Zagazig University
Faculty of Engineering
Computer and Systems Eng. Dept.

جامعة الزقازيق
كلية الهندسة
قسم هندسة الحاسبات والمنظومات

Workspace simulation using Matlab GUI

1. Introduction:

Workspace (Work Envelope): The Workspace of the manipulator is the total volume swept out by the end effector as the manipulator executes all possible motion. In this Lab, we are going to explore the simulation of 2D manipulator of RR, RP type. And a 3D design of RRR and RRP type.

2. Objectives

- Learn basic GUI principles in Matlab.
- Simulate 2D and 3D workspace
- Determine whether a point is reachable by the robot end effector using the simulated workspace.

3. Requirements

- Matlab 2012 (or Higher compatible versions)

Prerequisites:

- Basic Matlab Programming skills
- Basic knowledge of Robotics general concepts.

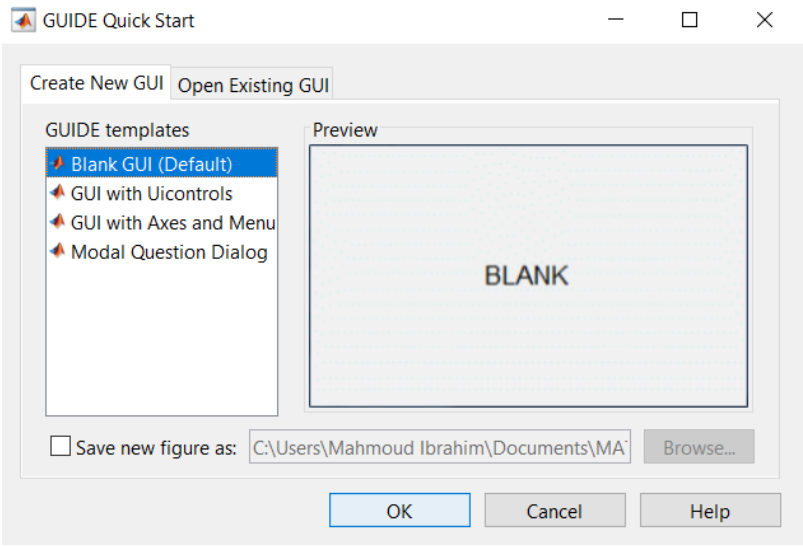
4. Procedure

Example 0: Design a Matlab GUI that implements a linear function y = x when a Draw button is pressed.

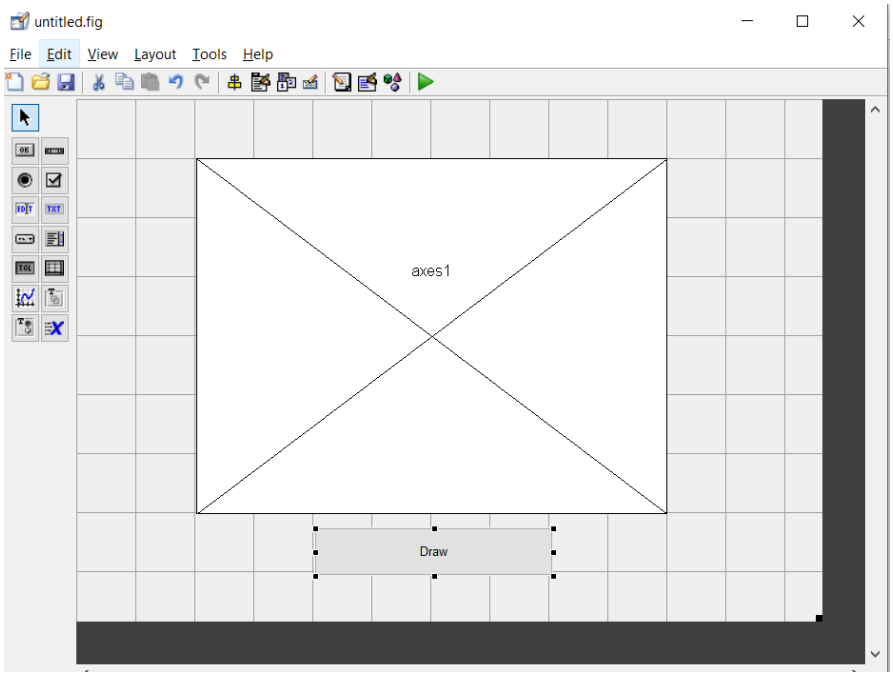
Steps:

1- In the Matlab command window type guide and then press enter.

