

Homework 1 with Solution:

Lab 0: MATLAB Basics:

Task 1:

Use the subplot command to provide following three plots in one figure:

- Plot 1: $\sin(t)$, $\sin(3t)$, $\sin(5t)$
- Plot 2: $\cos(t)$, $\cos(3t)$, $\cos(5t)$
- Plot 3: $\tan(t)$, $\tan(3t)$, $\tan(5t)$

For $t \in [-1, 1]$. It is important to use different color schemes to differentiate the harmonics; provide legend, xlabel, ylabel, and title for each plot.

Solution

```
1  clc; close all; clear;
2  t = -1:0.001:1;
3  y11 = sin(t);  y12 = sin(3*t);  y13 = sin(5*t);
4  y21 = cos(t);  y22 = cos(3*t);  y23 = cos(5*t);
5  y31 = tan(t);  y32 = tan(3*t);  y33 = tan(5*t);
6  figure;
7  subplot(3,1,1);
8  plot(t, y11, t, y12, t, y13);
9  xlabel('time, t');
10 ylabel('Amplitude, y');
11 title('Sine Plots');
12 legend('sin(t)', 'sin(3t)', 'sin(5t)');
13
14 subplot(3,1,2);
15 plot(t, y21, t, y22, t, y23);
16 xlabel('time, t');
17 ylabel('Amplitude, y');
18 title('Cosine Plots');
19 legend('cos(t)', 'cos(3t)', 'cos(5t)');
20
21 subplot(3,1,3);
22 plot(t, y31, t, y32, t, y33);
23 xlabel('time, t');
24 ylabel('Amplitude, y');
25 title('Tan Plots');
26 legend('tan(t)', 'tan(3t)', 'tan(5t)');
```

Task 2:

Write a MATLAB function, which takes the following two inputs:

A vector (of variable length) representing resistors in an electrical circuit A flag representing whether resistors are in parallel or series The function should return equivalent resistance.

Solution:

```
1
2 function Req = EquivalentResistance(R, inParallel)
3
4 if inParallel ~=0
5     Req = sum(R);
6 else
7     Req = 1/(sum(1./R));
8 end
9 end
```

Task 3:

Define the symbolic variable x. Make use of this variable to define the symbolic function sin(1/x). Then, plot the function from 0 to 2. What do you see? Why might the plotting routine be having trouble plotting this function?

Solution:

Plotting symbolic function with plot function is not possible, because, plot function is defined for numerical values. However, plot can be used when we substitute numerical value like:

```
1 clc; close all; clear;
2 syms x
3 f(x) = sin(1/x);
4 x = 0.001:0.001:2;
5 y = double(subs(f));
6 plot(x, y)
```

Task 4:

The natural response of a series RC circuit is given by:

$$v(t) = V_0 e^{-\frac{t}{RC}}$$

Write a MATLAB function that takes two inputs R and C, and plots the natural response of this circuit for $t \in [0, 2]$.

Solution:

```
1
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
2 function naturalResponseRC(R, C)
3
4 V0 = 10;
5 t = 0:0.01:2;
6 v = V0*exp(-t/(R*C));
7 plot(v,t);
8 xlabel('time, t (sec)');
9 ylabel('Voltage, v (volts)');
10 title('Natural Response of RC Circuit');
11
12 end
```

Lab. 1 Task with Solution: Signals & Systems Properties

Task 1:

Accumulator or summer is a discrete time system with memory, given by:

$$y[n] = \sum_{k=-\infty}^n x[k]$$

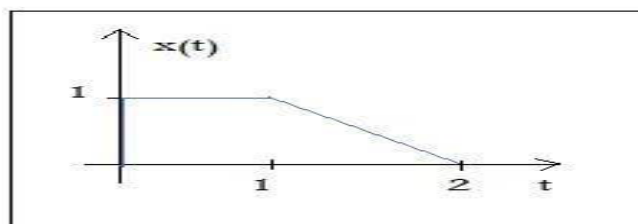
Write a MATLAB Script for the accumulator.

Solution

```
1 clc; close all; clear;
2
3 x = [1 2 3 4 5 4 3 2 1];
4 y = zeros(size(x));
5 for n = 1:length(x)
6     y(n) = sum(x(1:n));
7 end
8
9 figure;
10 subplot(2,1,1);
11 stem(x);
12 xlabel('n');
13 ylabel('x[n]');
14 title('Input to Accumulator');
15
16 subplot(2,1,2);
17 stem(y);
18 xlabel('n');
19 ylabel('y[n]');
20 title('Output of Accumulator');
```

Task 2:

Consider the following figure:



Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

Plot the following in MATLAB:

- a) $x(t+1)$
- b) $x(-t+1)$
- c) $x\left(\frac{3}{2}t\right)$
- d) $x\left(\frac{3}{2}t-1\right)$

