

**برنامه نویسی ربات شرکت ABB مدل IRB140
توسط نرم افزار متلب به وسیله کتابخانه ARTE**

ارائه دهنده: احسان شهنازی

استاد: آقای دکتر خوش بیانی

فروردین ماه ۱۳۹۴
دانشگاه آزاد اسلامی واحد کاشان

در این آموزش از کتابخانه arte برای برنامه نویسی ربات استفاده شده است. این کتابخانه همراه با متلب نصب نمی شود بلکه باید به صورت جداگانه آن را دانلود کرد و سپس به صورت زیر آن را در متلب فراخوانی کرد.
کتابخانه را در مسیر زیر کپی میکنیم.

```
C:\Program Files\MATLAB\R2010a\toolbox
```

و بعد در برنامه متلب مسیر File=>set path را میرویم. در پنجره باز شده کتابخانه arte را از طریق گزینه add With subfolder انتخاب میکنیم.
سپس با استفاده از دستور

```
>>init_lib
```

می توانیم بفهمیم که آیا کتابخانه افزوده شده است یا نه.
در صورتی که کتابخانه با موفقیت به متلب افزوده شده باشد کدهای زیر در ادامه دستور بالا به نمایش در می آیند.

```
% ARTE (A Robotics Toolbox for Education)
```

```
% Copyright (C) 2012 Arturo Gil Aparicio, arturo.gil@umh.es
```

```
% http://arvc.umh.es/arte
```

```
%
```

```
% This program is free software: you can redistribute it and/or modify
```

```
% it under the terms of the GNU Lesser General Public License as published by
```

```
% the Free Software Foundation, either version 3 of the License, or
```

```
% any later version.
```

نحوه برنامه نویسی در متلب با استفاده از کتابخانه ARTE به صورت زیر است: (در این مقاله قرار است با نحوه انجام کارهای زیر آشنا شویم.)

- ۱- با استفاده از رابط گرافیکی (فراخوانی با دستور teach) ربات و نقاط هدف را مشخص میکنیم.
کاربر ربات را در نقاط مختلف از فضای کاری که مورد نیاز خواهد بود، قرار می دهد.
(برای مثال، گرفتن یک قطعه و گذاشتن آن داخل جعبه.)
- ۲- این نقاط در یک ام فایل نوشته می شوند. به منظور انجام این کار، نقاط در رابط گرافیکی ایجاد و سپس آن را به یک ام فایل انتقال می دهند.
- ۳- برنامه نوشت شده می تواند در نرم افزار متلب شبیه سازی شود. با استفاده از ابزارهای اشکال یابی می توان گام به گام برنامه را تحت نظر گرفت و ایرادات آن را برطرف نمود.

۴- در نهایت، این برنامه را می توان به یک فایل RAPID ترجمه کرد. این کار با استفاده از تابع matlab2RAPID قابل انجام است. فایل بدست آمده می تواند برای برنامه ریزی یک ربات واقعی مورد استفاده قرار گیرد.

راه های مختلفی برای انتقال این برنامه ترجمه شده به ربات وجود دارد که میتوان برای نمونه به برنامه Robot studio اشاره کرد و یا استفاده از برنامه هایی که از پروتکل FTP برای انتقال اطلاعات استفاده می کنند.

ابتدا با استفاده از دستورات زیر ربات abb- irb140 را فراخوانی میکنیم.

```
>> init_lib  
>> robot=load_robot('abb', 'IRB140');
```

سپس با استفاده از دستور teach که در اسلاید قبلی در مورد آن بحث شد محیط گرافیکی مورد نظر را لود می کنیم.

```
>> teach
```

علاوه بر ربات abb می توانیم ربات های دیگری را نیز فراخوانی کنیم.
لیست ربات هایی که در کتابخانه **arte** هستند عبارتند از:

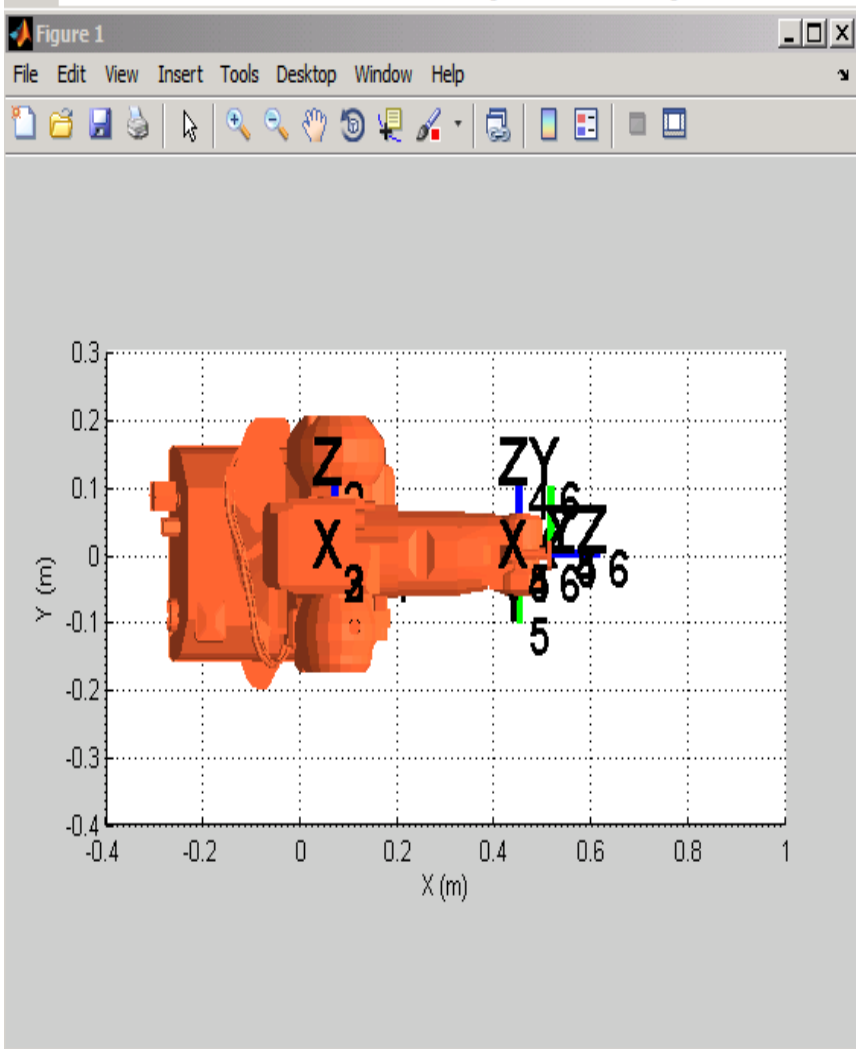
ABB
ADEPT
EPSON
MITSUBISHI
FANUC
KUKA
STÄUBLI
UNIMATE

این ربات ها از آدرس **arte => robots** قابل دسترس هستند.

بعد از اجرای دستور teach رابط گرافیکی به صورت شکل زیر باز می شود.

linear_velmax: 2.5000

T0: [4x4 double]



teach

Move joints

Current value (deg, m)

q_1 0

q_2 0

q_3 0

q_4 0

q_5 0

q_6 0

Reset all

Move end effector

Y+ Z+ X+ Z- Y- X-

Kind of movement Line

Coordinates Base

Resolution High

Direct/Inverse kinematics

T=

-0.000	-0.000	1.000	0.515
-0.000	1.000	0.000	0.000
-1.000	-0.000	-0.000	0.712

Move to

Q=

0.707	-0.000	0.707	0.000
-------	--------	-------	-------

Move to

(Px, Py, Pz)

0.515	0.000	0.712
-------	-------	-------

Target points

Save current targ... RT_tp1

Show current tar...

Save TPs to file

Refresh

Load Robot Load end tool Load equip... Load a piece

>> teach

جهت مشخص کردن نقاط حرکت ربات از قسمت 'Target points' نقاط هدف را مشخص میکنیم.

