TERM: M1-RAIA

SEMESTER: 1

AY: 2022-2023

Abdelbacet Mhamdi

Dr.-Ing. in Electrical Engineering

Senior Lecturer at ISET Bizerte

abdelbacet.mhamdi@bizerte.r-iset.tn

ARTIFICIAL INTELLIGENCE - PART 1

LAB MANUAL

BIZERTE
RESEAU
INSTITUT SUPÉRIEUR
DES ÉTUDES
TECHNOLOGIQUES

Higher Institute of Technological Studies of Bizerte

Available at https://github.com/a-mhamdi/isetbz/

Honor code

The University of North Carolina at Chapel Hill
Department of Physics and Astronomy
http://physics.unc.edu/undergraduate-program/labs/general-info/

"During this course, you will be working with one or more partners with whom you may discuss any points concerning laboratory work. However, you must write your lab report, in your own words.

Lab reports that contain identical language are not acceptable, so do not copy your lab partner's writing.

If there is a problem with your data, include an explanation in your report. Recognition of a mistake and a well-reasoned explanation is more important than having high-quality data, and will be rewarded accordingly by your instructor. A lab report containing data that is inconsistent with the original data sheet will be considered a violation of the Honor Code.

Falsification of data or plagiarism of a report will result in prosecution of the offender(s) under the University Honor Code.

On your first lab report you must write out the entire honor pledge:

The work presented in this report is my own, and the data was obtained by my lab partner and me during the lab period.

On future reports, you may simply write *"Laboratory Honor Pledge"* and sign your name."

Contents

In order to activate the virtual environment and launch **Jupyter Notebook**, we recommend you to proceed as follow

① Press simultaneously the keys ⊞ & R on the keyboard. This will open the dialog box `Run`;

② Then enter `cmd` in the command line and confirm with ↵ key on the keyboard;

③ Type the instruction `jlai.bat` in the console prompt line;

```
Command Prompt

C:\Users\admin> jlai.bat
```

④ Finally press the ↵ key.

Leave the system console active.

1 | *Julia* Onramp

Student's name	. . . .	. . . .	. . . .
Score ***/20***	. . . .	. . . .	. . . .

Detailed Credits

Anticipation ***(4 points)***	. . . .	. . . .	. . . .
Management ***(2 points)***	. . . .	. . . .	. . . .
Testing ***(7 points)***	. . . .	. . . .	. . . .
Data Logging ***(3 points)***	. . . .	. . . .	. . . .
Interpretation ***(4 points)***	. . . .	. . . .	. . . .

Goals

★ Learn the essentials of *Julia* on commonly used features & workflows.

❗ The notebook is available at `https://github.com/a-mhamdi/cosnip/` ⟶ *Julia* ⟶ *julia-onramp.ipynb*

2 | Tipping Problem

Student's name	. . . .	. . . .	. . . .
Score ***/20***	. . . .	. . . .	. . . .

Detailed Credits

Anticipation ***(4 points)***	. . . .	. . . .	. . . .
Management ***(2 points)***	. . . .	. . . .	. . . .
Testing ***(7 points)***	. . . .	. . . .	. . . .
Data Logging ***(3 points)***	. . . .	. . . .	. . . .
Interpretation ***(4 points)***	. . . .	. . . .	. . . .

Goals

⋆ Construct algorithms to help decide in given ambiguous situation.

The code is available at https://github.com/a-mhamdi/isetbz/ → *Artificial Intelligence* → *Codes* → *Matlab* → *Tipper.∗*

Matlab is an interesting tool for engineers. All you need in *Matlab* is the basics pieces of information needed to solve the problem. There is no need to configure your environment to adopt your algorithm, just know a little bit about the syntax. It is an interactive matrix calculator, with a full-fledged programming environment. *Matlab* allows to handle with general tasks and very delicate purpose. The main focus with is what you need to program not how to program it.

Matlab is an easy to use environment, it is a fourth-generation programming language[1] (4GL). It is a very high level language.

[1] http://en.wikipedia.org/wiki/Fourth-generation_programming_language

Matlab desktop

Desktop layout

Like any other environment, *Matlab* has a menubar and a toolbar where user can find the major commands needed to configure his/her preferences, this means how to let *Matlab* appear or behave. Then, the main interface is split into four majors areas as shown in Fig. **??**.

Area 1

it is the Command Window, where user can strike the Matlab commands and see results displayed on the same screen.