Solution:

```
1  clc; close all; clear;
2
3  t = 0:0.01:2;
4  x = 1*(t<=1) + (2-t).*(t>1);
5
6  figure;
7  plot(t, x);
8  xlabel('t');
9  ylabel('x');
10 title('Original, x(t)')
11
12 figure;
13 plot(t-1, x);
14 xlabel('t');
15 ylabel('x');
16 title('x(t+1)')
17
18 figure;
19 plot(-t+1, x);
20 xlabel('t');
21 ylabel('x');
22 title('x(-t+1)')
23
24 figure;
25 plot(t/(3/2), x);
26 xlabel('t');
27 ylabel('x');
28 title('x(3t/2)')
29
30 figure;
31 plot(t/(3/2)+1, x);
32 xlabel('t');
33 ylabel('x');
34 title('x(3t/2 - 1)')
35
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

Task 3:

Analyze each of the following continuous-time systems in terms of the six basic system properties. In addition, summarize your conclusions in the table below:

- a) $S_1(t) = x(t - 2) + x(2 - t)$
- b) $S_2(t) = \cos(3t)x(t)$
- c) $S_3(t) = \begin{cases} 0 & t < 0 \\ x_1(t) + x(t - 2) & t \geq 0 \end{cases}$

Solution:

Systems	Memory	Causality	Invertibility	Stability	Time invariance	Linearity
S_1	Yes	No	Yes	No	Yes	Yes
S_2	No	Yes	Yes	Yes	No	No
S_3	Yes	Yes	No	Yes	Yes	No

Task 4:

Record your own sound saying the sentence, "The quick brown fox jumps over the lazy dog". Play this recording in MATLAB with and without time reversal, also comment on the observations.

Solution:

```
1 clc; close all; clear;
2
3 recObj = audiorecorder
4
5 disp('Start speaking. ');
6 recordblocking(recObj, 5);
7 disp('End of Recording. ');
8
9 play(recObj);
10
11 y = getaudiodata(recObj);
12
13 subplot(2,1,1)
14 plot(y);
15 title('Voice without Time Reversal')
16
17 subplot(2,1,2)
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
18 plot(y(end:-1:1));  
19 title('Voice with Time Reversal')
```

Lab. 2 Tasks with Solution: Signals, Periodicity and Harmonics

Task 1:

Compute Odd and Even part of a step function for $n \in [-5, 5]$. A step function is given by:

$$u[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

Soution:

```
1  clc; close all; clear;
2
3  n = -5:5;
4  u = double(n>=0);
5
6  u_ = u(end:-1:1);
7
8  u_even = 1/2*(u + u_);
9  u_odd = 1/2*(u - u_);
10
11 figure;
12 subplot(3,1,1)
13 stem(n, u);
14 xlabel('n');
15 ylabel('u');
16 title('Step Function');
17
18 subplot(3,1,2)
19 stem(n, u_even);
20 xlabel('n');
21 ylabel('u_even');
22 title('Even Component of Step Function');
23
24 subplot(3,1,3)
25 stem(n, u_odd);
26 xlabel('n');
27 ylabel('u_odd');
28 title('Odd Component of Step Function');
```

Task 2:

Plot 5 odd harmonics of cosine for $t \in [-10, 10]$ given by:

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

$$x(t) = \frac{4}{\pi} \times \left\{ \cos t - \frac{1}{3} \cos 3t + \frac{1}{5} \cos 5t - \frac{1}{7} \cos 7t + \frac{1}{9} \cos 9t \right\}$$

Solution:

```
clc; close all; clear;
```

```
t = -10:0.01:10;
```

```
y1 = cos(t);
```

```
y2 = -1/3*cos(3*t);
```

```
y3 = 1/5*cos(5*t);
```

```
y4 = -1/7*cos(7*t);
```

```
y5 = 1/9*cos(9*t);
```

```
y = (y1+y2+y3+y4+y5)*4/pi;
```

```
plot(t,y);
```

```
xlabel('t');
```

```
ylabel('y');
```

```
title('5 Odd Harmonics of Cosine');
```

Homework 2 Task with Solution

Homework 2:

Plot 100 Odd harmonics of Cosine using:

$$x(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^{k+1} \cos((2k-1)t)}{2k-1}$$

For $t \in [-10, 10]$

Solution:

```
1  clc; close all; clear;
2
3  t = -10:0.01:10;
4  y = zeros(size(t));
5  for k = 1:100
6      y = y + (-1)^(k+1)*cos((2*k-1)*t)/(2*k - 1);
7  end
8  plot(t,y);
9  xlabel('t');
10 ylabel('y');
11 title('Sum of 100 Odd Harmonics of Cosine');
```

Lab 3 Task with Solution: Graphical User Interface (GUI)

Task 1:

Design a MATLAB GUI that takes two numbers and prints the sum when a pushbutton is pressed. Give meaningful names and tags to GUI component.

Solution

In the following code, `add_Callback()` function (line: 138-148) correspond to the pushbutton with Tag 'add'. 'handles.n1' and 'handles.n2' correspond to the edit fields while 'handles.sum' corresponds the static text field where final sum is shown to the user.

Please consider the highlighted piece of code.