The screenshot shows the 'teach' software interface with the following sections:

- Move joints:** Six sliders for joints q_1 to q_6, each with a 'Current value (deg, m)' of 0. A 'Reset all' button is below.
- Move end effector:** A directional pad with X-, X+, Y+, Y-, Z+, Z- buttons. Below are dropdowns for 'Kind of movement' (Line), 'Coordinates' (Base), and 'Resolution' (High).
- Direct/inverse kinematics:** Two tables for transformation matrices T and Q, and a position vector (Px, Py, Pz). Each has a 'Move to' button.
- Target points:** A section with buttons 'Save current targ...', 'Show current tar...', and 'Save TPs to file'. It also shows a list of target points with 'RT_tp1' selected and a 'Target' button highlighted with a red circle.

At the bottom are buttons: 'Load Robot', 'Load end tool', 'Load equip...', 'Load a piece', and 'Refresh'.

-0.000	-0.000	1.000	0.515
-0.000	1.000	0.000	0.000
-1.000	-0.000	-0.000	0.712

0.707	-0.000	0.707	0.000
-------	--------	-------	-------

0.515	0.000	0.712
-------	-------	-------

توضیح قسمت های مختلف پنجره teach (رابط گرافیکی کنترل بازوهای ربات)

Load Robot

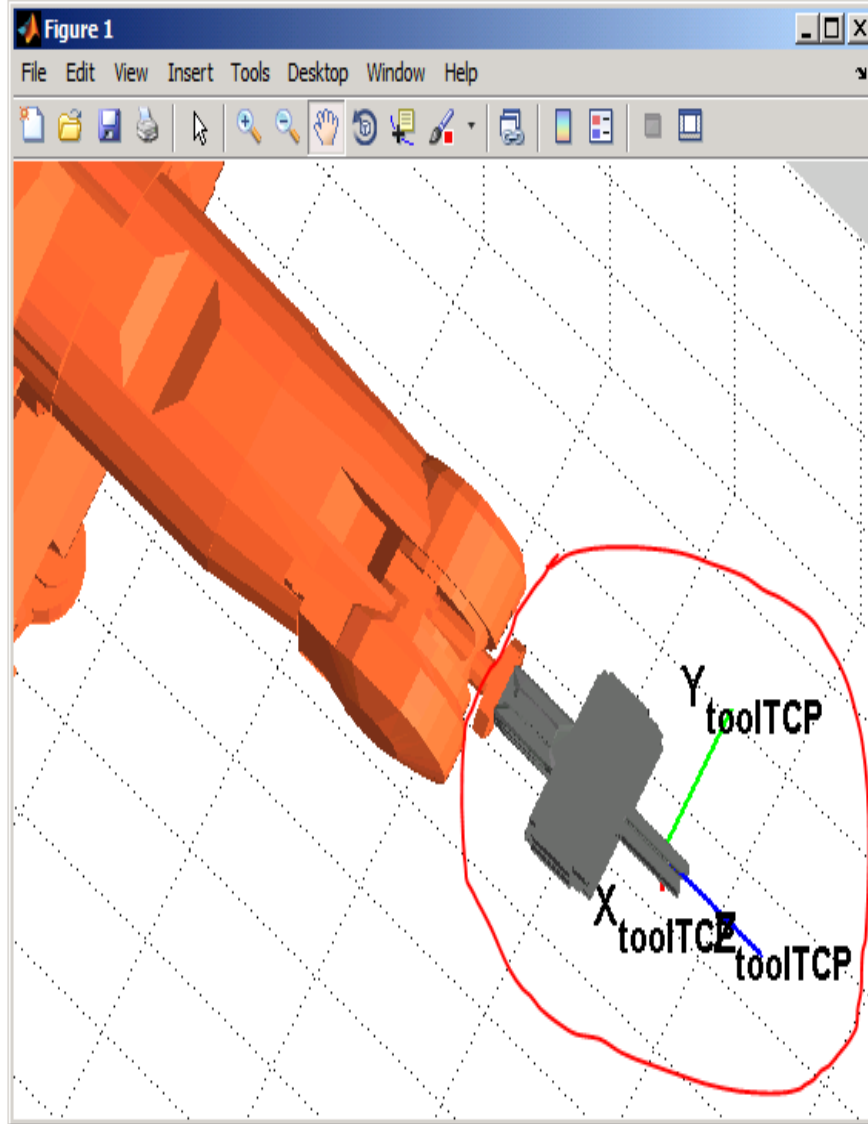
برای فراخوانی ربات مورد نظر به محیط شبیه ساز استفاده می شود. در اینجا ربات مورد نظر ما abb-irb140 می باشد. که در اینجا ما با استفاده از دستور load_robot این کار را کردیم.