Area 2

It is the Workspace. All variables are saved in this area. If you need to know more about saved variables, just try this command on area 1:

```
>> whos
```

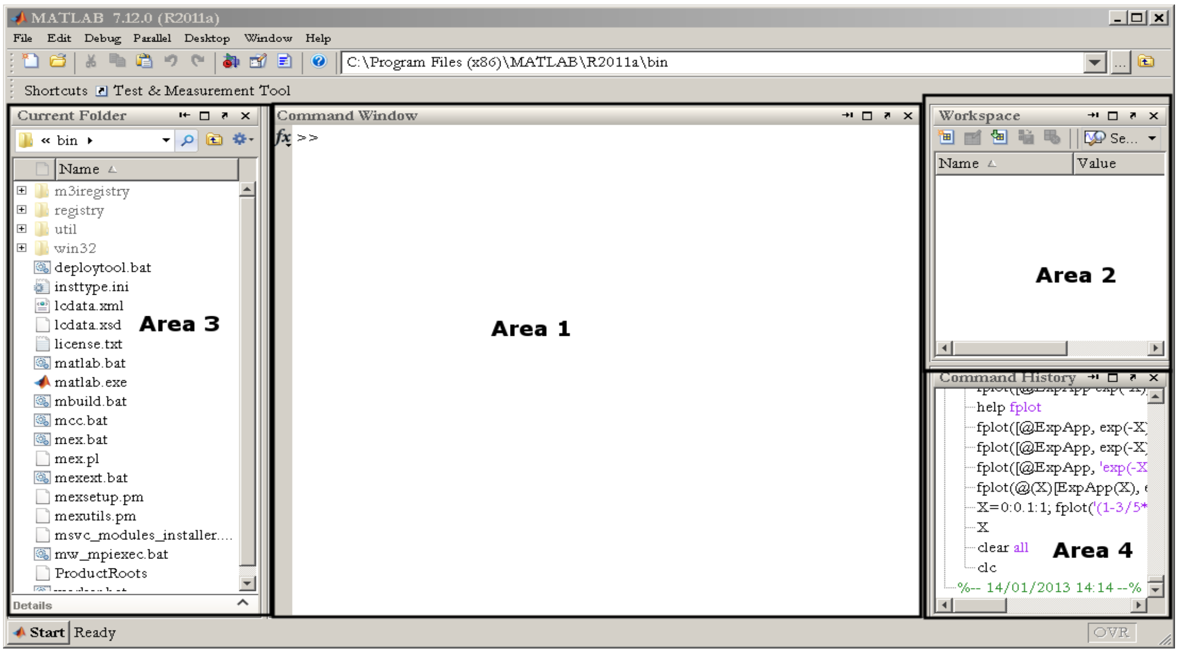
The results are the names of variables, the min and max values for every variables, class and the number of bytes needed to save it.

Area 3

It is the Current Folder, the path indicated by Matlab to execute a particular code. if you need to execute an algorithm which is not on the path shown by are 4, you have then to change the folder. Otherwise, an error message will pop-up on command window if it is not the absolute path.