```
1 function varargout = myAdder(varargin)
2 % MYADDER MATLAB code for myAdder.fig
3 %     MYADDER, by itself, creates a new MYADDER or raises the
4 %     existing
5 %     singleton*.
6 %
7 %     H = MYADDER returns the handle to a new MYADDER or the handle
8 %     to
9 %     the existing singleton*.
10 %
11 %     MYADDER('CALLBACK',hObject,eventData,handles,...) calls the
12 %     local
13 %     function named CALLBACK in MYADDER.M with the given input
14 %     arguments.
15 %
16 %     MYADDER('Property','Value',...) creates a new MYADDER or raises
17 %     the
18 %     existing singleton*. Starting from the left, property value
19 %     pairs are
20 %     applied to the GUI before myAdder_OpeningFcn gets called. An
21 %     unrecognized property name or invalid value makes property
22 %     application
23 %     stop. All inputs are passed to myAdder_OpeningFcn via
24 %     varargin.
25 %
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
26 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
27 only one
28 % instance to run (singleton)".
29 %
30 % See also: GUIDE, GUIDATA, GUIHANDLES
31
32 % Edit the above text to modify the response to help myAdder
33
34 % Last Modified by GUIDE v2.5 04-Nov-2021 03:12:22
35
36 % Begin initialization code - DO NOT EDIT
37 gui_Singleton = 1;
38 gui_State = struct('gui_Name',       mfilename, ...
39                   'gui_Singleton',   gui_Singleton, ...
40                   'gui_OpeningFcn',   @myAdder_OpeningFcn, ...
41                   'gui_OutputFcn',    @myAdder_OutputFcn, ...
42                   'gui_LayoutFcn',    [] , ...
43                   'gui_Callback',     []);
44 if nargin && ischar(varargin{1})
45     gui_State.gui_Callback = str2func(varargin{1});
46 end
47
48 if nargout
49     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
50 else
51     gui_mainfcn(gui_State, varargin{:});
52 end
53 % End initialization code - DO NOT EDIT
54
55
56 % --- Executes just before myAdder is made visible.
57 function myAdder_OpeningFcn(hObject, eventdata, handles, varargin)
58 % This function has no output args, see OutputFcn.
59 % hObject    handle to figure
60 % eventdata  reserved - to be defined in a future version of MATLAB
61 % handles    structure with handles and user data (see GUIDATA)
62 % varargin   command line arguments to myAdder (see VARARGIN)
63
64 % Choose default command line output for myAdder
65 handles.output = hObject;
66
67 % Update handles structure
68 guidata(hObject, handles);
69
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
70 % UIWAIT makes myAdder wait for user response (see UIRESUME)
71 % uiwait(handles.figure1);
72
73
74 % --- Outputs from this function are returned to the command line.
75 function varargout = myAdder_OutputFcn(hObject, eventdata, handles)
76 % varargout cell array for returning output args (see VARARGOUT);
77 % hObject handle to figure
78 % eventdata reserved - to be defined in a future version of MATLAB
79 % handles structure with handles and user data (see GUIDATA)
80
81 % Get default command line output from handles structure
82 varargout{1} = handles.output;
83
84
85
86 function n1_Callback(hObject, eventdata, handles)
87 % hObject handle to n1 (see GCBO)
88 % eventdata reserved - to be defined in a future version of MATLAB
89 % handles structure with handles and user data (see GUIDATA)
90
91 % Hints: get(hObject,'String') returns contents of n1 as text
92 % str2double(get(hObject,'String')) returns contents of n1 as a
93 double
94
95
96 % --- Executes during object creation, after setting all properties.
97 function n1_CreateFcn(hObject, eventdata, handles)
98 % hObject handle to n1 (see GCBO)
99 % eventdata reserved - to be defined in a future version of MATLAB
100 % handles empty - handles not created until after all CreateFcns
101 called
102
103 % Hint: edit controls usually have a white background on Windows.
104 % See ISPC and COMPUTER.
105 if ispc && isequal(get(hObject,'BackgroundColor'),
106 get(0,'defaultUicontrolBackgroundColor'))
107 set(hObject,'BackgroundColor','white');
108 end
109
110
111
112 function n2_Callback(hObject, eventdata, handles)
113 % hObject handle to n2 (see GCBO)
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
114 % eventdata reserved - to be defined in a future version of MATLAB
115 % handles structure with handles and user data (see GUIDATA)
116
117 % Hints: get(hObject,'String') returns contents of n2 as text
118 % str2double(get(hObject,'String')) returns contents of n2 as a
119 double
120
121
122 % --- Executes during object creation, after setting all properties.
123 function n2_CreateFcn(hObject, eventdata, handles)
124 % hObject handle to n2 (see GCBO)
125 % eventdata reserved - to be defined in a future version of MATLAB
126 % handles empty - handles not created until after all CreateFcns
127 called
128
129 % Hint: edit controls usually have a white background on Windows.
130 % See ISPC and COMPUTER.
131 if ispc && isequal(get(hObject,'BackgroundColor'),
132 get(0,'defaultUicontrolBackgroundColor'))
133 set(hObject,'BackgroundColor','white');
134 end
135
136
137 % --- Executes on button press in add.
138 function add_Callback(hObject, eventdata, handles)
139 % hObject handle to add (see GCBO)
140 % eventdata reserved - to be defined in a future version of MATLAB
141 % handles structure with handles and user data (see GUIDATA)
142 n1 = str2double(get(handles.n1, 'String'));
143 n2 = str2double(get(handles.n2, 'String'));
144 disp(n1)
145 disp(n2)
146 sm = n1+n2;
147 set(handles.sum, 'String', num2str(sm))
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

