Available online www.jsaer.com

Journal of Scientific and Engineering Research, **2017, 4(6):176-180**

Review Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Introduction to MATLAB for Researchers and Engineering Students

Onwusuru I[1], Okolo Chidiebere C[1], Uka Chukwuma[2], Ohuabunwa Austine[1], Nwafor Chibuikem M[1]

[1]Electronics Development Institute, Awka, Anambra State, Nigeria
[2]Federal College Of Education (Technical), Umunze, Anambra State, Nigeria

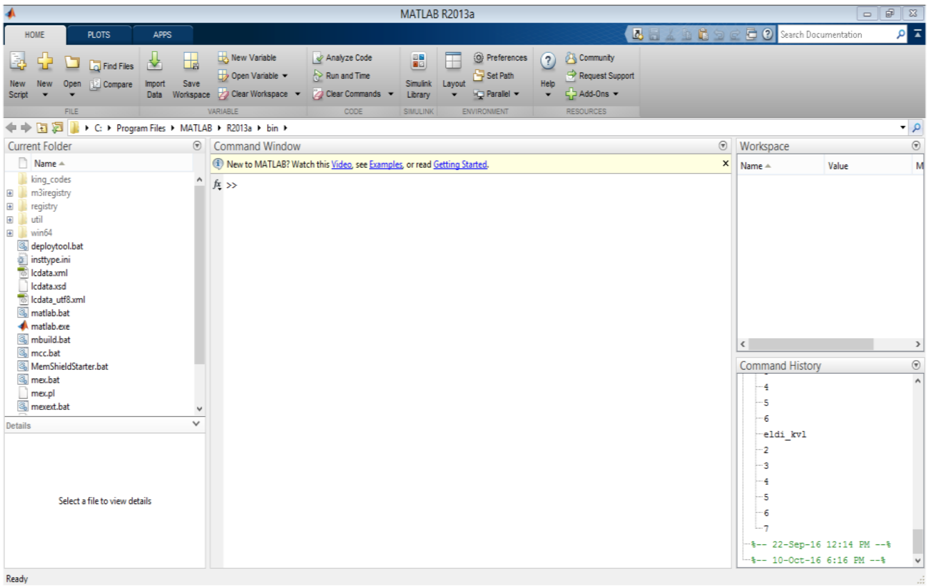
Abstract This article explainsthe technicalsoftware known as MATLAB and the simulation techniques. MATLAB has many advantages compared to most conventional computer language for solving technical problems. Is has an interactive system whose basic data element is an array that does not require dimensioning. This article gives any user a guide on how to operate the software and simulate with it as well etc.

Keywords Matlab, Simulation, Vectors, matrices

Introduction

MATLAB stands for MATrixLABoratory and the software is built up around vectors and matrices. MATLAB is a high-performance language for technical computing. This makes the software particularly useful for linear algebra but MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration. MATLAB has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language, and is one of the easiest programming languages for writing mathematical programs. MATLAB also has some tool boxes useful for signal processing, image processing, optimization, etc

The MATLAB Interface



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>>
```

Select a file to view details

```
4
5
6
eldi_kvl
2
3
4
5
6
7
%-- 22-Sep-16 12:14 PM --%
%-- 10-Oct-16 6:16 PM --%
```

Figure 1: A MATLAB Window

The MATLAB Environment

Note: From now on an instruction on pressing a certain key will be denoted by <>, e.g., pressing the enter key will be denoted as <enter>. The MATLAB environment (on most computer systems) consists of menus, buttons and a writing area similar to an ordinary word processor. There are plenty of help functions that you are encouraged to use. The writing area that you will see when you start MATLAB, is called the *command window*. In this window you give the commands to MATLAB. For example, when you want to run a program you have written for MATLAB you start the program in the command window by typing its name at the prompt. The command window is also useful if you just want to use MATLAB as a scientific calculator or as a graphing tool. If you write longer programs, you will find it more convenient to write the program code in a separate window, and then run it in the command window.

In the command window you will see a prompt that looks like >> . You type your commands immediately after this prompt. Once you have typed the command you wish MATLAB to perform, press <enter>. If you want to *interrupt a command* that MATLAB is running, type <ctrl> + <c>.

The commands you type in the command window are stored by MATLAB and can be viewed in the *Command History* window. To repeat a command you have already used, you can simply double-click on the command in the history window, or use the <up arrow> at the command prompt to iterate through the commands you have used until you reach the command you desire to repeat.

Useful functions and operations in MATLAB

Using MATLAB as a calculator is easy.

Example: Compute 5 sin(2.5$^{3\text{-pi}}$)+1/75. In MATLAB this is done by simply typing

