Lab 1

ELECENG 3TP3

Gurleen Dhillon 400301955

Yuvraj Sandhu 400319134

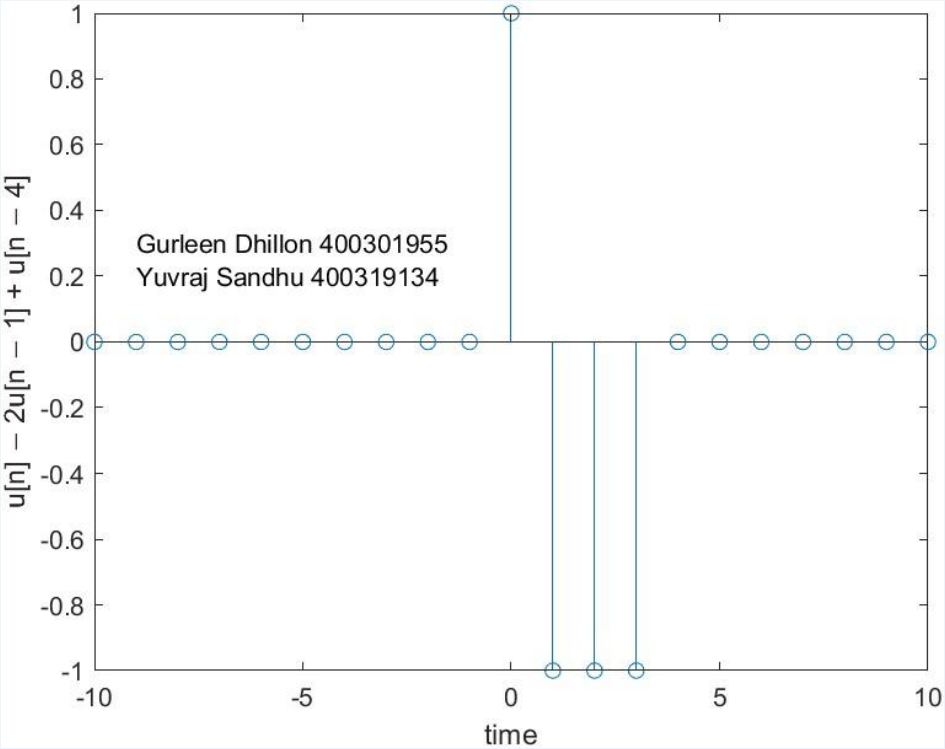
2022-10-03

Part 1

1A: x[n] = u[n] − 2u[n − 1] + u[n − 4]

```
clear
%creating a SimpleFunctions object
t = -10:10;
f = SimpleFunctions;

%plotting the discrete time function x1
x1 = f.unitstep(t) - 2*f.unitstep(t-1) + f.unitstep(t-4);
stem(t,x1)
xlabel('time')
ylabel('u[n] - 2u[n - 1] + u[n - 4]')
text(-9,0.3,'Gurleen Dhillon 400301955');
text(-9,0.2,'Yuvraj Sandhu 400319134');
exportgraphics(gcf, '3tp3_lab1_1a.jpg');
```