2 – From Create New GUI section choose Blank GUI and press OK.



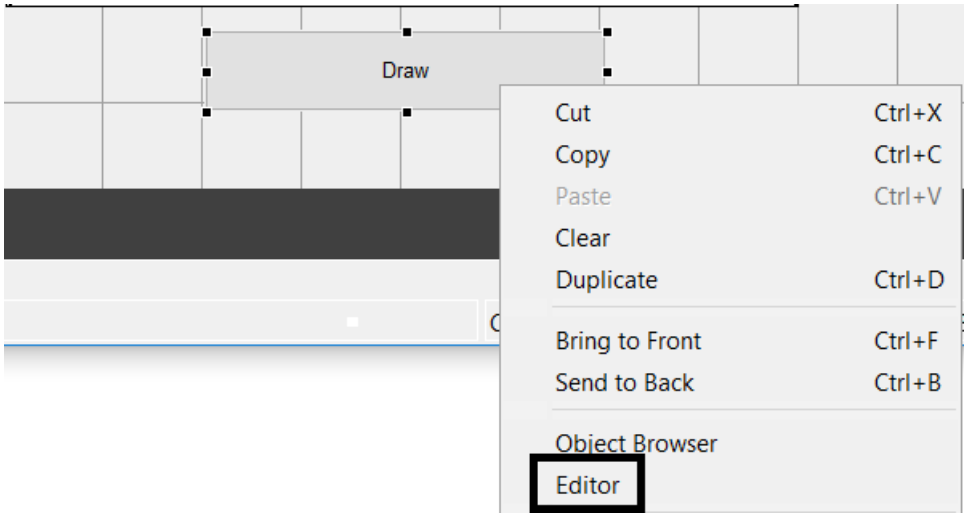
3- Choose the Pushbutton and Axes from the left side to form the following shape.



4-Double click the PushButton And change the String property value to “Draw”.

5- From your keyboard press Ctrl + S to save the Project with appropriate name e.g. (“Example0”). Don’t use any spaces or special symbols.

6- Right click the Draw button and choose editor to go to the Code.



7- To execute an action when the Draw button is pressed, Matlab uses what’s called a callback function (e.g. `pushbutton1_Callback`). For each object (e.g. Button), there’s a separate callback function where the required code should be placed.