Task 2:

Write a MATLAB Code that takes n and m from the user and plots n numbers of Odd Cosine Harmonics for the time 0 to m seconds when pushbutton 'PLOT' is pressed.

n: No. of Harmonics

m: No. of Seconds

Solution:

Fs

```
1 function varargout = myHarmonicPlot(varargin)
2 % MYHARMONICPLOT MATLAB code for myHarmonicPlot.fig
3 %     MYHARMONICPLOT, by itself, creates a new MYHARMONICPLOT or
4 %     raises the existing
5 %     singleton*.
6 %
7 %     H = MYHARMONICPLOT returns the handle to a new MYHARMONICPLOT
8 %     or the handle to
9 %     the existing singleton*.
10 %
11 %     MYHARMONICPLOT('CALLBACK',hObject,eventData,handles,...) calls
12 %     the local
13 %     function named CALLBACK in MYHARMONICPLOT.M with the given
14 %     input arguments.
15 %
16 %     MYHARMONICPLOT('Property','Value',...) creates a new
17 %     MYHARMONICPLOT or raises the
18 %     existing singleton*. Starting from the left, property value
19 %     pairs are
20 %     applied to the GUI before myHarmonicPlot_OpeningFcn gets
21 %     called. An
22 %     unrecognized property name or invalid value makes property
23 %     application
24 %     stop. All inputs are passed to myHarmonicPlot_OpeningFcn via
25 %     varargin.
26 %
27 %     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
28 %     only one
29 %     instance to run (singleton)".
30 %
31 % See also: GUIDE, GUIDATA, GUIHANDLES
32
33 % Edit the above text to modify the response to help myHarmonicPlot
34
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
35 % Last Modified by GUIDE v2.5 04-Nov-2021 04:04:10
36
37 % Begin initialization code - DO NOT EDIT
38 gui_Singleton = 1;
39 gui_State = struct('gui_Name',       mfilename, ...
40                   'gui_Singleton',   gui_Singleton, ...
41                   'gui_OpeningFcn',   @myHarmonicPlot_OpeningFcn, ...
42                   'gui_OutputFcn',    @myHarmonicPlot_OutputFcn, ...
43                   'gui_LayoutFcn',    [], ...
44                   'gui_Callback',     []);
45 if nargin && ischar(varargin{1})
46     gui_State.gui_Callback = str2func(varargin{1});
47 end
48
49 if nargout
50     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
51 else
52     gui_mainfcn(gui_State, varargin{:});
53 end
54 % End initialization code - DO NOT EDIT
55
56
57 % --- Executes just before myHarmonicPlot is made visible.
58 function myHarmonicPlot_OpeningFcn(hObject, eventdata, handles,
59 varargin)
60 % This function has no output args, see OutputFcn.
61 % hObject    handle to figure
62 % eventdata  reserved - to be defined in a future version of MATLAB
63 % handles    structure with handles and user data (see GUIDATA)
64 % varargin   command line arguments to myHarmonicPlot (see VARARGIN)
65
66 % Choose default command line output for myHarmonicPlot
67 handles.output = hObject;
68
69 % Update handles structure
70 guidata(hObject, handles);
71
72 % UIWAIT makes myHarmonicPlot wait for user response (see UIRESUME)
73 % uiwait(handles.figure1);
74
75
76 % --- Outputs from this function are returned to the command line.
77 function varargout = myHarmonicPlot_OutputFcn(hObject, eventdata,
78 handles)
```


Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
79 % varargout    cell array for returning output args (see VARARGOUT);
80 % hObject      handle to figure
81 % eventdata    reserved - to be defined in a future version of MATLAB
82 % handles       structure with handles and user data (see GUIDATA)
83
84 % Get default command line output from handles structure
85 varargout{1} = handles.output;
86
87
88
89 function m_Callback(hObject, eventdata, handles)
90 % hObject      handle to m (see GCBO)
91 % eventdata    reserved - to be defined in a future version of MATLAB
92 % handles       structure with handles and user data (see GUIDATA)
93
94 % Hints: get(hObject,'String') returns contents of m as text
95 %         str2double(get(hObject,'String')) returns contents of m as a
96 %         double
97
98
99 % --- Executes during object creation, after setting all properties.
100 function m_CreateFcn(hObject, eventdata, handles)
101 % hObject      handle to m (see GCBO)
102 % eventdata    reserved - to be defined in a future version of MATLAB
103 % handles       empty - handles not created until after all CreateFcns
104 % called
105
106 % Hint: edit controls usually have a white background on Windows.
107 %       See ISPC and COMPUTER.
108 if ispc && isequal(get(hObject,'BackgroundColor'),
109 get(0,'defaultUicontrolBackgroundColor'))
110     set(hObject,'BackgroundColor','white');
111 end
112
113
114
115 function n_Callback(hObject, eventdata, handles)
116 % hObject      handle to n (see GCBO)
117 % eventdata    reserved - to be defined in a future version of MATLAB
118 % handles       structure with handles and user data (see GUIDATA)
119
120 % Hints: get(hObject,'String') returns contents of n as text
121 %         str2double(get(hObject,'String')) returns contents of n as a
122 %         double
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

