Electrical Engineering 3TP3
Lab 1
Matlab Introduction and Refresher

This lab is intended as a Matlab introduction and refresher. Some simple discrete-time signal plotting is first done. Matlab is then used to read and process the output of a spreadsheet file that contains student course grades. Finally, Matlab is used to read a supplied image and improve its quality by adjusting its contrast levels. This is done by displaying and manipulating its luminosity histogram.

1 Preparation

1. Make sure that you have installed either Matlab or Octave on your PC or laptop. Refer to the Laboratory section of the course web site for details.
2. Read through Chapter 1 in the textbook. Review and try all of the Matlab examples on your PC or laptop.
3. There are many good online Matlab tutorials. Some of these have been listed on the course web site. Spend some time looking through them as a refresher.

2 Procedure

Question 1 below, contains discrete-time functions that include products of unit-step functions. To evaluate the given functions, define a function of your own that accepts a time vector, and returns the discrete-time unit step function of that input. Here is an example of that function that you can place in the M-file unitstep.m

```
function y = unitstep(x)
% The unit step function, u(x).

    if (nargin != 1)
     disp('unit step requires 1 argument!');
     return
  end

  y = cast(x >= 0, class(x));
```

You can now include unitstep(t) in the expressions that you create to evaluate the given functions[1] Time shifts will also work, e.g., unitstep(t-2). Create a unit-pulse function in the same way. So for example, a unit-ramp function and one shifted one time unit to the right and evaluated over the integers from -10 to 10 would be

```
t = -10:10;
unitramp = t.*unitstep(t);
unitramp_rightshift = (t-1).*unitstep(t-1);
```

1. Use Matlab to plot the following discrete time signals.

   (a) $x[n] = u[n] - 2u[n-1] + u[n-4]$

   (b) $x[n] = (n+2)u[n+2] - 2u[n] - nu[n-4]$ (c)

   $x[n] = \delta[n+1] - \delta[n] + u[n+1] - u[n-2]$ (d)

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

2. In this section, use Matlab to process student grades obtained from a spreadsheet file. Download the file course grades 2020 .csv, which is a comma-separated values (csv) file exported from the supplied course grades 2020 .xlsx spreadsheet (Before exporting the csv file, lines 1-3 and 5 were first deleted from the spreadsheet).

   The file contains student grade records that start on Line 2 with student ID number 1863684. You can see from course grades 2020 .xlsx that the student marks include 4 labs, a midterm, and 6 exam questions. Note that the entries on Line 1 give the maximum grades for each grade entry type (See course grades 2020 .xlsx.)

   You can use the Matlab csvread function to read the csv file, where specified sub-block offsets can be accessed. For example, if you use

   csvread('course_grades.csv', 1, 0, [1, 0, 5, 0])'

   then it will read starting at file offset 1,0 and read the block [1, 0, 5, 0], i.e., it will return a vector of the first 5 student ID numbers[2]

   [1863684 1812868 1839570 1840883 1884504]

   Perform the following tasks. Use vector operations in your Matlab code rather than using Matlab program loops.

   (a) Write a Matlab function that accepts: 1) the set of student grade records as a ma- trix (read from the csv file), 2) the maximum grade vector, and, 3) a vector of column indices. The function will generate a vector of grade averages, i.e., one entry for each student, for the associated grades corresponding to the column indices.

[1]Make sure that you do that correctly, i.e., you must use element-wise operations.

[2]Note that the syntax used for Octave is csvread('course grades.csv', [1, 0, 5, 0])'. You need to be careful when using Matlab csvread, since it expects to find numeric comma-separated values in all the lines that it processes. If you process the entire csv file directly after exporting it from the xlsx file, i.e., without deleting lines 1-3 and 5, then there will be lines that contain text that csvread will reject. You can still read in the file but you have to specify csvread block offsets that do not include those lines.

For example, if the function is named csv_get_averages, then if we call it with csv_get_averages(m, max, [2:5]) (where m is the student grade record matrix), and max is the maximum grades vector, it will return a vector of average lab marks (i.e., using columns 2 to 5), one entry for each student record.

(b) Write a Matlab script that uses the function that you wrote to output the following: 1) average lab marks for each student, 2) overall average lab grade for the course, 3) average exam marks for each student, and, 4) overall average exam grade for the course.

(c) Write a Matlab script to obtain an array of the final grade for each student assuming that the exam is worth 40%, labs are 30% and the midterm is 30%. Create a plot of the final grades, in decreasing order of final grade.

3. In this section, use Matlab to do some simple image processing. Download the image ee3tp3picture2020.png from the course web site and import it as a matrix into Mat- lab. This can be done using the statement

```
image = imread('ee3tp3picture2020.png');
```

ee3tp3picture2020.png is a greyscale image of unsigned 8-bit integers (pixels) in the range [0, 255], where 0 is black and 255 is white. The quality of the provided image is not very good and needs to be fixed.

Before continuing on, convert the pixel value types to double precision floating point numbers. Otherwise, the processing you do later may not work as expected. This can be done using the statement

```
image_of_doubles = double(image);
```

(a) Plot a histogram of the pixel values in ee3tp3picture2020.png over the range [0, 255] using the hist Matlab function. To visualize the histogram better, generate it using hist and use the bar function for plotting as follows

```
[n_elements, centers] = hist(image_of_doubles(:), num_bins);
bar(centers, n_elements);
xlim([0 255]);
```

You can use num_bins to set the number of bins (maybe 20).

(b) Display the image from inside Matlab. This can be done using

```
imshow(uint8(image_of_doubles));
```

What is wrong with the image quality?

(c) Use Matlab to improve the image[3]. If the value of the pixel at (x, y) is v(x, y), try using the following transformation on a pixel-by-pixel basis.

$$v^0(x, y) = \alpha\, v(x, y) + \beta \qquad (1)$$

[3]Do not use any of Matlab's built-in image processing functions! Instead, do the improvement yourself.

where α and β are appropriately chosen constants and $v^0(x, y)$ is the new pixel value. These can be chosen by looking at the original histogram and figuring out how you want to transform it.

(d) Plot the final histogram once you have fixed it.

(e) View and print out the final image. If you want to save the image you need to convert the pixel values back to 8-bit unsigned integers before exporting it, i.e.,

```
image_to_save = uint8(fixed_image);
imwrite(image_to_save, 'saved_image.png');
```

3 Important Notes

1. When generating the graphs, make sure that you save it in a format (e.g., jpg) that can be imported into your lab report. In Matlab you can easily do this from the plot window. In Octave (or Matlab) you do it in your M-file, e.g., print -djpg plot.jpg will create a jpeg of your current plot.
2. Place text on each graph that gives the name(s) and student ID number(s) of those in your group. Otherwise, your report will not be accepted. This can be done using the Matlab text command, e.g., text(100, 1, 'John Smith 5551212'); will place John Smith 5551212 at coordinate location (100, 1) on the current plot.
3. Include listings of any Matlab code that you write, including comments describing how it works.

4 Write-up

Submit a write-up for the lab. Each group (of 2 maximum) may submit a single write-up. Include in your write-up a short description of everything that you did including all M-files and graphs.

5 Due Date

The due date for this lab is announced in class and is available on the course web site.