

T-61.3010 Digital Signal Processing and Filtering

(v. 1.00, ..2010xxx), Matlab #1 (21.1., 22.1., 23.1., 24.1.2009)

Registration in WebOodi. Bring your own **headphones** if you have. The assistant will guide you through the exercises, but you may go on your own speed. Feel free to ask the assistant, if you have troubles. You can also consult http://www.cis.hut.fi/Opinnot/T-61.3010/how_to_start_with_matlab.shtml or [kuinka_aloitan_matlabin.shtml](http://www.cis.hut.fi/Opinnot/T-61.3010/kuinka_aloitan_matlabin.shtml).

Getting started: In **Windows** just click **Programs - Matlab**. Write down the code into separate files in your working directory (e.g. Z:\DSP\) for future use. Set the “Current Directory” in Matlab to point to the working directory (or type `cd <workdir>`).

The problems marked with [Pxx] are from the course exercise material (Spring 2009).

In the end of this session **you should know**: (a) Matlab 7 user interface, (b) how to find help in Matlab, (c) some elementary math commands, (d) how to write Matlab scripts and functions using the editor, (e) how to plot data, (f) how to export figures into a file.

1. [M2050] You can use Matlab as a calculator. Type each of the following on separate lines.

```
x = 5, z = 3 + 4*j           % real, complex numbers
exp(-j*pi)                  % complex exponential
log10(100), log2(8), atan(4/3), sqrt(25),
abs(z), angle(z), real(z), imag(z)
```

Task: What can you see in the tab “Workspace” and in the window “Command history”. Type 1 (letter “l” without enter) in the command window and use (a) up/down arrow keys, (b) tabulator - how does Matlab respond? What does the semicolon “;” in the end of line do?

2. [M2051] Matlab deals with matrices (sizes: **matrix** $M \times N$, where M rows and N columns; **column vector** $M \times 1$; **row vector** $1 \times N$, **scalar** 1×1). Run the following lines – there will be also some errors due to illegal operations (correspondingly to math matrix calculus).

```
y = [3 2 -1 6]           % row vector
x = [2 : pi : 12].’      % column vector: [start : interval : end]
y*x                      % inner product, dim: (1 x 4)*(4 x 1) = (1 x 1)
XY = x*y                 % outer product, dim: (4 x 1)*(1 x 4) = (4 x 4)
size(x*y)                % [#rows #columns]
XY(3, 2:4) = 0           % assign zeros, position: matrix(rows, columns)
y*y                      % matrix dimensions: (???)*(???) = (???)
y.*y                     % itemwise: == [y(1)*y(1) y(2)*y(2) ...]
length(y.*y)             % max{#columns, #rows}
y./x’                    % itemwise: == [y(1)/x(1)’ y(2)/x(2)’ ...]
plot(x, y)                % figure: adds a line between the four points
                          % (x1,y1) -- (x2,y2) -- (x3,y3) -- (x4,y4)
plot(x, y, ‘-o’, x, y, ‘o’) % line AND circles at each data point
rand(2,5), randn(3,2), ones(5,1), zeros(4,1)
```

Task: If **A** is a matrix of size 3×5 , and **b** is a row vector of size 1×5 , give an example of a legal matrix product. What is difference between \mathbf{b}^T (**b.’**) and \mathbf{b}^H (**b’**)?

```
A = [5 2 -1 4 9; 3 -2 j 9 0.2; 8 4 -2 -2 0]
b = [j 4+j 2-j 1 -2]
```

3. [M2052] How to find help in Matlab?

```
help                       % as always, try this first
help <function>           % command line help for a specific <function>
doc <function>            % full documentation for a specific <function>
which <function>          % path name of <function>
```

Task: Create the following curve of with green color. See `plot` command for details. What does the plotting command `plot(y)` do?

```
x = [0 : 0.1 : 0.5];
y = rand(1,6);
plot(x, y, <addhere>) % ‘help plot’
plot(y)                % see x-axis values!
```

You can also go to the support page <http://www.mathworks.com/support/>, or to Matlab Central <http://www.mathworks.com/matlabcentral/> in order to find technical solutions or Matlab files.