8- add the following code after this line

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

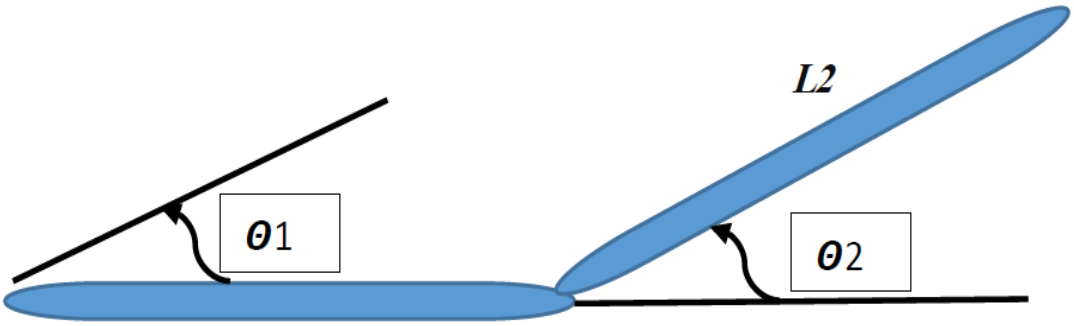
like this

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
x=1:10; % x axis data
y=1:10; % y axis data
plot(handles.axes1,x,y) % plot command in the axes1 object
```

---

you could use the plot (x , y) command as well but it’s useful to know that the plot function could be overloaded with the axes name in case there were several axes in other situations.

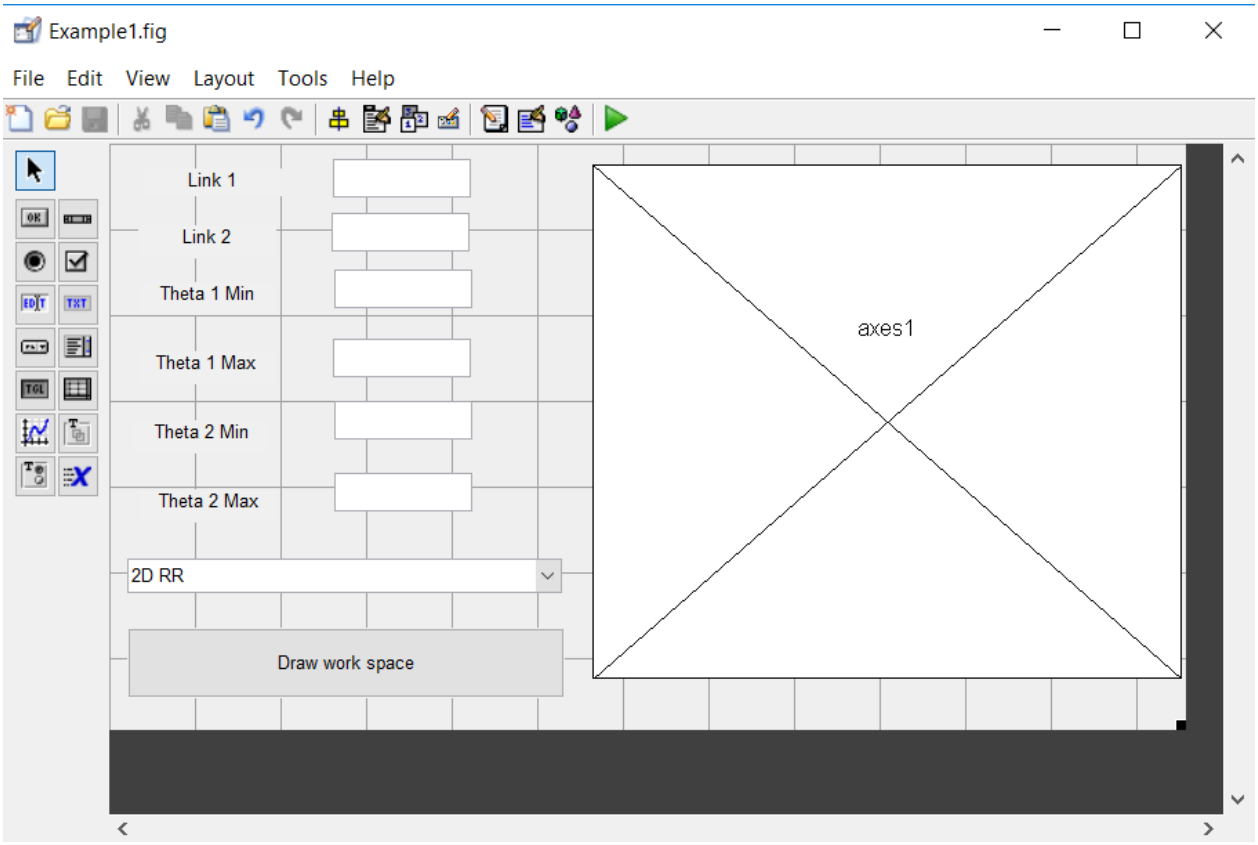
Example 1: Draw a simulation of an RR robot GUI with L1 and L2 lengths given to the user from an Edit Text (Text Box). Also, theta 1 and theta 2 Boundaries are both given.



Solution

Steps:

1- Arrange the components like that (PopUpMenu – PushButton – EditText – Static text – axes)



2- Configure the Objects with suitable properties:

Object	*Property*	*Value*
First link (Label)	**String**	Link 1
	Tag	STLink1
First link (Text Box)	**String**	
	Tag	ETLink1
***Second* link (Label)**	**String**	Link 2
	Tag	STLink2
***Second* link (*Text Box*)**	**String**	
	Tag	ETLink2