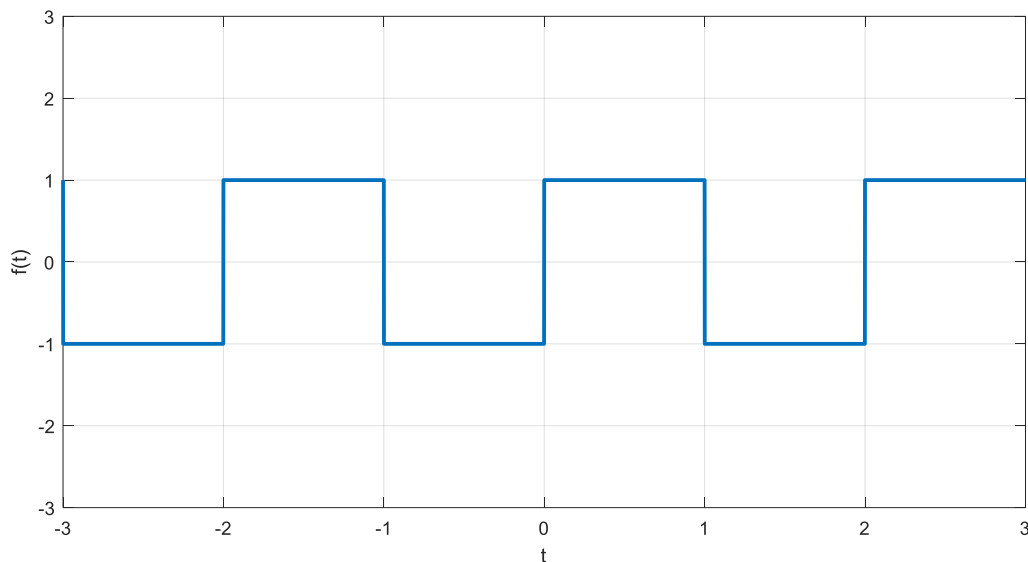
Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
123
124
125 % --- Executes during object creation, after setting all properties.
126 function n_CreateFcn(hObject, eventdata, handles)
127 % hObject    handle to n (see GCBO)
128 % eventdata  reserved - to be defined in a future version of MATLAB
129 % handles    empty - handles not created until after all CreateFcns
130 called
131
132 % Hint: edit controls usually have a white background on Windows.
133 %       See ISPC and COMPUTER.
134 if ispc && isequal(get(hObject,'BackgroundColor'),
135 get(0,'defaultUicontrolBackgroundColor'))
136     set(hObject,'BackgroundColor','white');
137 end
138
139
140 % --- Executes on button press in plot.
141 function plot_Callback(hObject, eventdata, handles)
142 % hObject    handle to plot (see GCBO)
143 % eventdata  reserved - to be defined in a future version of MATLAB
144 % handles    structure with handles and user data (see GUIDATA)
145 n = str2double(get(handles.n, 'String'));
146 m = str2double(get(handles.m, 'String'));
147
148 t = 0:0.01:m;
149 y = zeros(size(t));
150
151 for k = 1:n
152     y = y + (-1)^(k+1)*cos((2*k-1)*t)/(2*k - 1);
153 end
154 plot(handles.axes1,t,y);
155 xlabel(handles.axes1, 't');
156 ylabel(handles.axes1, 'y');
```

Lab 4 Tasks & Solution: Fourier Series

Task 1:

Plot $f(t)$ for 100 harmonics using Fourier Series:



where

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nw_0 t) + b_n \sin(nw_0 t)$$

$$a_0 = \int_{-\frac{T_0}{2}}^{+\frac{T_0}{2}} f(t) dt$$

$$a_n = \int_{-\frac{T_0}{2}}^{+\frac{T_0}{2}} f(t) \cos(nw_0 t) dt$$

$$b_n = \int_{-\frac{T_0}{2}}^{+\frac{T_0}{2}} f(t) \sin(nw_0 t) dt$$

$$w_0 = \frac{2\pi}{T_0}$$

Solution:

In this case, we have $T_0 = 2 \Rightarrow w_0 = \pi$. Consider MATLAB code below:

Signal & Systems Lab - CS341L - Fall2021

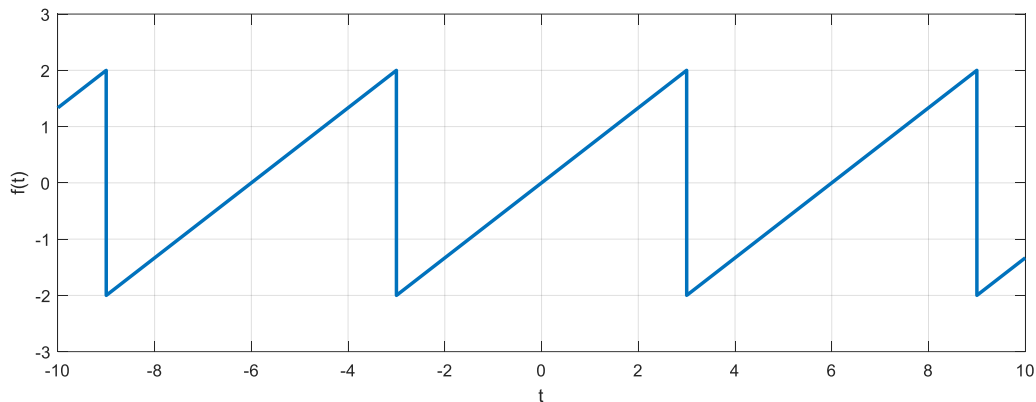
Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
1  clc; close all; clear;
2
3  T0 = 2;
4  w0 = 2*pi/T0;
5
6  syms f1(t) f2(t) n t;
7
8  f1(t) = -1;
9  f2(t) = 1;
10 a0 = int(f1, [-T0/2 0]) + int(f2, [0 T0/2]);
11 an = int(f1*cos(n*w0*t), [-T0/2 0]) + int(f2*cos(n*w0*t), [0 T0/2]);
12 bn = int(f1*sin(n*w0*t), [-T0/2 0]) + int(f2*sin(n*w0*t), [0 T0/2]);
13
14 % syms is only used for calculating integral.
15 % converting from syms to numerical, as:
16
17 t = -5:0.001:5;
18 n = 1:100;
19
20 an = double(subs(an));
21 bn = double(subs(bn));
22
23 if isscalar(an)
24     an = an*ones(size(t));
25 end
26 if isscalar(bn)
27     bn=bn*ones(size(t));
28 end
29
30
31 y = zeros(size(t));
32
33 for idx = n
34     y = y + an(idx)*cos(idx*w0*t) + bn(idx)*sin(idx*w0*t);
35 end
36 y = a0 + y;
37
38 plot(t, y);
39 xlabel('t');
40 ylabel('f(t)');
41 title('Sum of First 100 Harmonics of f(t)');
```