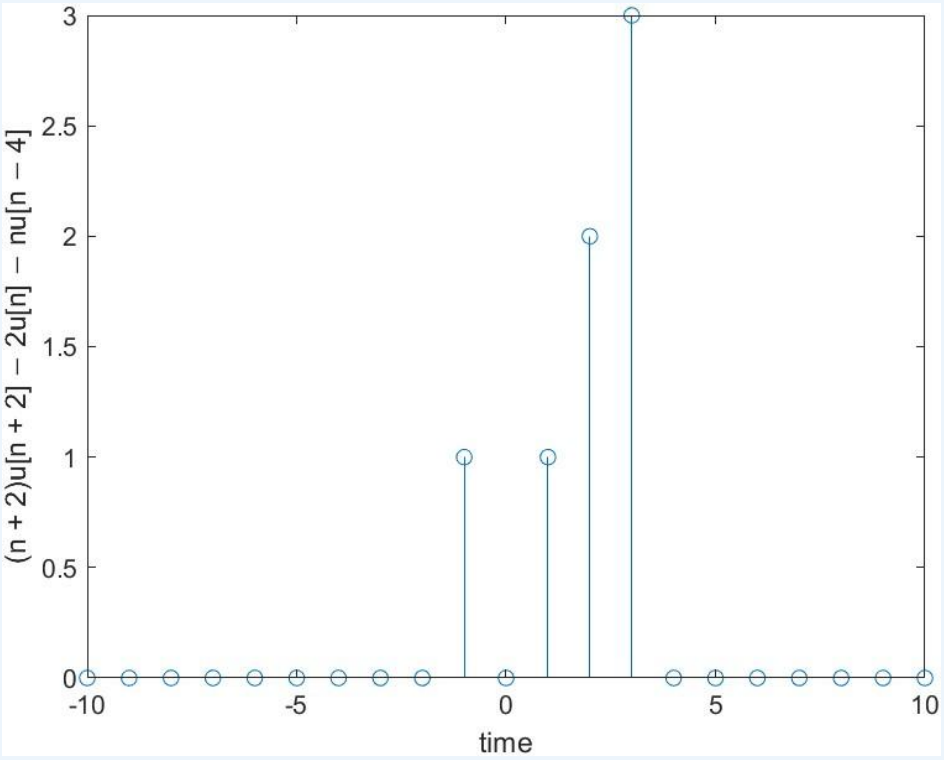


For this code, we plotted the function x[n] = u[n] − 2u[n − 1] + u[n − 4] using the SimpleFunctions method for implementing unistep on MATLAB within the range of -10 to 10. This function was then graphed on MATLAB as well.

1B: x[n] = (n + 2)u[n + 2] − 2u[n] − nu[n − 4]

```
clear
%creating a SimpleFunctions object
t = -10:10;
f = SimpleFunctions;

%plotting the discrete time function x2
x2 = (t + 2).*(f.unitstep(t + 2)) - 2.*f.unitstep(t) -
t.*f.unitstep(t - 4);
stem(t, x2)
xlabel('time')
ylabel('(n + 2)u[n + 2] - 2u[n] - nu[n - 4]')
text(-9,9,'Gurleen Dhillon 400301955');
text(-9,8,'Yuvraj Sandhu 400319134');
exportgraphics(gcf, '3tp3_lab1_1b.jpg');
```

