

Introduction to MATLAB

Applied Signal Processing Laboratory 2023



**Politecnico
di Torino**



What is MATLAB?

- Wikipedia definition:
 - *"MATLAB (an abbreviation of "MATrix LABoratory") is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix multiplications, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages"*
- MATLAB provides a scientific-friendly programming environment that ease many kind of applications development
 - Lot of libraries for a variety of fields
 - Interpreted language (but code could be also compiled to create app)
 - Management of variables type without user intervention
 - Particularly powerful for vectorial operations and plot generation
 - Mostly used by academics and research institutions, but no open access
 - Less used by companies (Python is a good alternative)

Interface

CURRENT FOLDER

Actual position in the File Explorer. Available file and subfolders are here.

WORKSPACE

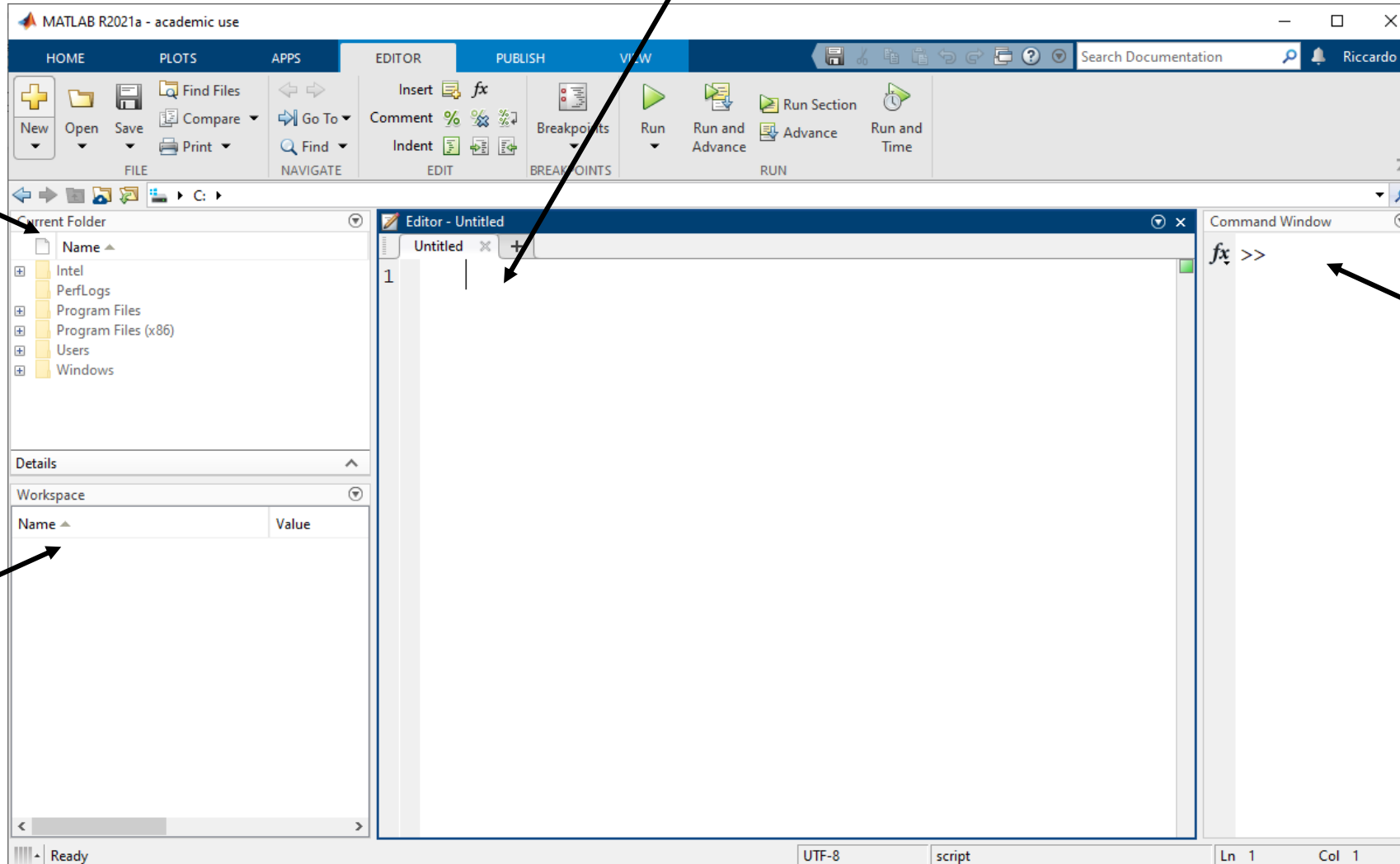
All the variables generated during the code run will be reported here. They can be accessed, and their content can be checked (very useful for debugging).