Task 2:

Plot $f(t)$ for 100 harmonics using Fourier Series:



$$f(t) = \frac{2}{3}t \quad \text{for} \quad -3 < t \leq 3$$

Solution:

```
clc; close all; clear;
```

```
T0 = 6;  
w0 = 2*pi/T0;
```

```
syms f(t) n t;
```

```
f(t) = 2/3*t;  
a0 = 1/T0*int(f, [-T0/2 T0/2]);  
an = 2/T0*int(f*cos(n*w0*t), [-T0/2 T0/2]);  
bn = 2/T0*int(f*sin(n*w0*t), [-T0/2 T0/2]);
```

```
% syms is only used for calculating integral.  
% converting from syms to numerical, as:
```

```
t = -5:0.001:5;  
n = 1:100;
```

```
an = double(subs(an));  
bn = double(subs(bn));
```

```
if isscalar(an)
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

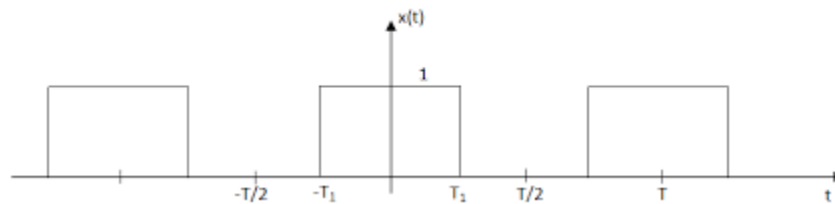
Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
    an = an*ones(size(t));  
end  
if isscalar(bn)  
    bn=bn*ones(size(t));  
end  
  
y = zeros(size(t));  
  
for idx = n  
    y = y + an(idx)*cos(idx*w0*t) + bn(idx)*sin(idx*w0*t);  
end  
y = a0 + y;  
  
plot(t, y);  
xlabel('t');  
ylabel('f(t)');  
title('Sum of First 100 Harmonics of f(t)');
```

Lab 5 Tasks & Solution: Fourier Series & Fourier Transform:

Task 1:

Find the effect of time period on Fourier Series Coefficients a_0 , a_n & b_n for the following figure having $T_1 = 5$ and observe the effect when $T_0 \rightarrow \infty$.



Solution:

Suppose we observe 5th harmonic:

```
1  clc; close all; clear;
2
3  T1 = 5;
4  syms f(t) n t T0;
5  f(t) = 1;
6  a0 = 1/T0*int(f, [-T1 T1]);
7  an = 2/T0*int(f*cos(n*2*pi*t/T0), [-T1 T1]);
8  bn = 2/T0*int(f*sin(n*2*pi*t/T0), [-T1 T1]);
9
10 % for fifth harmonic, put = 5
11 n = 5;
12
13 T0 = T1+0.01:0.01:50;
14 plot(T0, subs(an));
15 xlabel('T_0');
16 ylabel('a_n');
17 title('a_n vs T_0');
18
19 figure, plot(T0, subs(a0));
20 xlabel('T_0');
21 ylabel('a_0');
22 title('a_0 vs T_0');
23
24 figure, plot(T0, subs(bn));
25 xlabel('T_0');
```

```
26 ylabel('b_n');  
27 title('b_n vs T_0');
```

Task 2:

Write Fourier Transform (Magnitude and Phase plots) of the following signal for the following discrete time signal.