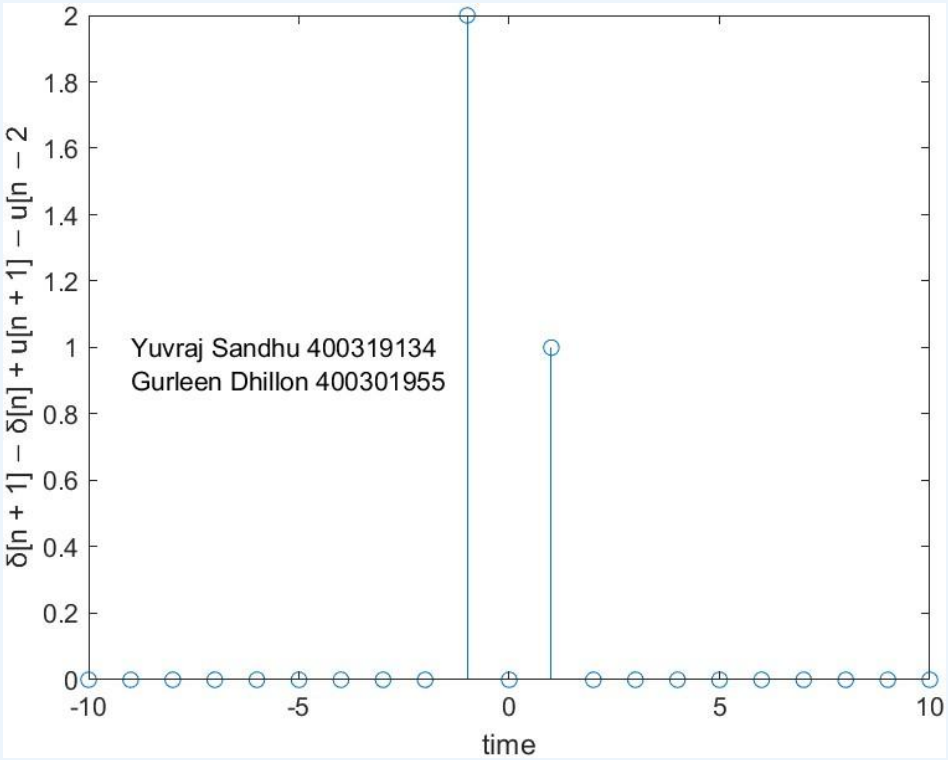


For this code, we plotted the function x[n] = (n + 2)u[n + 2] − 2u[n] − nu[n − 4] using the SimpleFunctions method for implementing unistep on MATLAB within the range of -10 to 10. This function was then graphed on MATLAB as well.

1C: x[n] = δ[n + 1] − δ[n] + u[n + 1] − u[n − 2]

```
clear
%creating a SimpleFunctions object
t = -10:10;
f = SimpleFunctions;

%plotting the discrete time function x3
x3 = f.delta(t + 1) - f.delta(t) + f.unitstep(t + 1) -
f.unitstep(t - 2);
stem(t, x3)
xlabel('time')
ylabel('δ[n + 1] − δ[n] + u[n + 1] − u[n − 2]')
text(-9,0.9,'Gurleen Dhillon 400301955');
text(-9,1,'Yuvraj Sandhu 400319134');
exportgraphics(gcf, '3tp3_lab1_1c.jpg');
```

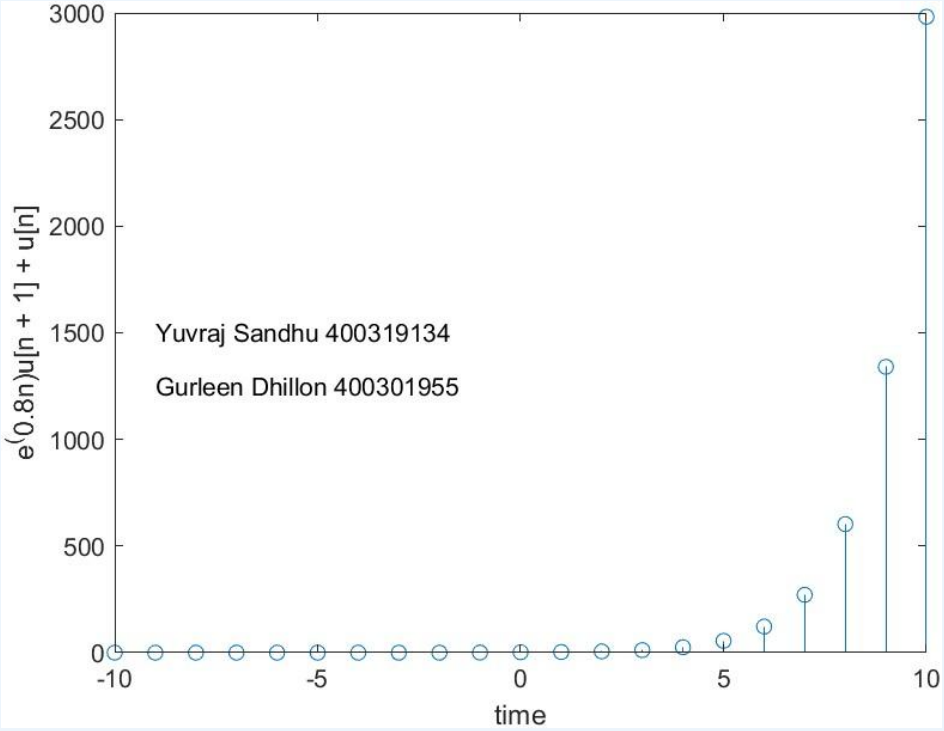


For this code, we plotted the function x[n] = δ[n + 1] − δ[n] + u[n + 1] − u[n − 2] using the SimpleFunctions method for implementing unistep and delta on MATLAB within the range of -10 to 10. This function was then graphed on MATLAB as well.