EDITOR

Text editor for code writing

COMMAND WINDOW

Standard input (command trial) and standard output (text output and errors)
e.g. try to write:
"help length"



Variables (1)

- Wikipedia definition:
 - *"Abstract storage location paired with an associated symbolic name, which contains some known or unknown quantity of information referred to as a value; or in simpler terms, a variable is a named container for a particular set of bits or type of data (like integer, float, string etc...)."*
- A variable is a container of some kind of data stored in the memory. It is characterized by a name assigned by the programmer, which should be descriptive of the variable's role.
- E.g.: height = 4.10
 - Variable named "height", which contains the value 4.10

Variables (2)

- Example of variable declaration and assignment in MATLAB

```
10      % Variables
11
12      % Numeric values
13 -    height = 4.10;
14      % String
15 -    address = "378 Main St. 10020";
16      % Arrays
17 -    daily_rains = [10, 0, 3, 0, 5, 20, 15, 0];
```

- We do not have to specify the type of variables; MATLAB automatically determine them for us

Basic operations

```
25
26     % Operations
27
28 -   a = 10;
29 -   b = 12;
30 -   s = a + b;      % sum
31 -   m = a * b;      % product
32 -   d = a / b;      % division
33 -   p = a^2;        % power
34 -   r = sqrt(a);    % square root (= a^(1/2))
35 -   rem = mod(a,b); % remainder of a division
36
37
```

Operations with arrays

- Element-wise multiplication:
 $a = [a_1, a_2, a_3]; b = [b_1, b_2, b_3]$ NOTE: they have the same number of values!
 $a .* b = [a_1*b_1, a_2*b_2, a_3*b_3]$
- Matrix multiplication (row-column product):
 - $a = [1 \times 3]$ array (row vector); $b = [3 \times 1]$ array (column vector)
 - $a * b = [1 \times 1]$ array (= scalar)
 - a' is the transpose array of a
 - $a' * b' = [3 \times 3]$ array (= matrix)
 - $a' * b = \text{error}$
 - $a * b' = \text{error}$
 - Analogous for matrices: the second dimension of the first matrix must be equal to the first dimension of the second matrix

Conditional statements

- Conditional expressions are fundamental for code development with programming languages
- They permit to define conditions that must be verified by the machine, and that can produce different outcomes depending on that condition

- E.g.:

```
if traffic_light == "red"
```

```
    Stop = 1;
```

```
else
```

```
    Stop = 0;
```

```
end
```

condition

instruction executed if condition is **true**

instruction executed if condition is **false**

Loops (control flow statements)

- Programming languages constructs to perform repetitive tasks
- E.g. *for* loop:

```
for i = 1:10
```

```
    accum = accum+value(i);
```

```
end
```

loop counter

instruction to be repeated

- Another kind of loop is the *while*

```
while i < 11
```

```
    accum = accum+value(i);
```

```
    i = i+1;
```

```
end
```

loop condition

instruction to be repeated

counter update

Functions (1)

- Wikipedia definition:
 - *"Sequence of program instructions that performs a specific task, packaged as a unit. This unit can then be used in programs wherever that particular task should be performed."*
- A function is a piece of code which can accept one or more input values and can return one or more output values, obtained through a sequence of operations (instructions) on the input.