Area 4

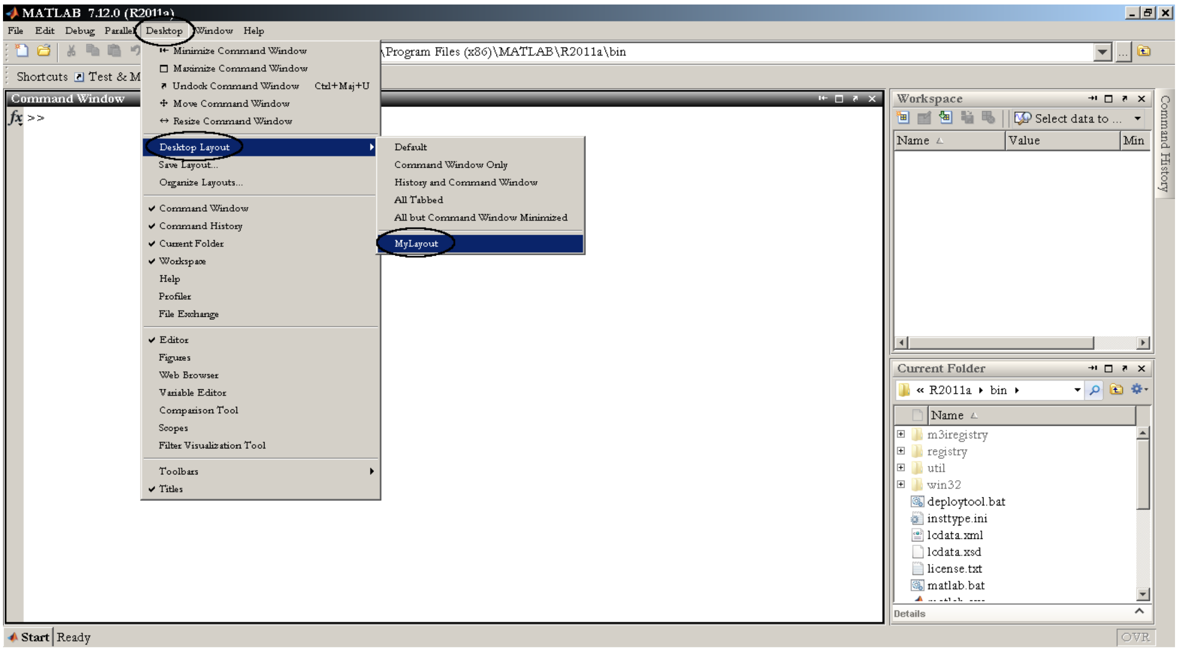
It is the Command History, where all instructions are saved in a panel from the date of installing Matlab on your machine until current time.



```
help fplot
fplot([@ExpApp, exp(-X
fplot([@ExpApp, exp(-X
fplot([@ExpApp, 'exp(-X
fplot(@(X)[ExpApp(X), e
X=0:0.1:1; fplot('(1-3/5*
X
clear all
clc
%-- 14/01/2013 14:14 --%
```

2.0.1 How to personalize the desktop layout

In the menu bar, click on Desktop, Desktop Layout then load your preferences. For example, here the customized layout was saved under the name MyLayout. You just click on it, and your layout appears like the display shown on the following image.

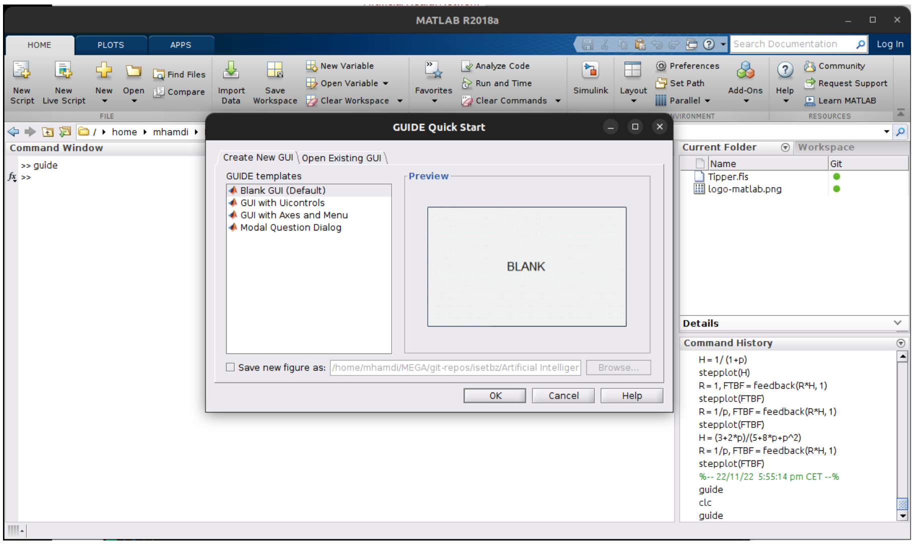


Matlab is not just an environment for technical computing, where we can solve equations. It is also an environment of graphical interfaces development. This integrated software is called **"GUIDE"**, Graphical User Interface Development Environment.

Try, at the command window, the instruction:

```
>> guide
```

or simply move your mouse on the toolbar, you will have a shortcut like the icon . Just click on it, you will then launch the environment of graphics. You will get on your screen, the dialog shown here. You can access the templates or simply click next.



```
>> guide
>>
```