1D: $x[n] = e^{0.8n}u[n + 1] + u[n]$

```
clear
%creating a SimpleFunctions object
t = -10:10;
f = SimpleFunctions;

%plotting the discrete time function x4
x4 = (exp(0.8*t).*f.unitstep(t + 1)) + f.unitstep(t);
stem(t, x4)
xlabel('time')
ylabel('e^(0.8n)u[n + 1] + u[n]')
text(-9, 1250,'Gurleen Dhillon 400301955');
text(-9,1500,'Yuvraj Sandhu 400319134');
exportgraphics(gcf, '3tp3_lab1_1d.jpg');
```



For this code, we plotted the function $x[n] = e^{0.8n}u[n + 1] + u[n]$ using the SimpleFunctions method for implementing unistep on MATLAB within the range of -10 to 10. This function was then graphed on MATLAB as well.

Part 2

2A

```
clear
%opening excel file
opts = detectImportOptions('course_grades_2022.xlsx');
opts = setvartype(opts, {'ID_Number', 'Name'}, 'string');
table = readtable('course_grades_2022.xlsx', opts);

%save lab marks from table as local variables
lab1 = table.Lab_1(2:end);
lab2 = table.Lab_2(2:end);
lab3 = table.Lab_3(2:end);
lab4 = table.Lab_4(2:end);

%add all lab marks
total_marks = lab1 + lab2 + lab3 + lab4;

%finding highest mark and saving the number and its index
[highest_mark, i] = max(total_marks);
%using the index to find the name and display it
highest_name = table.Name(i+1)
highest_mark %display grade after name
```

After importing the excel table, we set each lab column (excluding the max mark entry) to its' own variable so that we could add all the lab columns into a new column matrix for the total. Now, we can use the max function to determine the highest mark in the labs, and at which position in the column it is in, and with that index we can find the corresponding name of the highest mark.

OUTPUT:

```
highest_name = "Morgan Bush"

highest_mark = 32
```

2B

```
clear
%opening excel file
opts = detectImportOptions('course_grades_2022.xlsx');
opts = setvartype(opts, {'ID_Number', 'Name'}, 'string');
table = readtable('course_grades_2022.xlsx', opts);
```

```
%save exam marks from table as local variables
exam1 = table.Exam_1(2:end);
exam2 = table.Exam_2(2:end);
exam3 = table.Exam_3(2:end);
exam4 = table.Exam_4(2:end);

%add all exam marks
total_marks = exam1 + exam2 + exam3 + exam4;

%finding highest mark and saving the number and its index
[highest_mark, i] = max(total_marks);
%using the index to find the name and display it
highest_name = table.Name(i+1)
highest_mark %display grade after name
```

After importing the excel table, we set each exam column (excluding the max mark entry) to its' own variable so that we could add all the exam columns into a new column matrix for the total. Now, we can use the max function to determine the highest mark in the exams, and at which position in the column it is in, and with that index we can find the corresponding name of the highest mark.

OUTPUT:

```
highest_name = "Anthony Bernard"

highest_mark = 37
```

2C

```
clear
%opening excel file
opts = detectImportOptions('course_grades_2022.xlsx');
opts = setvartype(opts, {'ID_Number', 'Name'}, 'string');
table = readtable('course_grades_2022.xlsx', opts);

%save marks from table as local variables
lab1 = table.Lab_1(2:end);
lab2 = table.Lab_2(2:end);
lab3 = table.Lab_3(2:end);
lab4 = table.Lab_4(2:end);
exam1 = table.Exam_1(2:end);
exam2 = table.Exam_2(2:end);
exam3 = table.Exam_3(2:end);
exam4 = table.Exam_4(2:end);
```

```
midterm = table.Midterm(2:end);

%add all marks
total_marks = lab1 + lab2 + lab3 + lab4 + midterm + exam1 +
exam2 + exam3 + exam4;

%finding highest mark and saving the number and its index
[highest_mark, i] = max(total_marks);
%using the index to find the name and display it
highest_name = table.Name(i+1)
highest_mark %display grade after name
```

After importing the excel table, we set each lab, midterm, and exam column (excluding the max mark entry) to its' own variable so that we could add all the lab, midterm, and exam columns into a new column matrix for the total. Now, we can use the max function to determine the highest mark in the for all labs, exams, and midterm, and at which position in the column it is in, and with that index we can find the corresponding name of the highest mark.