```
5*sin(2.5^(3-pi))+1/75
```

at the prompt. Be careful with parentheses and don't forget to type * whenever you multiply!

Note that MATLAB is *case sensitive*. This means that MATLAB knows a difference between letters written as lower and upper case letters. For example, MATLAB will understand sin(2) but will not understand Sin(2).

Here is a table of useful operations, functions and constants in MATLAB.

Table 1: Useful operations, functions and constants in MATLAB

Operation, function or constant	MATLAB command
+ (addition)	+
- (subtraction)	-
x (multiplication)	x
/ (division)	/
|x| (absolute value of x)	abs(x)
square root of x	sqrt(x)
e^x	exp(x)
ln x (natural log)	log(x)
$\log_{10}$ x (base 10 log)	$\log_{10}$(x)
Sin x	Sin x
Cos x	Cos x
Tan x	Tan x
Cot x	Cot x
arcsin x	asin x
arccos x	acos x
arctan x	atan x
arccot x	acot x
n! (n factorial)	gamma(n+1)
e (2.71828...)	exp(1)
π (3.14159265...)	pi
i (imaginary unit, sqrt(-1))	i

Variables In MATLAB

We can easily define our own variables in MATLAB. Let's say we need to use the value of 3.5sin(2.9) repeatedly. Instead of typing 3.5*sin(2.9)over and over again, we can denote this variable as x by typing the following:

```
x=3.5*sin(2.9)
```

Now type

```
x+1
```

and observe what happens. Note that we did not need to declare x as a variable that is supposed to hold a floating point number as we would need to do in most programming languages.

Often, we may not want to have the result of a calculation printed-out to the command window. To suppress this output, we put a semi-colon at the end of the command; MATLAB still performs the command in "the background". If you defined x as above, now type

```
y=2*x;
y
```

And observe what happened.

In many cases we want to know what variables we have declared. We can do this by typing *whos*. Alternatively, we can view the values by opening the "Workspace" window. This is done by selecting the Workspace option from the View menu. If you want to erase all variables from the MATLAB memory, type clear. To erase a specific variable, say x, type clear x. To clear two specific variables, say x and y, type clear x y, that is separate the different variables with a space. Variables can also be cleared by selecting them in the Workspace window and selecting the delete option.

Vectors and matrices in MATLAB

We create a vector in MATLAB by putting the elements within [] brackets.

Example: x=[1 2 3 4 5 6 7 8 9 10]

We can also create this vector by typing x=1:10. The vector (1 1.1 1.2 1.3 1.4 1.5) can be created by typing x=[1 1.1 1.2 1.3 1.4 1.5] or by typing x=1:0.1:1.5.

Matrices can be created according to the following example. The matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

is created by typing

A=[1 2 3 ; 4 5 6; 7 8 9],

i.e., rows are separated with semi-colons.

How to Plot with MATLAB

There are different ways of plotting in MATLAB. The following two techniques, illustrated by examples, are probably the most useful ones.

Example 1: Plot sin(x^2) on the interval [-5,5]. To do this, type the following:

```
x=-5:0.01:5;
y=sin(x.^2);
plot(x,y)
```

and observe what happens.

A Sample MATLAB Program

```
r1 = 1; %input('input resistor 1 ');
r2 = 2; %input('input resistor 2 ');
r3 = 3; %input('input resistor 3 ');
r4 = 4; %input('input resistor 4 ');
r5 = 5; %input('input resistor 5 ');
```

```
r6 = 12; %input('input resistor 6 ');