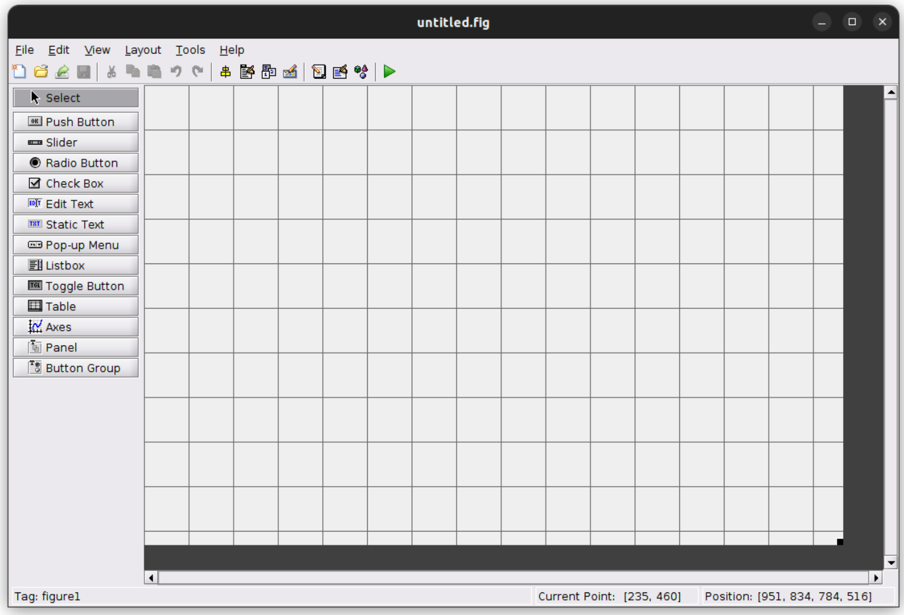
GUIDE templates
- Blank GUI (Default)
- GUI with Uicontrols
- GUI with Axes and Menu
- Modal Question Dialog

□ Save new figure as: /home/mhamdi/MEGA/git-repos/isetbz/Artificial Intelliger

Name	Git
Tipper.fis	●
logo-matlab.png	●

```
H = 1/ (1+p)
stepplot(H)
R = 1, FTBF = feedback(R*H, 1)
stepplot(FTBF)
R = 1/p, FTBF = feedback(R*H, 1)
stepplot(FTBF)
H = (3+2*p)/(5+8*p+p^2)
R = 1/p, FTBF = feedback(R*H, 1)
stepplot(FTBF)
%-- 22/11/22  5:55:14 pm CET --%
guide
clc
guide
```

Now, the environment is ready to be used.

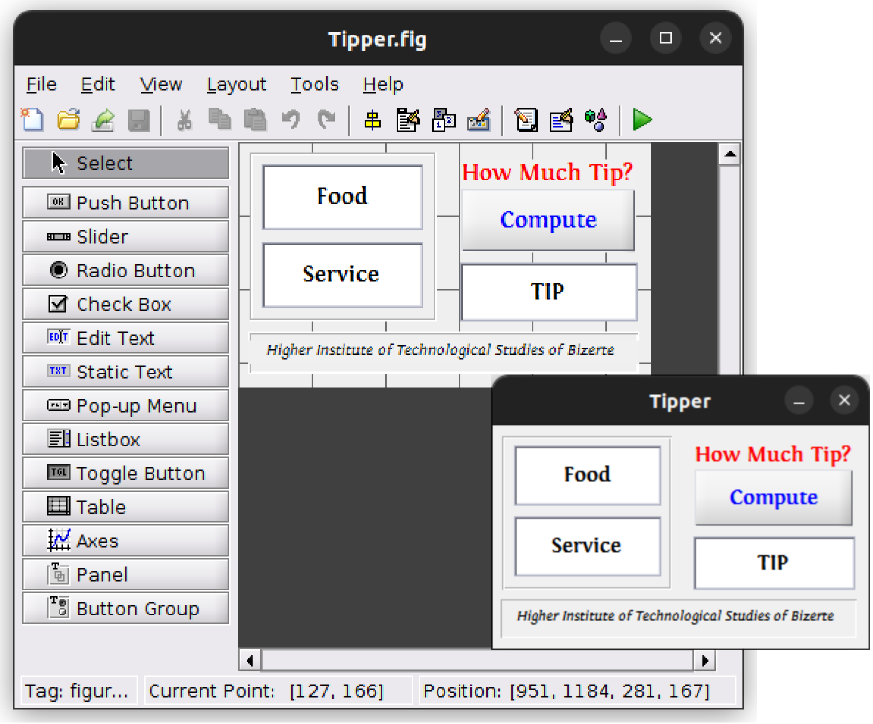


Save your work, you will notice the creation of two files on you directory with the same name, with two different extensions, one has this termination *\.*fig* an the second one is an *m-file*.

The Fig-file contains the graphical declaration or the instantiation of the all used graphical objects in your interface. The main GUI that appears to user in order to customize his interface contains a palette where you can easily access the preconfigured objects like “pushbutton”, “toggle button” and “slider”. You can even add some external objects via the activeX control which is not in our current scope. Try to place and organize the required objects.