OUTPUT:

```
highest_name = "Anthony Bernard"

highest_mark = 79
```

2D

```
%write names
writematrix('Gurleen Dhillon', 'course_grades_2022.xlsx',
'WriteMode','append')
writematrix('Yuvraj Sandhu', 'course_grades_2022.xlsx',
'WriteMode','append')

%write numbers
n = [400301955 10 10 10 10 20 9 9 9 9; 400319134 9 9 9 9 20
10 10 10 10]
writematrix(n, 'course_grades_2022.xlsx', 'Range',
'B23:K24')
```

Since strings and integers are not able to fit into a matrix together, we first appended our names into the first column. After that, we were able to create a matrix for student numbers and marks together as they are all integers. We can that place this matrix in any box on the excel sheet if we know the coordinates, which are right after the input of our names.

OUTPUT:

	1 Name	2 ID_Number	3 Lab_1	4 Lab_2	5 Lab_3	6 Lab_4	7 Midterm	8 Exam_1	9 Exam_2	10 Exam_3	11 Exam_4
1	"Maximum...	"0"	10	10	10	10	20	10	10	10	10
2	"Kacie Ste...	"1803933"	7	2	9	0	9	4	5	8	10
3	"Yassin Jor...	"1884159"	1	2	10	3	8	3	9	5	7
4	"Lowri Mat...	"1853847"	2	0	0	2	17	6	10	7	4
5	"Tiya Sheri...	"1810192"	7	1	0	6	15	8	7	6	6
6	"Nikola For...	"1891352"	1	7	0	6	5	0	5	5	10
7	"Veer Blair"	"1811313"	4	8	5	3	12	7	4	0	2
8	"Isabelle M...	"1804841"	6	7	4	0	13	8	9	6	4
9	"Samir Gre...	"1881925"	9	3	7	1	6	4	6	5	9
10	"Zander Ke...	"1877711"	8	10	5	4	17	4	8	10	2
11	"Shahzaib ...	"1830894"	4	5	7	9	8	5	7	0	6
12	"Morgan B...	"1855191"	9	6	7	10	1	5	7	2	8
13	"Amaan Ro...	"1821012"	1	8	4	4	8	0	9	5	8
14	"Theodore ...	"1844339"	5	7	10	7	14	9	2	2	9
15	"Ace Branc...	"1898468"	2	1	3	7	11	9	9	3	6
16	"Anthony B...	"1883633"	4	1	10	8	19	10	9	9	9
17	"Tobey Bell"	"1808742"	0	10	8	2	10	9	0	8	6
18	"Jannat Cas...	"1863450"	1	2	4	5	10	4	5	9	3
19	"Imran Mar...	"1830190"	2	9	1	6	17	10	0	7	5
20	"Amani Ca...	"1835544"	8	9	5	7	3	7	6	8	4
21	"Blanka Ho...	"1820930"	6	5	2	0	8	6	0	7	10
22	"Gurleen D...	"400301955"	10	10	10	10	20	9	9	9	9
23	"Yuvraj San...	"400319134"	9	9	9	9	20	10	10	10	10

Part 3

```
clear
%original image
img = imread('ee3tp3picture2022.jpg');
subplot(1, 2, 1);
imshow(img, []);
title('original');

r = img(:,:, 1);
g = img(:,:, 2);
b = img(:,:, 3);
rNew = r*6;
gNew = g*6;
reconstructed_image = cat(3, rNew, gNew, b);

%new image
subplot(1, 2, 2);
imshow(reconstructed_image, []);
title('new');
imwrite(reconstructed_image, 'bunny.jpg');
```

To create this image, we first set up the code so that it would display both images side by side using the subplot feature. After a few rounds of trial and error of multiplying the red and green values by the same number, we found a version which we thought would be the closest to the original image in colour. The blue value was not altered as it was already at its highest point in the original photo and altering that value would not help recolour the picture.