GENERAL SIGNAL GENERATOR

Mini project 1



- | | |
|-------------------------|----------|
| ○ Eslam Abdellatif Dyab | 18010333 |
| ○ Eslam Ashraf Muhammed | 18010324 |
| ○ Mayar Gamal Samada | 18011918 |
| ○ Mai Bendary Soliman | 18011912 |

- First of all we ask the user to enter the Sampling frequency , Start and end of the signal by using (Prompt).

```
%1
%-----%
%a) provide the sample rate
prompt = 'please provide the sample rate: ';
sample_rate = input(prompt);

%b) provide the start and end of time scale
prompt = 'please provide the start of time scale: ';
start = input(prompt);
prompt = 'please provide the end of time scale: ';
endd = input(prompt);
```

- Then we ask for the number of Break Points if the number is more than Zero we generate an array with a number of elements equal to the number of break points
- After that we ask for the position of each break point if it is between the start point and the end point we store its value in the position array if not we ask for a correct position

```
% time
t = linspace(start,endd,(endd - start)*sample_rate );

%c) provide number of the break points and positions
prompt = 'please provide number of the break points: ';
num_break_points = input(prompt);

if (num_break_points > 0)
    positions = zeros(1, num_break_points);
    fprintf('please provide the position of break points: \n');
else
    positions = [0];
end

i=1;
while( i <= num_break_points )
    position = input('');
    % being sure the position of break points is valid
    if (position < endd && position > start)
        positions(i) = position;
    else
        fprintf('please provide a correct position for break points!!: \n');
        i = i -1;
    end
    i = i +1;
end
```

→ Generating the signal

- By checking the number of break points if its greater than Zero we generate an array called times array contains start ,array of positions and end if not then it contains the start and the end only .
- Then we generate an empty array called signal
- After that we generate a while loop which make a portion signals by using a portion_signal_generator function

```
function signal = signals_generator(start,endd,sample_rate,positions,num_break_points)
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes here

if (num_break_points > 0)
    times = [start positions endd];
else
    times = [start endd];
end

signal = [];
j = 1;
while(j <= num_break_points+1)

    time = linspace(times(j),times(j+1), (times(j+1) - times(j))*sample_rate);

    prompt = 'please provide the name of portion of the signal: ';
    signal_name = input(prompt,'s');

    portion_signal = portion_signal_generator(signal_name,time);

    signal = [signal, portion_signal];

    j = j +1;
end
```

→ Portion signal generator function

- We import the signal name from signals_generator_function
- And check if it is sin , dc ,ramp, exponential or polynomial by using strcmp

➡ Portion signal generator function

```
function portion_signal = portion_signal_generator(signal_name,time)
%UNTITLED3 Summary of this function goes here
% Detailed explanation goes here

if strcmp(signal_name , 'sinusoidal')
    prompt = 'please provide the Amplitude of the signal: ';
    amp = input(prompt);

    prompt = 'please provide the Frequency of the signal: ';
    freq = input(prompt);

    prompt = 'please provide the Phase of the signal: ';
    phase = input(prompt);

    portion_signal = amp*sin(2*pi*time*freq + phase);

elseif strcmp(signal_name, 'dc')
    prompt = 'please provide the Amplitude of the signal: ';
    amp = input(prompt);
    le = length(time);
    portion_signal = amp*ones(1,le);

elseif strcmp(signal_name, 'ramp')
    prompt = 'please provide the Amplitude of the signal: ';
    amp = input(prompt);

    prompt = 'please provide the intercept of the signal: ';
    intercept = input(prompt);

    portion_signal = amp*time + intercept;

elseif strcmp(signal_name, 'exponential')
    prompt = 'please provide the Amplitude of the signal: ';
    amp = input(prompt);

    prompt = 'please provide the exponent of the signal: ';
    exponent = input(prompt);

    portion_signal = amp*exp(exponent*time);

elseif strcmp(signal_name, 'polynomial')
    prompt = 'please provide the power of the signal: ';
    power = input(prompt);
    i=1;
    fprintf('please provide the coefficients of the signal from higher to lower order:\n');
    coefficients = zeros(1, power+1);
    portion_signal = 0;
    while (i<=power+1)
        coefficients(i)=input('');
        portion_signal=portion_signal+coefficients(i)*time.^((power+1)-i);
        i=i+1;
    end
end
end
```