A = [r1+r2 -r2 0; -r2 r2+r3+r4 -r4; 0 -r4 r4+r5+r6];
V = [20; 0; 0]; %input voltage is 20
i = A\V;

current1 = i(1);
current2 = i(2);
current3 = i(3);

vA = (current1 - current2)*r2
vB = (current2 - current3)*r4
vC = (current3)*r6

total_voltage = (current1*r1)+((current1 - current2)*r2)+current2*r2+((current2 -
current3)*r4)+current3*r5+current3*r6

v = 0:5:100;
k = length(v);
for k = 1:length(v)
    V = [v(k);0;0]
i = A\V
vC(k) = i(3)*r6
end
subplot(2,2,3)
plot(vC,v)
holdon
gridon
title('Response of Pi Resistive Network');
xlabel('Output Voltage');
ylabel('Input Voltage');
```

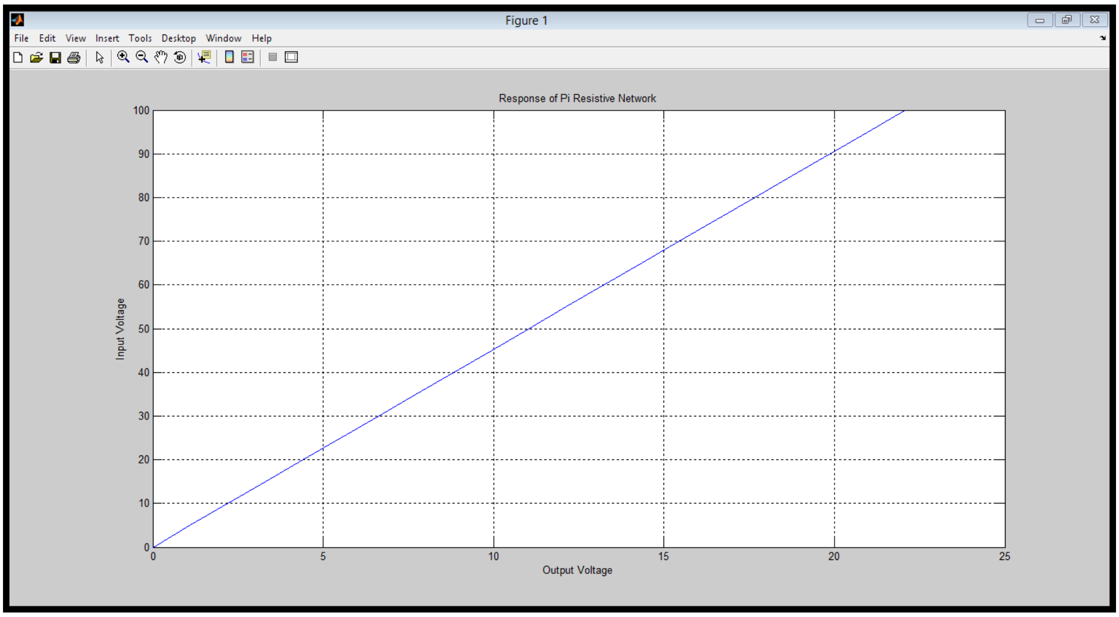


Figure 1
File Edit View Insert Tools Desktop Window Help
Response of Pi Resistive Network
Input Voltage
Output Voltage
0
10
20
30
40
50
60
70
80
90
100
0
5
10
15
20
25

Figure 2: The Result of the Above Sample Program

The following commands are useful when plotting:

Table 2: Useful commands when plotting

Graphing functions	**MATLAB command**
Label the horizontal axis.	xlabel('text')
Label the vertical axis.	ylabel('text')
Attach a title to the plot.	title('text')
Change the limits on the x and y axis.	axis([xminxmaxyminymax])
"Keep plotting in the same window."	hold on
Turn off the "keep-plotting-in-the-same-window-command".	hold off

Note that all text must be put within ''. The last two commands (hold on and hold off) are best explained by trying them next time you plot

Conclusion

MATLAB integrates computation, visualization, and programming in an easy to use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB has also evolved over a period of years with input from many users. It is the standard instructional tool for introductory and advanced courses in mathematics, engineering and sciences in the university environment while in the industry; it is the tool of choice for high-productivity research, development and analysis.

References

[1]. J. W. Demmel. Applied Numeric LINEAR Algebra. Siam, 1997.

[2]. S. J. Chapman. MATLAB Programming for Engineers. Thomson, 2004.

[3]. A. Gilat. MATLAB: An introduction withApplications. John Willey and Sons, 2004.

[4]. David Houcque. Introduction to MATLAB for Engineering Students. Version 1.2, August 2005.