Notice that you have for any object an Object Browser which, when pointed on an object, list all properties relative to this one and even the set of methods called when evoking it. You may also need to know that there exist an uipanel, use it in case of you need to regroup elements by similarities.

Now, you can launch your application. Your conceived interface may look like this one indicated below.



It is preferable to load the fuzzy inference system, denoted hereafter by *fis*, in the opening function of the gui.

```
% --- Executes just before Tipper is made visible.
function Tipper_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Tipper (see VARARGIN)
fis =readfis('Tipper.fis');
handles.fis = fis;
```

Before, quitting the function, you have to update the GUI and save all variable in the general structure with handles and user data[2].

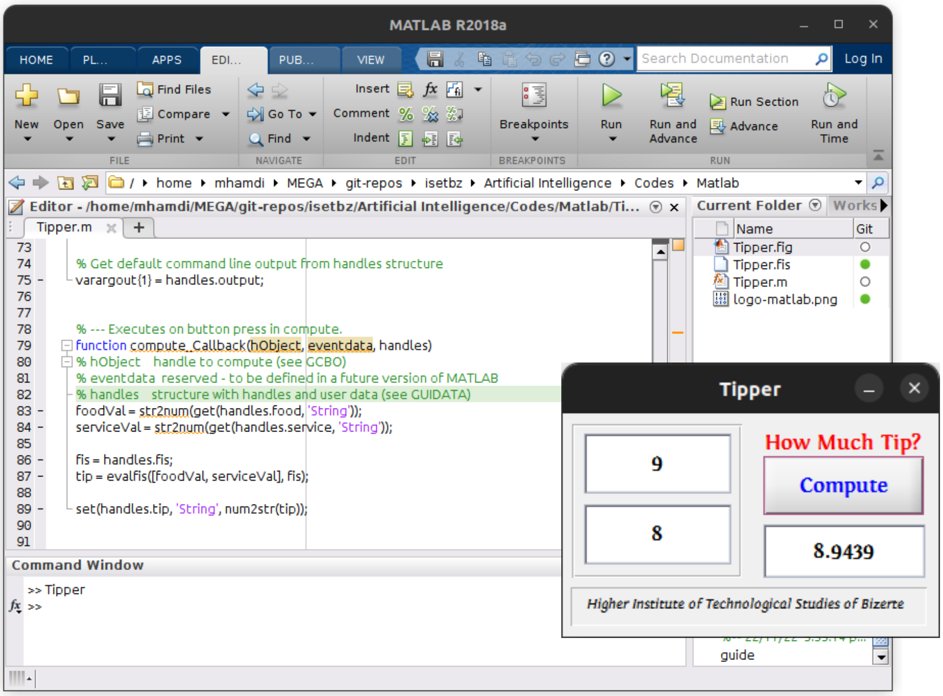
```
% Choose default command line output for Tipper
handles.output = hObject;

% Update handles structure
```

[2] Visit: http://www.mathworks.com/help/matlab/gui-building-basics.html

```
guidata(hObject, handles);

% UIWAIT makes Tipper wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Executes on button press in compute.
function compute_Callback(hObject, eventdata, handles)
% hObject    handle to compute (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
foodVal = str2num(get(handles.food, 'String'));
serviceVal = str2num(get(handles.service, 'String'));

fis = handles.fis;
tip = evalfis([foodVal, serviceVal], fis);

set(handles.tip, 'String', num2str(tip));
```

The ultimate application will behave like the example displayed on the follwoing figures. Enter your desired values and then just click **"Compute"**. The result will be displayed on right half of the interface.



```
    % Get default command line output from handles structure
    varargout{1} = handles.output;


    % --- Executes on button press in compute.
    function compute_Callback(hObject, eventdata, handles)
    % hObject    handle to compute (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    foodVal = str2num(get(handles.food, 'String'));
    serviceVal = str2num(get(handles.service, 'String'));

    fis = handles.fis;
    tip = evalfis([foodVal, serviceVal], fis);

    set(handles.tip, 'String', num2str(tip));
```

```
>> Tipper
>>
```

