EE 3054: Signals, Systems, and Transforms

Matlab Exercises

1. The lab will meet every week. Be sure to review the lab ahead of the lab session. Part of your lab grade will be based on your progress during the lab session.

2. Labs and the data files required for some labs will be available on the web at:

   `http://taco.poly.edu/selesi/EE3054labfiles/`

3. A lap report for each lab (except for Lab 1) will be due the following week in lab. No report is required for Lab 1.

4. There will be a paper/pencil quiz early in the semester on basic Matlab usage.

5. The lab consists of computer-based exercises. You are required to bring your laptop computer to lab with Matlab installed. In case you do not have Matlab installed on your laptop you can go to the laptop help desk to have it installed. You will need the *Signal Processing Toolbox*.

6. The first lab introduces you to Matlab. You should go through the tutorial *Getting Started with Matlab*. This tutorial can be downloaded for free from the Mathworks website:

   `http://www.mathworks.com/access/helpdesk/help/pdf_ doc/matlab/getstart.pdf`

   Other documentation can be obtained at:

   `http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml`

   For convenience, the Matlab tutorial is also availabe at:

   `http://taco.poly.edu/selesi/EE3054labfiles/`

Lab 1: Introduction to Matlab

1. Creating vectors

   (a) Generate the following vectors:

   `A = [1 0 4 5 3 9 0 2]`

   `a = [4 5 0 2 0 0 7 1]`

   Be aware that Matlab are case sensitive. Vector A and a have different values.

   (b) Generate the following vectors:

   `B = [A a]`

   `C = [a,A]`

   Concatenation is the process of joining small matrices to make bigger ones. In fact, you made your first matrix by concatenating its individual elements. The pair of square brackets, `[]`, is the concatenation operator.

   (c) Generate the following vectors using function `zeros` and `ones`:

   `D = [0 0 0 ... 0]` with fifty 0's.

   `E = [1 1 1 ... 1]` with a hundred 1's.

   (d) Generate the following vectors using the colon operator

   `F = [1 2 3 4 ... 30]`

   `G = [25 22 19 16 13 10 7 4 1]`

   `H = [0 0.2 0.4 0.6 ... 2.0]`

   The colon, :, is one of Matlab's most important operators.

2. Operate with the vectors

   `V1 = [1 2 3 4 5 6 7 8 9 0]`

   `V2 = [0.3 1.2 0.5 2.1 0.1 0.4 3.6 4.2 1.7 0.9]`

   `V3 = [4 4 4 4 3 3 2 2 2 1]`

   (a) Calculate, respectively, the sum of all the elements in vectors V1, V2, and V3

   (b) How to get the value of the fifth element of each vector?

   What happens if we execute the command V1(0) and V1(11)?

   Remember if a vector has N elements, their subscripts are from 1 to N.

(c) Generate a new vector V4 from V2, which is composed of the first five elements of V2.

Generate a new vector V5 from V2, which is composed of the last five elements of V2.

(d) Derive a new vector V6 from V2, with its 6th element omitted.

Derive a new vector V7 from V2, with its 7th element changed to 1.4.

Derive a new vector V8 from V2, whose elements are the 1st, 3rd, 5th, 7th, and 9th elements of V2

(e) What are the results of

```
9-V1

V1*5

V1+V2

V1-V3

V1.*V2

V1*V2

V1.^2

V1.^V3

V1^V3

V1 == V3

V1>6

V1>V3

V3-(V1>2)

(V1>2) & (V1<6)

(V1>2) | (V1<6)

any(V1)

all(V1)
```

3. Using Matlab help system, click on

   ```
   Help -> MATLAB help
   ```

   or type `helpdesk` to can open the help files. For description of a single function or command, type

   ```
   help command_name
   ```

   on the command line, or use 'search' in the help window.

   For example, type

   ```
   help plot
   ```

   on the command line.

4. Flow control

   What are the results of these sets of commands? Think them over and run them with Matlab to see if you are right.

   (a)
   ```
   A = zeros(1,5);
   for n = 1:4
      for m = 1:3
         A = A + n*m;
      end
   end
   A
   ```

   (b)
   ```
   B = [1 0];
   if (all(B))
      B = B + 1;
   else if (any(B))
      B = B + 2;
   else
      B = B + 3;
   end
   B
   ```

(c)
```
C = 7:2:22
num = 0;
while (all( C>0))
   C = C - 3;
   num = num + 1;
end
C
num
```

(b) Situations under which loops can be avoided. Do the following sets of commands have same results?

```
for i = 1:1000000
   H(i) = i * 5;
end
```

```
H = 1:1000000;
H = H*5;
```

Do the following sets of commands have same results?

```
for n = 1:100000
    x(n) = sin(n*pi/10);
end
```

```
n = 1:100000;
x = sin(n*pi/10);
```

5. Compare a script and a function

(a) Write a script: In the main menu of Matlab, select

```
file -> new -> M-file
```

A new window will pop up. Input the following commands:

```
x = 1:5;
y = 6:10;
g = x+y;
```

and then save the file as `myscript.m` under the default path matlab/work

(b) Write a function: Create a new `m` file following the procedure of above. Type in the commands:

```
function g = myfunction(x,y)
g = x + y;
```

and then save it as `myfunction.m`

(c) Compare their usage

(1) run the commands one by one:

```
myscript
x
y
g
z = myscript    (error?)
```

(2) Run command `clear` to remove all variables from memory

(3) Run the commands one by one:

```
x = 1:5;
y = 6:10;
myfunction     (error?)
z = myfunction(x,y)
g             (error?)
a = 1:10;
b = 2:11;
myfunction(a,b)
```