Functions (2)

- Es:

```
c = sum_1(a,b)
```

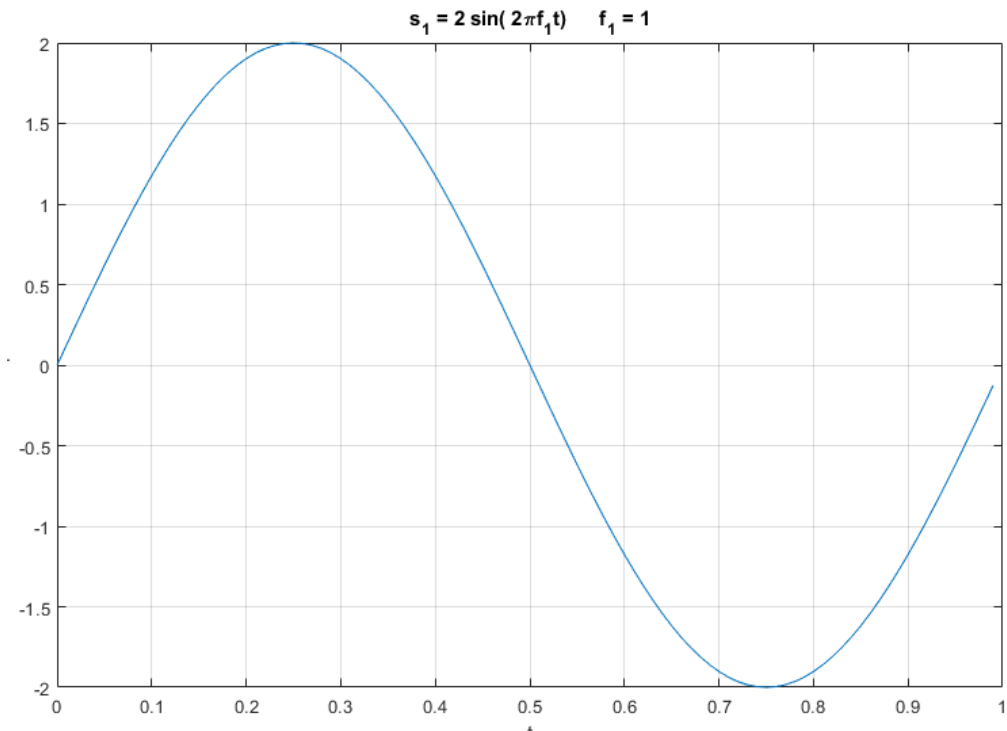
```
function [result] = sum_1(first, second)
result = first + second;
end
```

- "sum_1" is a function which requires as input two variables, *a* and *b*, and return as output a new value which is the sum of the two, saved in variable *c*
- The code of the function can be moved into a different file, e.g. "sum_1.m"; if it is in the same folder as the main program, MATLAB will load "sum_1.m" when we "call" (use) the *sum()* function