load end tool (effector)

برای انتخاب کارگیر مناسب از این گزینه استفاده می کنیم. کارگیرهای مختلفی در کتابخانه arte وجود دارند که می توانید از طریق آدرس arte/robots/equipment/end_tools به آنها دسترسی پیدا کنید.

برای انتخاب کارگیر مورد نظر شما باید فایل parameters.m را فراخوانی نمایید. بعد از فراخوانی کارگیر روی محیط ۳ بعدی نمایش داده می شود. (البته بدون ربات اصلی abb-ir140- برای مشاهده کارگیر روی ربات باید از پنجره teach گزینه refresh را بزنید.

تصویر موجود در اسلاید بعدی اتصال کارگیر به ربات را نشان داده است.



teach

Move joints

Current value (deg, m)

q_1 0

q_2 0

q_3 0

q_4 0

q_5 0

q_6 0

Reset all

Move end effector

Y+ X- Z+ X+ Z- Y-

Kind of movement Line

Coordinates Base

Resolution High

Target points

Save current targ... RT_tp1

Show current tar... Target

Save TPs to file

Refresh

Direct/inverse kinematics

T=

-0.000	-0.000	1.000	0.640
-0.000	1.000	0.000	0.000
-1.000	-0.000	-0.000	0.712

Move to

Q=

0.707	-0.000	0.707	0.000
-------	--------	-------	-------

Move to

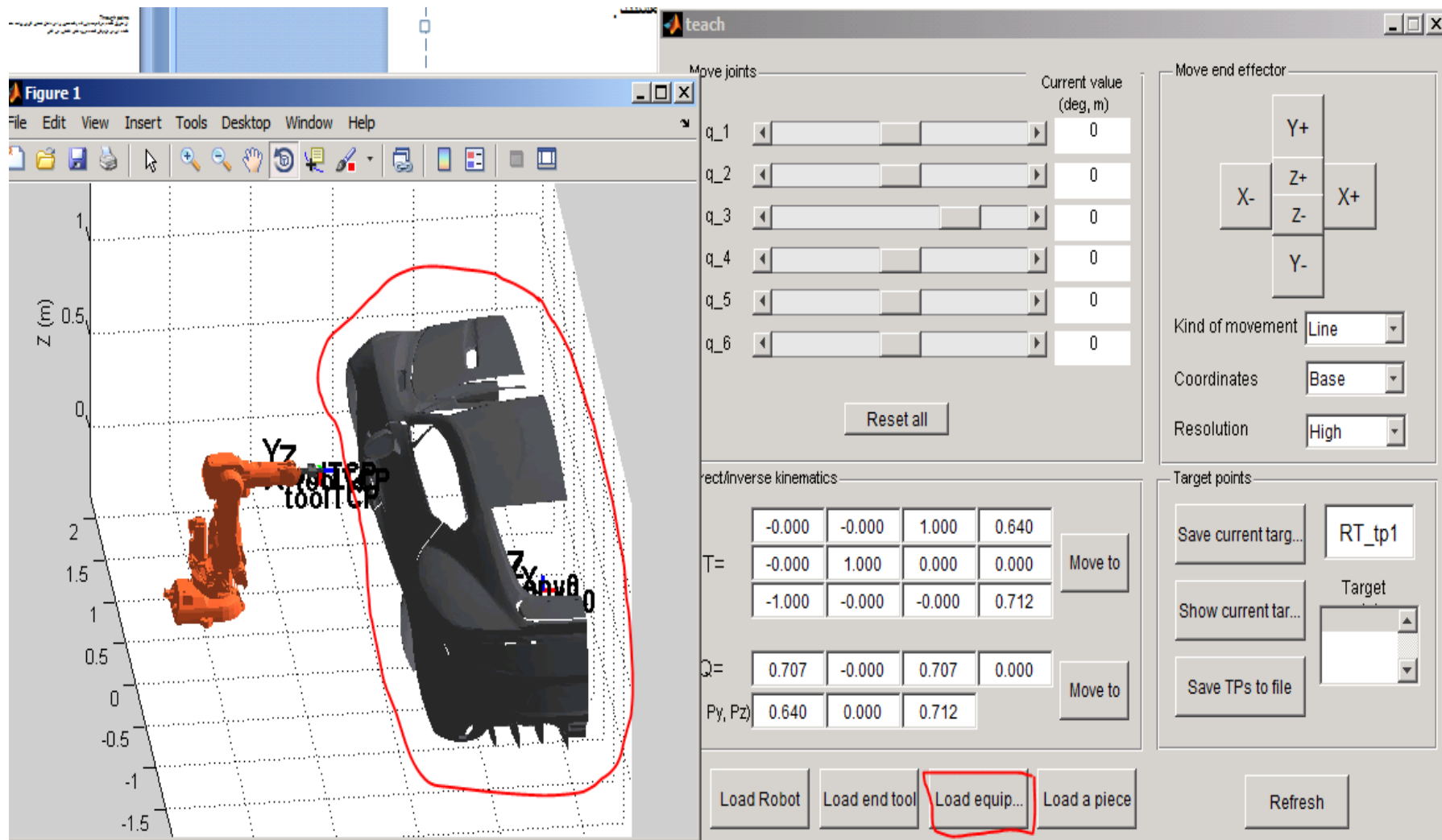
Py, Pz)