➡ Operation on signal function

```
function operation_on_signal(t,signal)
%UNTITLED4 Summary of this function goes here
% Detailed explanation goes here

prompt = 'Do you want to perform any operation on the signal? (yes/no)\n';
answer = input(prompt,'s');

if strcmp(answer , 'yes')
    fprintf('The operations you can perform on the signal are:\na-Amplitude Scaling, \nb-Time reversal, \n')
    prompt = 'Enter the number of operations you want to perform (from 1 to 5): ';
    num_operations = input(prompt);

    figure_num = 2;
    while (num_operations > 0)
        prompt = 'Enter the name of operations you want to perform\ntype a or b or c or d or e or f: ';
        operation = input(prompt, 's');

        %a
        if strcmp(operation, 'a')
            prompt = 'Enter the valule of amplification: ';
            amp = input(prompt);
            signal_modified = amp*signal;
            figure.figure_num
            plot(t,signal_modified)
            title('Amplified Signal')

        %b
        elseif strcmp(operation,'b')
            time = -1.*t;
            figure.figure_num
            plot(time,signal)
            title('Time Reversed Signal')

        %c
        elseif strcmp(operation,'c')
            prompt = 'enter the value of the shift: ';
            shift = input(prompt);
            time= t+shift;
            figure.figure_num
            plot(time,signal)
            title('Shifted Signal')

        %d
        elseif strcmp(operation,'d')
            prompt = 'enter the value of Expanction: ';
            expan = input(prompt);
            time= expan*t;
            figure.figure_num
            plot(time,signal)
            title('Expanded Signal')

        %e
        elseif strcmp(operation,'e')
            prompt = 'enter the value of Compresion: ';
            comp = input(prompt);
            time = t./comp;
            figure.figure_num
            plot(time,signal)
            title('Comoresed Signal')

        %f
        elseif strcmp(operation,'f')
            signal_modified = signal;
        end

        num_operations = num_operations -1;
        figure_num = figure_num +1;
    end

elseif strcmp(answer , 'no')
    signal_modified = signal;
end

end
```

- First we ask the user if he wants to make any operation on the signal if no we end the program.
- If yes we inform him the operations which can be made on the signal and ask him about the number of operations that he wants to make.
- we generated a while loop -which contains definition of each operation- to perform the operations he wants to make .
- then we ask him about the name of operation
- the function import the signal from signals_generator_function and perform the required operations on it
- then plot the modified signals

➡ Running The Code

```

Command Window
please provide the sample rate: 200
please provide the start of time scale: 0
please provide the end of time scale: 6
please provide number of the break points: 1
please provide the position of break points:
3
-----
please provide the name of portion of the signal: sinusoidal
please provide the Amplitude of the signal: 2
please provide the Frequency of the signal: 1
please provide the Phase of the signal: 0
please provide the name of portion of the signal: dc
please provide the Amplitude of the signal: 1
-----
Do you want to perform any operation on the signal? (yes/no)
yes
The operations you can perform on the signal are:
a-Amplitude Scaling,
b-Time reversal,
c-Time shift,
d-Expanding the signal,
e-Compressing the signal,
f-None

Enter the number of operations you want to perform (from 1 to 5): 5

```

Do you want to perform any operation on the signal? (yes/no)

yes

The operations you can perform on the signal are:

a-Amplitude Scaling,

b-Time reversal,

c-Time shift,

d-Expanding the signal,

e-Compressing the signal,

f-None

Enter the number of operations you want to perform (from 1 to 5): 5

Enter the name of operations you want to perform

type a or b or c or d or e or f: a

Enter the value of amplification: 5

Enter the name of operations you want to perform

type a or b or c or d or e or f: b

Enter the name of operations you want to perform

type a or b or c or d or e or f: c

enter the value of the shift: 2

Enter the name of operations you want to perform

type a or b or c or d or e or f: d

enter the value of Expansion: 2

Enter the name of operations you want to perform

type a or b or c or d or e or f: e

enter the value of Compression: 2

>> |