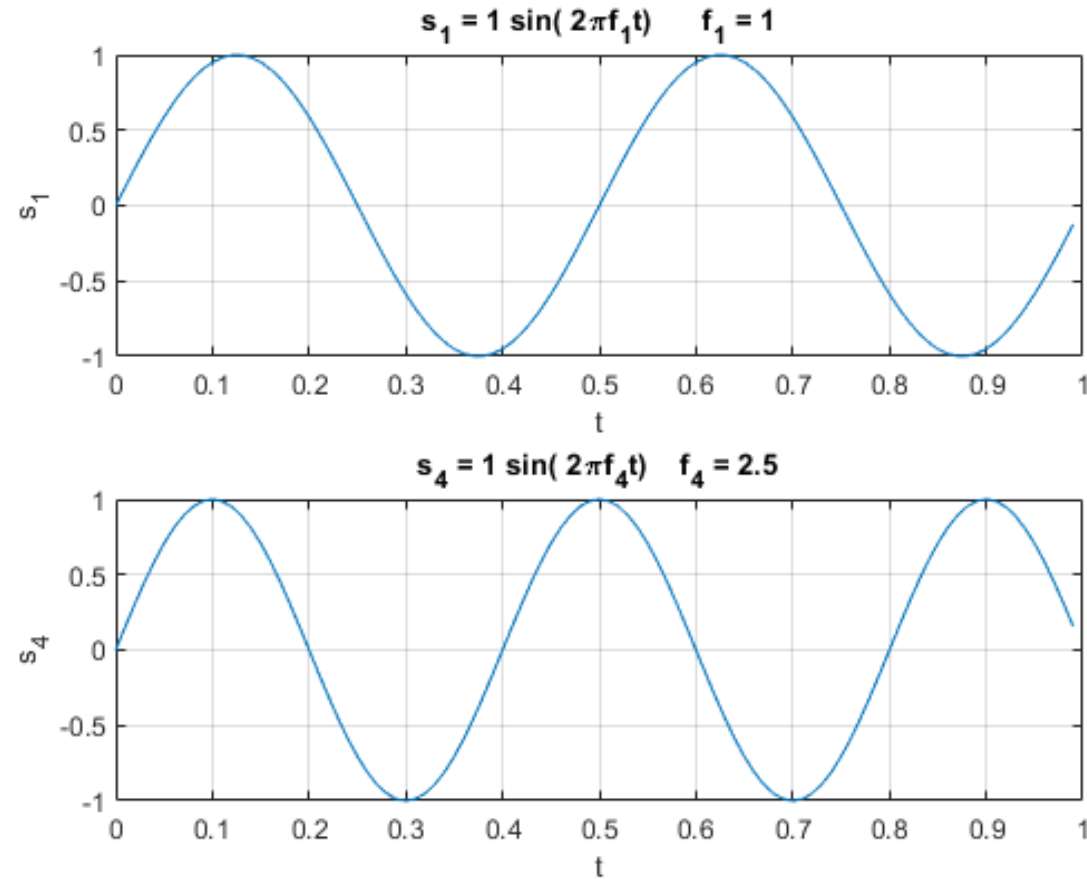
Ex0a - Signals generation and plot

- MATLAB example for sinusoidal signal generation, plot and correlation computation.

```
ex0a_orthogonal.m  +
1 - clear all
2 - close all
3 - clc
4
5 - fprintf("signal correlation \n \n");
6
7 - T=1;
8 - fs=100;
9 - Ts=1/fs;
10
11 - t=0:Ts:T-Ts;
12 - N=length(t);
13
14 %%
15
16 - A1=2;
17 - f1=1;
18
19 - s1 = A1 * sin(2 * pi * f1 * t);
```

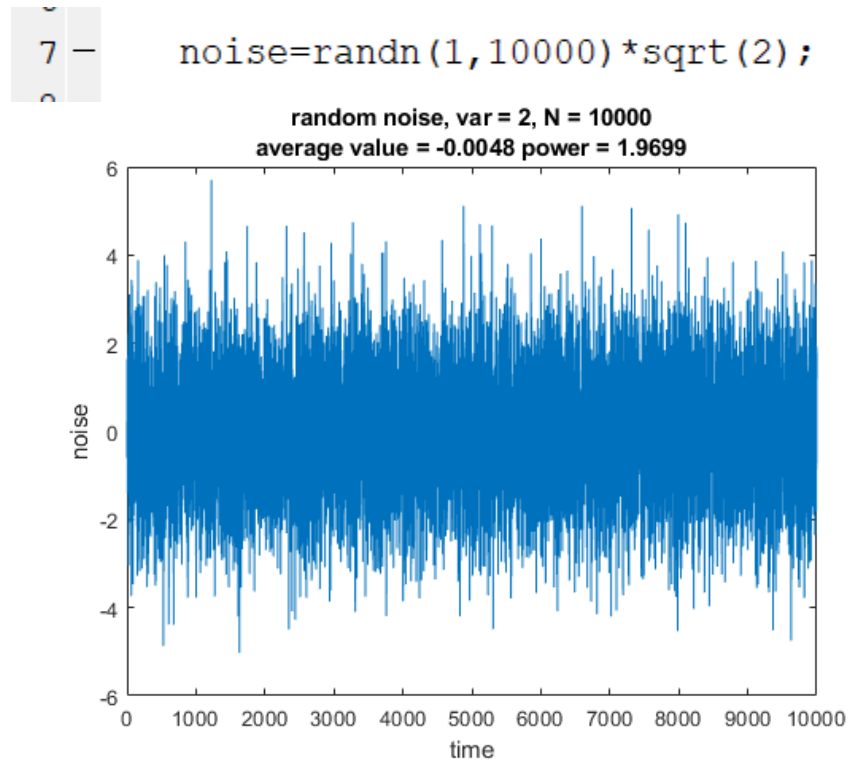


Ex0a - Signals with different frequencies



Ex0b - White Gaussian Noise

- MATLAB example for noise generation and histogram representation.



```
21  
22 - [muHat,sigmaHat] = normfit(noise)  
23  
24 - figure  
25 - histogram(noise,'Normalization','probability')
```