1st Rev. Angle min. value (label)	**String**	Theta 1 Min
	Tag	STTheta1Min
1st Rev. Angle min. value (Text Box)	**String**	
	Tag	ETTheta1Min
1st Rev. Angle max. value (label)	**String**	Theta 1 Max
	Tag	STTheta1Max
1st Rev. Angle max. value (Text Box)	**String**	
	Tag	ETTheta1Max
2nd Rev. Angle min. value (label)	**String**	Theta 2 Min
	Tag	STTheta2Min
2nd Rev. Angle min. value (Text Box)	**String**	
	Tag	ETTheta2Min
2nd Rev. Angle max.	**String**	

value (label)	**Tag**	STTheta2Max
2nd Rev. Angle max. value (Text Box)	**String**	
	Tag	ETTheta2Max
Pop Up Menu	**String**	2D RR 2D RP
	Tag	DOFType
Push Button	**String**	Draw workspace
	Tag	BtnWorkSpaceDraw

Note:

*when typing the Pop Up menu string you'll press (Ctrl + enter) to go to the next line.

** The String property: Contains the name displayed in the GUI in running time.

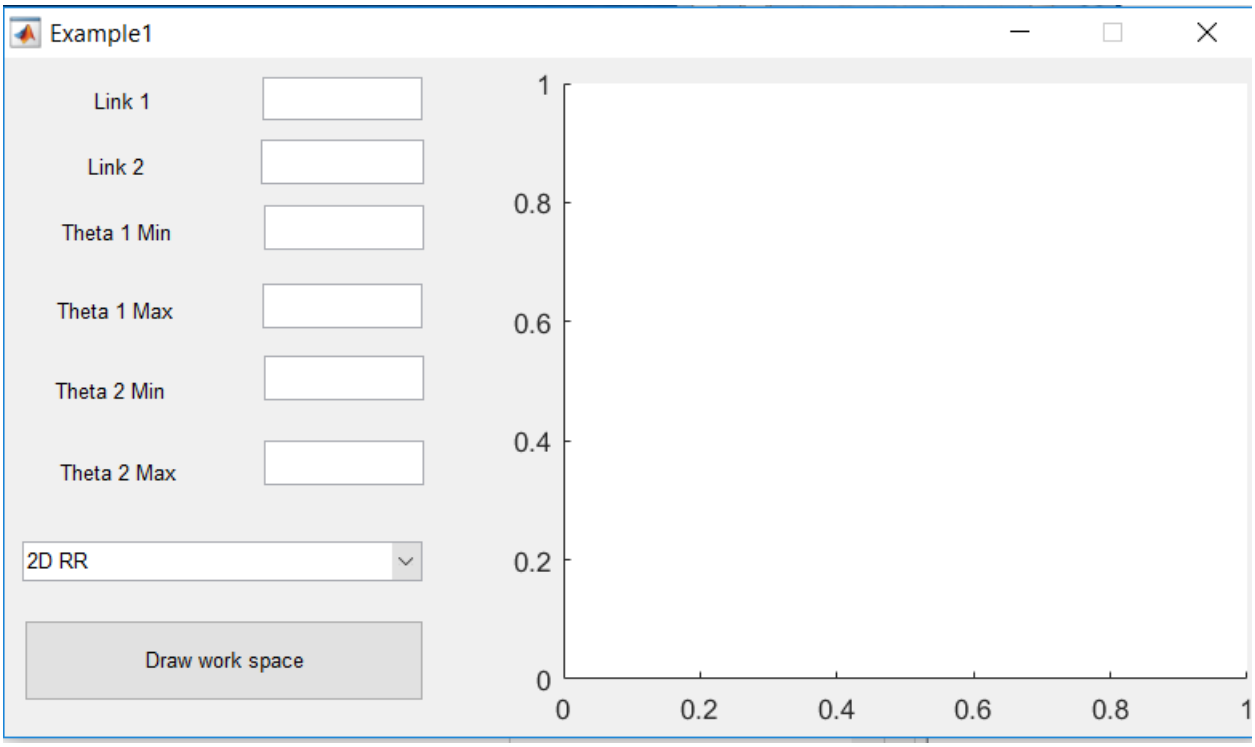
*** The tag property: Contains the name used while coding.

3- Then return to the Editor menu and type the following code after

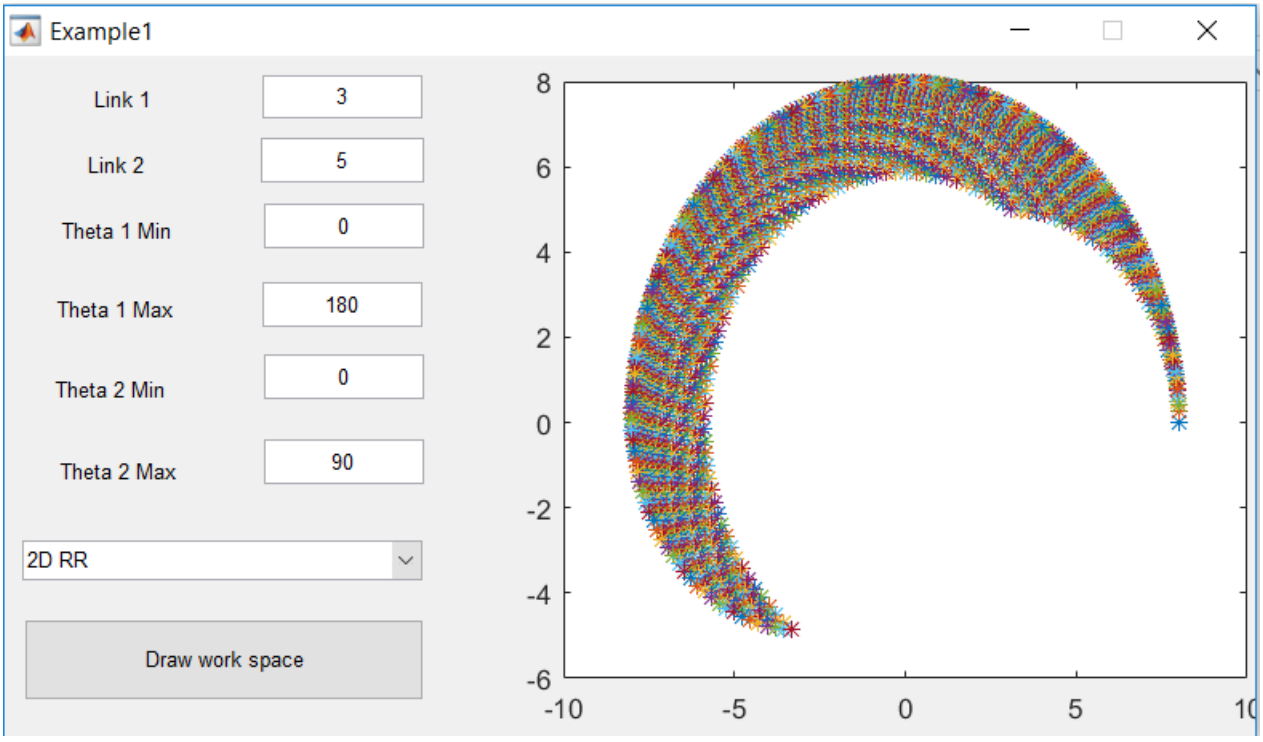
```
function BtnWorkSpaceDraw_Callback(hObject, eventdata, handles)
```