4. [M2053] Matlab **script** and **function** files both end with the extension `.m`. You can run them from command line with the base name without the extension. A function normally takes in some values and returns some back. A script is just a bunch of lines interpreted line after line in the main workspace.

Use **Matlab editor** which can be opened by typing `edit` (or any other text editor). Make sure that your “Current Directory” is the working directory (e.g. Z:\DSP\) or you have it in your Matlab path.

Next there is an example of a function, whose file name is `getdeg.m`, and which can be executed from the command window prompt by typing `[degrees] = getdeg(3+4*j)`. The first line contains a reserved word **function**, all output variables in brackets with comma separated, and all input variables after the function name in parentheses with comma separated. The following consecutive lines starting with `%` are comment lines that can be seen on screen when typing `help getdeg`.

```
function [deg] = getdeg(z)
% GETDEG computes the angle of the complex number z = x + yj in degrees
% Usage: [deg] = getdeg(z)
x = real(z);
y = imag(z);
deg = 180 * atan(y/x) / pi;
```

Task: Write a **function** which takes as an input argument the radius **r** of the circle, and returns the area **A** and the diameter **d**.

5. [M2054] **Task:** Compute some values of a complex-valued function $H(\omega)$ below and plot the results. Write down a **script file**, e.g. with name `myComplexFunctionH.m`. In Matlab 7 you can use two comment signs (`%%`) to divide the code into cells (blocks); choose “Cell - Enable Cell Mode” in Matlab editor (`edit`). You can also run the script from “Command window” in Matlab with `myComplexFunctionH` (without the extension `.m`).

```

%% Complex-valued function
% Matlab 1 / Problem 5
% Examine function
%
% $$H(\omega) = 2 - e^{-j \omega}$$
%
%% (a) Compute values
w = [0 : pi/4 : pi]'; % omega values, now as a column vector
H = 2 - exp(-j * w); % H(omega)
%% (b) Plotting in (x,y)
figure(1); clf; % open/activate Fig. 1, and clear it
x = % help real
y = % help imag
... % help plot
xlabel('x'); ylabel('y');
title('H in complex plane');
grid on;
%axis equal % useful or not ???
%% (c) Plotting |H| % abs, ...
r =
...
%% (d) Plotting <H % angle, ...
theta =
...
%% All values
disp([' w x y |H| <H']);
disp([w x y r theta]);

```

[4] Examine a complex-valued function

$$H(\omega) = 2 - e^{-j\omega}$$

where $\omega \in [0 \dots \pi] \in \mathbb{R}$.

- Compute values of Table 1 with a calculator. Euler: $e^{j\omega} = \cos(\omega) + j \sin(\omega)$.
- Draw the values at $\omega = \{0, \pi/4, \dots, \pi\}$ into complex plane (x, y) . Interpolate smoothly between the points.
- Sketch $|H(\omega)|$ as a function of ω . Interpolate smoothly.
- Sketch $\angle H(\omega)$ as a function of ω . Interpolate smoothly.

ω	$x = \text{Real}(H(\omega))$	$y = \text{Imag}(H(\omega))$	$r = H(\omega) $	$\theta = \angle H(\omega)$
0				
$\pi/4$				
$\pi/2$				
$3\pi/4$				
π				

Table 1: Problem 5: values of a complex-valued function in rectangular (x, y) and polar (r, θ) coordinates.

When working with Matlab, you can now try computing $H(\omega)$ densely, e.g., $w = [0 : \pi/256 : \pi]$. As you can see, now the curve **seems** to be smoothly continuous. You can also save (print) any figure to a file either from menu File - Save as or from the code using

```

figure(3); % activate the figure you want to save
print -dpng myFigure3.png % or function 'saveas'

```