0.640	0.000	0.712
-------	-------	-------

Load Robot Load end tool Load equip... Load a piece

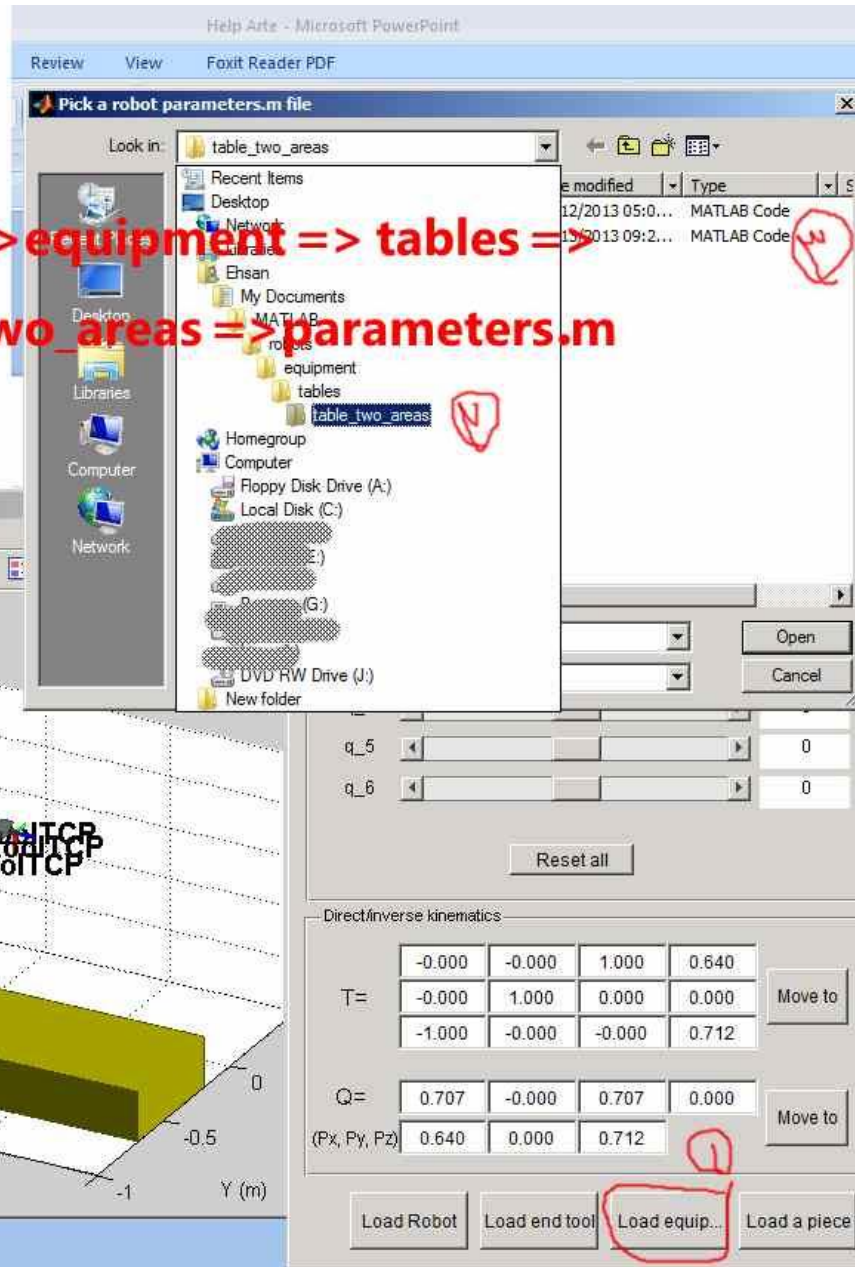
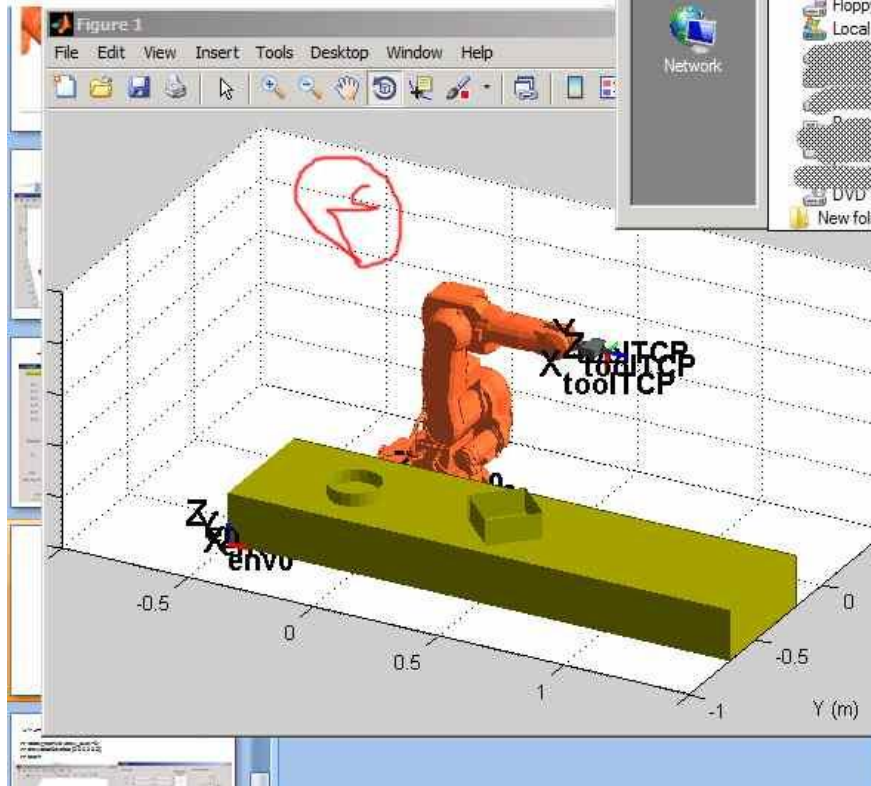
Load equipment

این گزینه برای آوردن تجهیزات مکمل و کمکی به محیط کار ربات است.
این تجهیزات از مسیر `arte/robots/equipment/bodywork/` قابل دسترس هستند.