- i. $x[n] = [1, 2, 3, 3, 2, 1]$
- ii. $x(t) = e^{-2t}u(t)$ for $-10 \leq t \leq 10$

Solution:

```
1  clc; close all; clear;  
2  % Part i:  
3  x = [1 2 3 3 2 1];  
4  X = fftshift(fft(x));  
5  figure, subplot(121)  
6  plot(abs(X))  
7  xlabel('DFT Samples')  
8  ylabel('Magnitude')  
9  title('Magnitude plot')  
10 subplot(122)  
11 plot(angle(X))  
12 xlabel('DFT Samples')  
13 ylabel('Phase (Radians)')  
14 title('Phase plot')  
15 % Part ii:  
16 t = -10:0.01:10;  
17 x = exp(-2*t);  
18 x(t <= 0) = 0;  
19 X = fftshift(fft(x));  
20 figure, subplot(121)  
21 plot(abs(X))  
22 xlabel('DFT Samples')  
23 ylabel('Magnitude')  
24 title('Magnitude plot')  
25 subplot(122)  
26 plot(angle(X))  
27 xlabel('DFT Samples')  
28 ylabel('Phase (Radians)')  
29 title('Phase plot')
```


Lab 6 Tasks & Solution: Fourier Transform:

Task 1:

Record 'Hello World' speech, evaluate and plot speech signal in frequency domain and then find the range of human voice frequency.

Note: Convert the audio into monophonic if it is a stereo sound.

Solution:

```
1 clc; close all; clear;
2
3 [y, fs] = audioread('myfile.mp4');
4
5 % Check if the audio is stereophonic, then convert into monophonic
6 If size(y,2) == 2
7     Y = mean(y,2);
8 end
9
10 Y = fftshift(fft(t));
11 f = fs*linspace(-0.5,0.5,length(Y));
12 % Convert the frequency vector into column vector
13 f = f(:)';
14 figure, subplot(121)
15 plot(f, abs(Y))
16 xlabel('frequency (Hz)')
17 ylabel('Magnitude')
18 title('Magnitude plot')
19 subplot(122)
20 plot(f, angle(Y))
21 xlabel('frequency (Hz)')
22 ylabel('Phase (Radians)')
23 title('Phase plot')
```

Task 2:

Do Task 1 for normalized frequency i.e. range $[-\pi, \pi]$ rad/sample.

Solution:

```
1 clc; close all; clear;
2 [y, fs] = audioread('myfile.mp4');
3
4 % Check if the audio is stereophonic, then convert into monophonic
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

```
5 If size(y,2) == 2
6     Y = mean(y,2);
7 end
8
9 Y = fftshift(fft(t));
10 f = fs*linspace(-pi, pi, length(Y));
11 % Convert the frequency vector into column vector
12 f = f(:)';
13 figure, subplot(121)
14 plot(f, abs(Y))
15 xlabel('frequency (rad/sample)')
16 ylabel('Magnitude')
17 title('Magnitude plot')
18 subplot(122)
19 plot(f, angle(Y))
20 xlabel('frequency (rad/sample)')
21 ylabel('Phase (Radians)')
22 title('Phase plot')
```

Lab 7 Tasks & Solution:

Nyquist's Criterion & Aliasing:

Task 1:

Define a time vector for a sinusoid, keep changing the sampling frequency and observe the spectrum for Nyquist's Criterion violation that results aliasing.

Solution:

```
1  clc; close all; clear;
2  f = 10; % in Hz
3  fs = 80; % in samples/sec
4  t = 0:1/fs:10;
5  x = sin(2*pi*f*t);
6  X = fftshift(fft(x));
7  f = fs*linspace(-0.5,0.5,length(X))
8  figure, plot(f, abs(X));
9  xlabel('f (Hz)');
10 ylabel('Magnitude');
11 f = 10; % in Hz
12 fs = 30; % in samples/sec
13 t = 0:1/fs:10;
14 x = sin(2*pi*f*t);
15 X = fftshift(fft(x));
16 f = fs*linspace(-0.5,0.5,length(X))
17 figure, plot(f, abs(X));
18 xlabel('f (Hz)');
19 ylabel('Magnitude');
20 f = 10; % in Hz
21 fs = 15; % in samples/sec
22 t = 0:1/fs:10;
23 x = sin(2*pi*f*t);
24 X = fftshift(fft(x));
25 f = fs*linspace(-0.5,0.5,length(X))
26 figure, plot(f, abs(X));
27 xlabel('f (Hz)');
28 ylabel('Magnitude');
29 % in Figure 3, folding/aliasing in the spectrum is obvious
30 % as it gives peak on 5 Hz instead of 10Hz
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

Task 2:

Read a sound file and store the sampling rate f_s , convert into monophonic if required, keep every fourth sample in time domain. After doing so, the sampling rate becomes $\frac{1}{4}f_s$.

Play the original sound with sampling rate f_s and the resampled sound with sampling rate $\frac{1}{4}f_s$ and notice the difference. Also, visualize the magnitude spectrum plot and see if there is aliasing effect.

