Signal Processing First
Lab 01: Introduction to MATLAB

Pre-Lab and Warm-Up: You should read at least the Pre-Lab and Warm-up sections of this lab assignment and go over all exercises in the Pre-Lab section before going to your assigned lab session.

---

1 Pre-Lab

In this first week, the Pre-Lab will be extremely short and very easy. Make sure that you read through the information below prior to coming to lab.

1.1 Overview

MATLAB will be used extensively in all the labs. The primary goal of this lab is to familiarize yourself with using MATLAB. Here are three specific goals for this lab:

1. Learn basic MATLAB commands and syntax, including the `help` system.
2. Learn to write and edit your own script files in M ATLAB, and run them as commands.
3. Learn a little about advanced programming techniques for MATLAB, i.e., vectorization.

1.2 Getting Started

(a) Explore the MATLAB help capability available at the command line. Try the following:

```
help
Help plot
help colon          %<--- a VERY IMPORTANT notation
help ops
help zeros
help ones
lookfor filter      %<--- keyword search
```

NOTE: it is possible to force MATLAB to display only one screen-full of information at once by issuing the command more on).

(b) Use MATLAB as a calculator. Try the following:

```
pi*pi - 10
sin(pi/4)
ans^2      %<--- "ans":holds the last result
```

(c) Do variable name assignment in MATLAB. Try the following:

```
x= sin( pi/5 );
cos( pi/5 )  %<--- assigned to what?
y = sqrt( 1 - x*x )
ans
```

(d) Complex numbers are natural in MATLAB. The basic operations are supported. Try the following:

```
z = 3 + 4i; w = -3 + 4j
real(z), imag(z)
abs([z,w])          %<-- Vector constructor
conj(z+w)
angle(z) exp( j*pi )
exp(j*[ pi/4, 0, -pi/4 ])
```

2 Warm-Up

2.1 MATLAB Array Indexing

(a) Make sure that you understand the **colon** notation. In particular, explain in words what the following MATLAB code will produce

```
jkl = 0 : 6
jkl = 2 : 4 : 17
jkl = 99 : -1 : 88
ttt = 2 : (1/9) : 4
tpi = pi * [ 0:0.1:2 ];

x=[zeros(2,2) [-5 -4 3 ; 4 1 3] ones(2,4)]

x(2,5)                          x(1,2:5)
x(1,:)                          y=x
x(:,1)                          y=(1:2,2:4)=pi*[1,-1,1;-1 1 -1]
x(:)
```

(b) Extracting and/or inserting numbers into a vector is very easy to do. Consider the following definition of xx:

```
   xx = [ zeros(1,3), linspace(0,1,5), ones(1,4) ]
xx(4:6)                                 xx(:)
size(xx)                                xx(1,:)
length(xx)                              xx(:,1)
xx(2:2:length(xx))                      xx(4,2)
```

Explain the results echoed from the last four lines of the above code.

(c) Observe the result of the following assignments:

```
yy = xx; yy(4:6) = pi*(1:3)
```

Now write a statement that will take the vector xx defined in part (b) and replace the even

indexed elements (i.e., xx(2), xx(4), etc) with the constant π^{π}

```
x=[zeros(2,2) [-5 -4 3 ; 4 1 3] ones(2,4)];

x(2,5);                          x(1,2:5);
x(1,:);                          y=x;
x(:,1);                          y=(1:2,2:4)=pi*[1,-1,1;-1 1 -1];
x(:);
```

2.3 MATLAB Script Files

(a) Use the built-in MATLAB editor to create a script file called mylab1.m containing the following lines:

```
tt = -1 : 0.01 : 1;
xx = cos( 5*pi*tt );
zz = 1.4*exp(j*pi/2)*exp(j*5*pi*tt);
```

i.

```
plot(xx) ; figure ; plot(tt,xx) ; figure
plot( tt, xx, 'b-', tt, real(zz), 'r--' ),grid on  %<--- plot a sinusoid
title('TEST PLOT of a SINUSOID')
xlabel('TIME (sec)') ; ylabel('AMP')
```

ii.

```
plot(tt,xx) ; hold; plot(zz)
```

iii.

```
plot(tt,xx)
figure
plot(tt,zz)
```

iv.

```
subplot(1,2,1) ; plot(tt,xx)
subplot(1,2,2) ; plot(tt,zz)
```