9

8

8.9439

The code is available at https://github.com/a-mhamdi/isetbz/ → *Artificial Intelligence* → *Codes* → *Julia* → *tipper.jl*

```julia
using Fuzzy
using Plots

score = range(0, 10, length=100)

food = Dict(
   "Rancid" => TrapezoidalMF(0, 0, 2, 4),
   "Delicious" => TrapezoidalMF(6, 8, 10, 10)
   )
food_chart = chart_prepare(food, score)

service = Dict(
   "Poor" => TrapezoidalMF(0, 0, 2, 4),
   "Good" => TrapezoidalMF(3, 4, 6, 7),
   "Excellent" => TrapezoidalMF(6, 8, 10, 10)
   )
service_chart = chart_prepare(service, score)

tip = Dict(
   "Cheap" => TrapezoidalMF(0, 0, 1, 3),
   "Average" => TrapezoidalMF(2, 4, 6, 8),
   "Generous" => TrapezoidalMF(7, 9, 10, 10)
   )
tip_chart = chart_prepare(tip, score)

rule_1 = Rule(["Rancid", "Poor"], "Cheap", "MAX")
rule_2 = Rule(["", "Good"], "Average", "MAX")
rule_3 = Rule(["Delicious", "Excellent"], "Generous", "MAX")

rules = [rule_1, rule_2, rule_3]

#= GRAPHS =#
p1 = plot(score, food_chart["values"], ylabel="Food", label=food_chart["names"], legend=:bottomright)

p2 = plot(score, service_chart["values"], ylabel="Service", label=service_chart["names"], legend=:bottomright)

p3 = plot(score, tip_chart["values"], xlabel="Score", ylabel="Tip", label=tip_chart["names"], legend=:bottomright)
```

```

graphs = plot(p1, p2, p3, layout=(3, 1), lw=2)
savefig(graphs, "mf-graphs.pdf")

# FUZZY INFERENCE SYSTEM: MAMDANI
fis = FISMamdani([food, service], tip, rules)
eval_fis(fis, [9., 8.])
```

3 Selection Process - Case of RAIA

Student's name	. . . .	. . . .	. . . .
Score ***/20***	. . . .	. . . .	. . . .

Detailed Credits

Anticipation ***(4 points)***	. . . .	. . . .	. . . .
Management ***(2 points)***	. . . .	. . . .	. . . .
Testing ***(7 points)***	. . . .	. . . .	. . . .
Data Logging ***(3 points)***	. . . .	. . . .	. . . .
Interpretation ***(4 points)***	. . . .	. . . .	. . . .

Goals

- ★ Design a fuzzy system to pick the most adequate candidates out of the applicants to the master program **RAIA**.

The notebook is available at `https://github.com/a-mhamdi/cosnip/` → *Julia* → *fuzzy* → *selection-process.jl*

4 Fuzzy Control of an Articulated System

Student's name	. . . .	. . . .	. . . .
Score ***/20***	. . . .	. . . .	. . . .
Detailed Credits			
Anticipation ***(4 points)***	. . . .	. . . .	. . . .
Management ***(2 points)***	. . . .	. . . .	. . . .
Testing ***(7 points)***	. . . .	. . . .	. . . .
Data Logging ***(3 points)***	. . . .	. . . .	. . . .
Interpretation ***(4 points)***	. . . .	. . . .	. . . .

Goals

- ★ Design and implement a fuzzy regulator to control the position of a disk.

The repository is available at `https://github.com/a-mhamdi/ball_and_beam/`

5 | Binary Classifier using ANN

Student's name	. . . .	. . . .	. . . .
Score */20*	. . . .	. . . .	. . . .

Detailed Credits

Anticipation ***(4 points)***	. . . .	. . . .	. . . .
Management ***(2 points)***	. . . .	. . . .	. . . .
Testing ***(7 points)***	. . . .	. . . .	. . . .
Data Logging ***(3 points)***	. . . .	. . . .	. . . .
Interpretation ***(4 points)***	. . . .	. . . .	. . . .

Goals

★ Train an artificial neural network to predict and classify categorical outcomes.

The notebook is available at `https://github.com/a-mhamdi/cosnip` → *Julia* → *nnet* → *clf-ann.jl*

The overall scope of this manual is to introduce **Artificial Intelligence (AI)** , through some numeric simulations, to the students enrolled at the master's program **RAIA**.

The topics discussed in this manuscript are as follow:

① Getting started with *Julia*

Get familiar with *Pluto* Notebook.

② Fuzzification, inference system & defuzzification

Membership functions; COG.

③ System control using fuzzy logic

④ How to build an ANN

Julia; REPL; *Pluto*; *Fuzzy*; *Flux*; CUDA; *Matlab*; artificial intelligence; system control; fuzzy; inference; ann.