Solution:

```
1  clc; close all; clear;
2  load handel.mat;
3  yres = y(1:4:end);
4  Y = fftshift(fft(y));
5  Yres = fftshift(fft(yres));
6  f = Fs*linspace(-0.5,0.5, length(Y));
7  fres = Fs*linspace(-0.5,0.5, length(Yres))/4;
8
9  f = f(:)';
10 fres = fres(:)';
11
12 sound(y, Fs);
13 pause(9);
14 sound(yres, Fs/4)
15
16 figure, plot(f, abs(Y))
17 xlabel('freq (Hz)');
18 ylabel('Magnitude');
19 title('Original - Magnitude Plot');
20
21 figure, plot(fres, abs(Yres))
22 xlabel('freq (Hz)');
23 ylabel('Magnitude');
24 title('Resampled - Magnitude Plot');
25
26 % Magnitude plot are different as well the heard sounds are also
27 different,
28 % so aliasing is detected
```

Lab 8 Tasks & Solution: Images & Sounds:

Task1:

Read the default image 'peppers.png' and display the RGB channels in its respective color.

Solution:

```
1  clc; close all; clear all;
2
3  img = imread('peppers.png');
4  img = double(img)/255;
5
6  imgR = zeros(size(img));
7  imgG = zeros(size(img));
8  imgB = zeros(size(img));
9
10 imgR(:,:,1) = img(:,:,1);
11 imgG(:,:,2) = img(:,:,2);
12 imgB(:,:,3) = img(:,:,3);
13
14 figure, subplot(221);
15 imshow(img);
16 title('Original');
17
18 subplot(222), imshow(imgR);
19 title('Red Channel');
20
21 subplot(223), imshow(imgG);
22 title('Green Channel');
23
24 subplot(224), imshow(imgB);
25 title('Blue Channel');
```

Task 2:

You need to implement Temporal Average Filter (TAF) using the provided images in the directory. The provided directory having n images having $p \times q$ pixels each image.

- Read the directory of the jpg files in MATLAB.
- Read and normalize each image to the range [0,1].
- Extract R, G and B channels and concatenate each channel into a separate array.
- Find the mean of n channels corresponding to R,G and B.

Signal & Systems Lab - CS341L - Fall2021

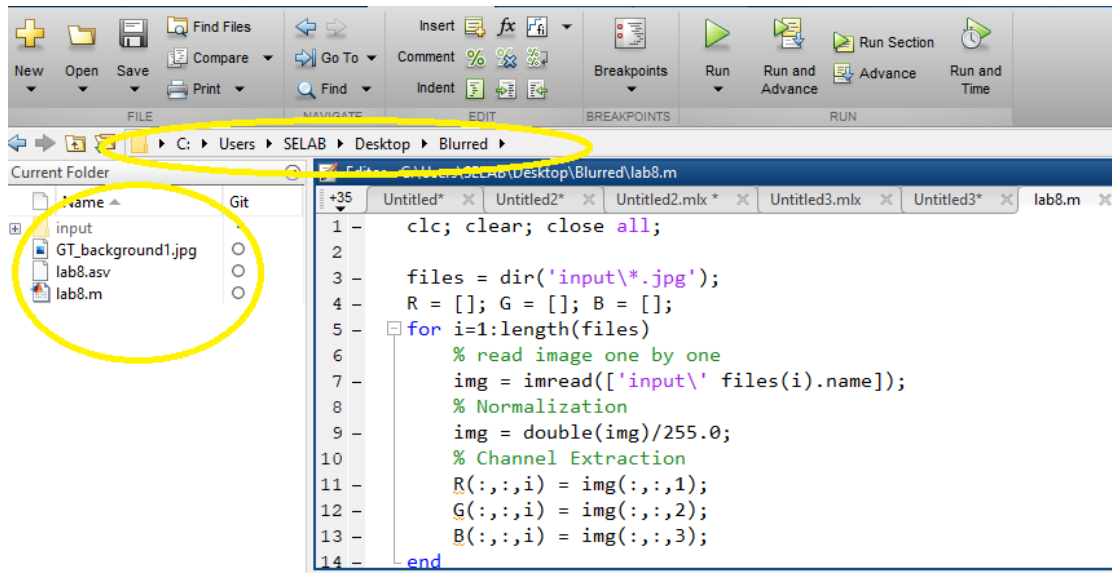
Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi

- Concatenate the means to obtain RGB image, and show it.

Solution:

Set MATLAB path according to the following snapshot and put the provided data in the current directory accordingly.



```
1 clc; clear; close all;
2 files = dir('input\*.jpg');
3 R = []; G = []; B = [];
4 for i=1:length(files)
5     % read image one by one
6     img = imread(['input\' files(i).name]);
7     % Normalization
8     img = double(img)/255.0;
9     % Channel Extraction
10    R(:,:,i) = img(:,:,1);
11    G(:,:,i) = img(:,:,2);
12    B(:,:,i) = img(:,:,3);
13 end
14 Rmean = mean(R, 3); Gmean = mean(G, 3); Bmean = mean(B, 3);
15 imgMean = cat(3, Rmean, Gmean, Bmean);
16 imshow(imgMean);
17 title('The Mean of all images');
18
19 % In the mean image you will hardly observed the moving objects,
20 % TAF is used for Background Estimation in image processing..
21 % The data is taken from Scene Background Challenge - 2017 - observe
22 % each variable size in workspace.
```

Signal & Systems Lab - CS341L - Fall2021

Faculty of Computer Science & Engg.

Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences & Technology, Topi