```
if handles.DOFType.Value == 1 % Executes when POP Up Menu
Equals RR
cla(handles.axes1) % clear any previous drawing on the axes
Drawing_Step=.05;  % Assign a fixed drawing step
%The following lines are used to Read the values in GUI Edit
Boxes
l1=str2num(handles.ETLink1.String);
l2=str2num(handles.ETLink2.String);
theta1min=pi*str2num(handles.ETTheta1Min.String)/180;
theta1max=pi*str2num(handles.ETTheta1Max.String)/180;
theta2min=pi*str2num(handles.ETTheta2Min.String)/180;
theta2max=pi*str2num(handles.ETTheta2Max.String)/180;
% calculate The Value of Each point
for count1 =theta1min:Drawing_Step:theta1max
for count2 =theta2min:Drawing_Step:theta2max
x=l1*cos(count1)+l2*cos(count1+count2);
y=l1*sin(count1)+l2*sin(count1+count2);
plot(handles.axes1,x,y,'*')
hold on
end end end
```

4 – Run the code and you should have an output similar to this:



5 – Try inserting values to the Edit text and press the “Draw workspace” button.



6- Go to task 1 and code for the second option in pop up menu (RP type).

Example 2: Draw the Reachable workspace by the following robot knowing that it's an RRR robot and the base joint can rotate horizontally by a max angle given by the user. Use your GUI from Example 1 to add a third option for a 3D workspace.

Solution

Steps:

1- Create a new static box with String value "Theta 0 Max" and an Empty Edit box.

2- make both of the previous boxes hidden from Visible Property

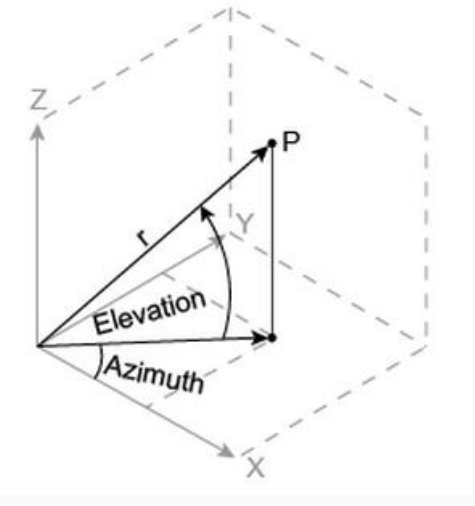
3- To make them appear when the user chooses the 3rd option right the following lines:

```
function DOFType_Callback(hObject, eventdata, handles)
if handles.DOFType.Value == 3
handles.ETTheta0Max.Visible='on';
handles.STTheta0Max.Visible='on';
end
```

4- Now all we need to do is to execute the code which would draw the workspace in 3D.

But before that remember the function **sph2cart** in Matlab which transforms the spherical to Cartesian Coordinates.

```
[x,y,z] = sph2cart(azimuth,elevation,r)
```

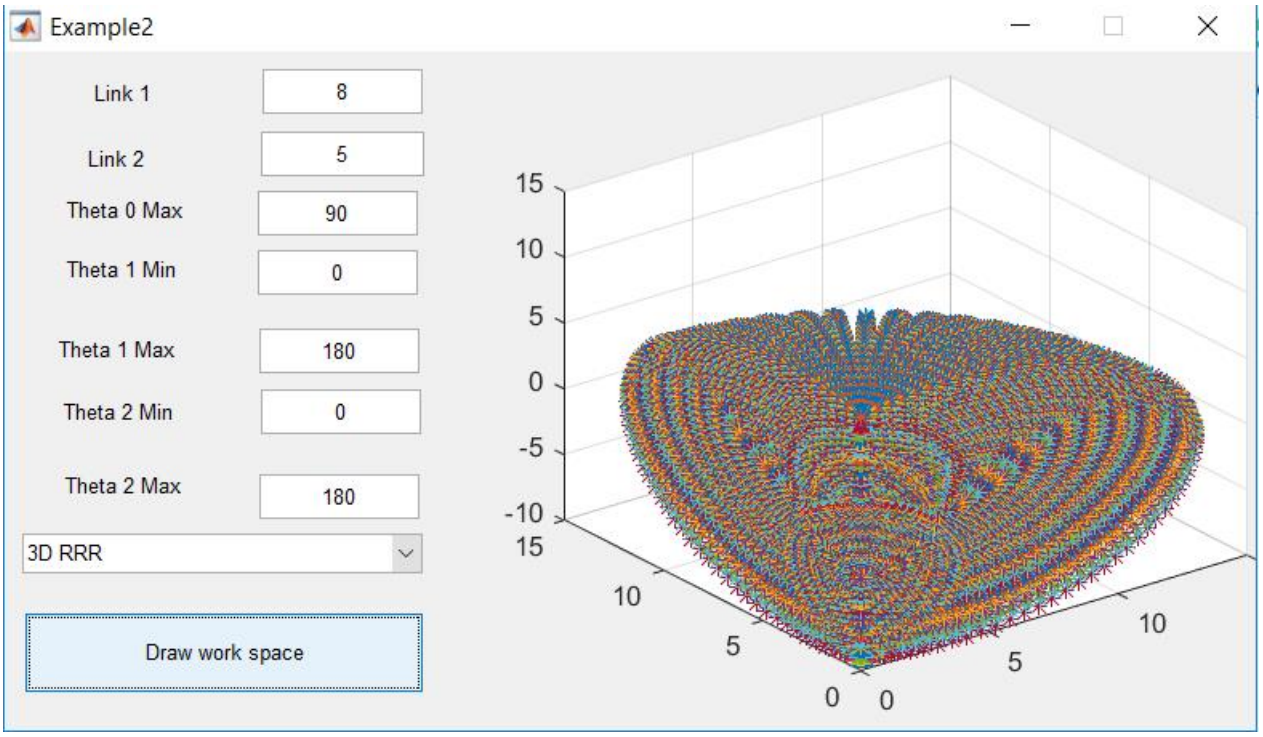


The program code is added to the end of the pushbutton callback function "BtnWorkSpaceDraw_Callback".

```
if handles.DOFType.Value == 3
%The following lines are used to Read the values in GUI Edit
Boxes
theta0Max=pi*str2num(handles.ETTheta0Max.String)/180;
l1=str2num(handles.ETLink1.String);
l2=str2num(handles.ETLink2.String);
theta1min=pi*str2num(handles.ETTheta1Min.String)/180;
theta1max=pi*str2num(handles.ETTheta1Max.String)/180;
theta2min=pi*str2num(handles.ETTheta2Min.String)/180;
theta2max=pi*str2num(handles.ETTheta2Max.String)/180;

% calculate The Value of Each point
for counter0=0:.1:theta0Max
for counter1=theta1min:.05:theta1max
 for counter2=theta2min:.05:theta2max
ElevationAngle=atan((l2*sin(counter1+counter2)+l1*sin(counter
1))/(l1*cos(counter1)+l2*cos(counter1+counter2)));
r=sqrt(l1^2+l2^2-2*l1*l2*cos(pi-(counter1+counter2)));
% transfere spherical to cartisian coordinates
[x,y,z]=sph2cart(counter0,ElevationAngle,r);
plot3(handles.axes1,x,y,z,'*')
hold on
grid on
end
end
end
end
```

5- Run the program and test the output:



6- Use Task 2 as an exercise to the 3D workspace plotting.

5. Task:

1 – Inside Example 1 environment modify the GUI code so that it could draw the workspace of RP type when it's chosen from the Pop Up menu.

Hints:

- you could use `DOFType_Callback` **function to hide redundant Edit and static texts and modify the static label of some objects when a change in the pop up menu occurs.**

```
function DOFType_Callback(hObject, eventdata, handles)
```

- you could use the same code in the example just add another if statement where PopUpMenu has a value of 2.

1 – Add to the pop up menu a 4^{th} option which is 3D RRP where the 3^{rd} joint is a prismatic with max length given by the user.

Bonus projects: (+1 for each requirement)

- **Make a slider that rotates the 3D representation of the workspace.**
- **Add a text box that received the coordinates of a point and prints on a static text whether this point belongs to the workspace or not.**

6. Evaluation Criteria:

Lab Attendance:

Deliver a working 2D simulation and fail to answer in the Evaluation Oral Exam questions:

Deliver a working 2D simulation and answer to the Evaluation Oral Exam questions:

Deliver a working 2D and 3D simulation and fail to answer in the Evaluation Oral Exam questions:

Deliver a working 2D and 3D simulation and answer to the Evaluation Oral Exam questions:

Notes:

* Max number of team members is 6.

** Only members who attend the Evaluation Exam would be graded.

7. Recommended readings:

- Robotics course material: https://sites.google.com/site/cse4316robots/
- Robot Modeling and Control (1st Edition). Wiley, ISBN: 978-0471649908.
- B. Siciliano and O. Khatib (2008). Springer Handbook of Robotics. Berlin Heidelberg: Springer-Verlag. ISBN: 978-3540239574.