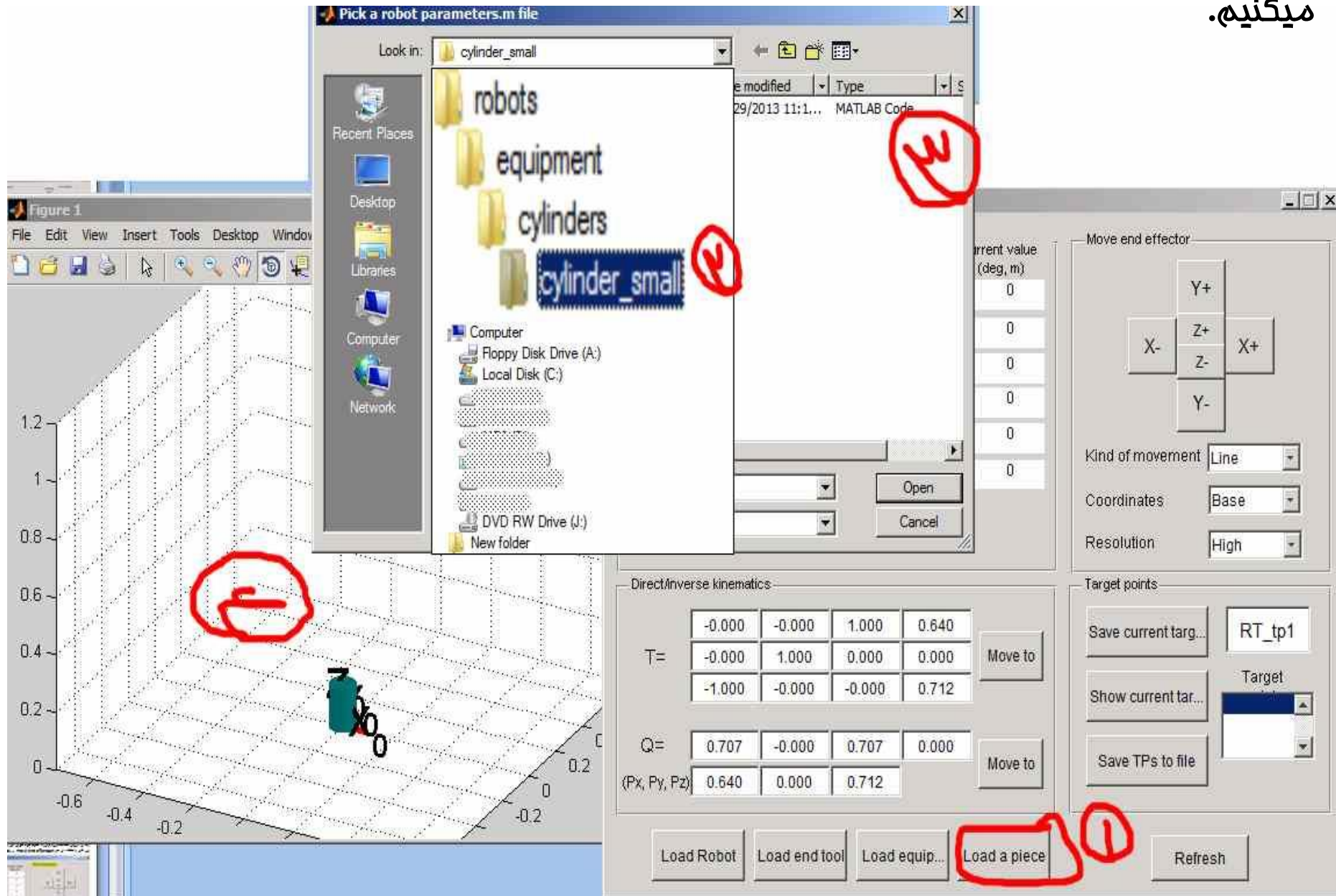


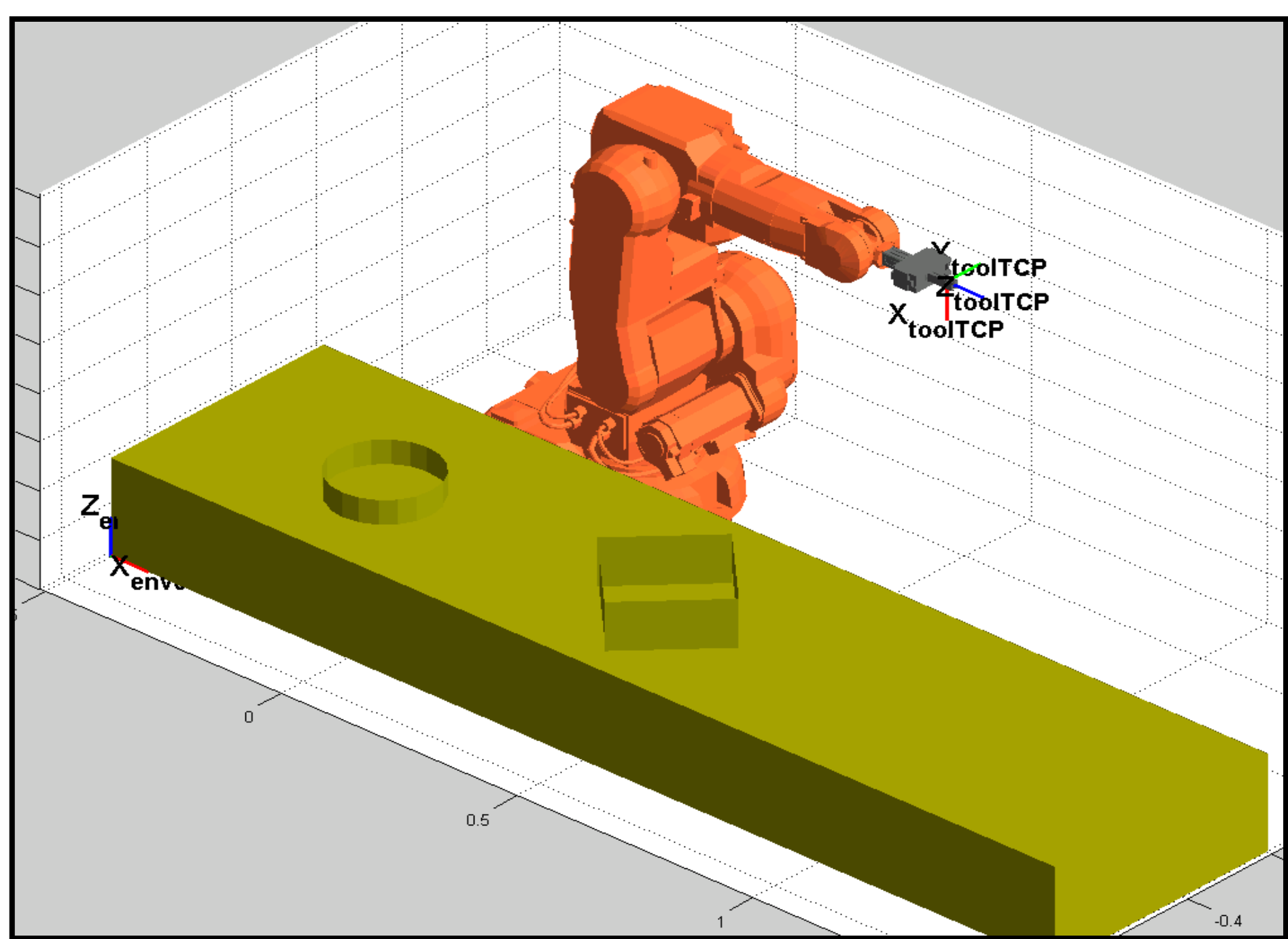
نمونه آوردن یک میز به محیط شبیه ساز در شکل زیر نمایش داده شده است.

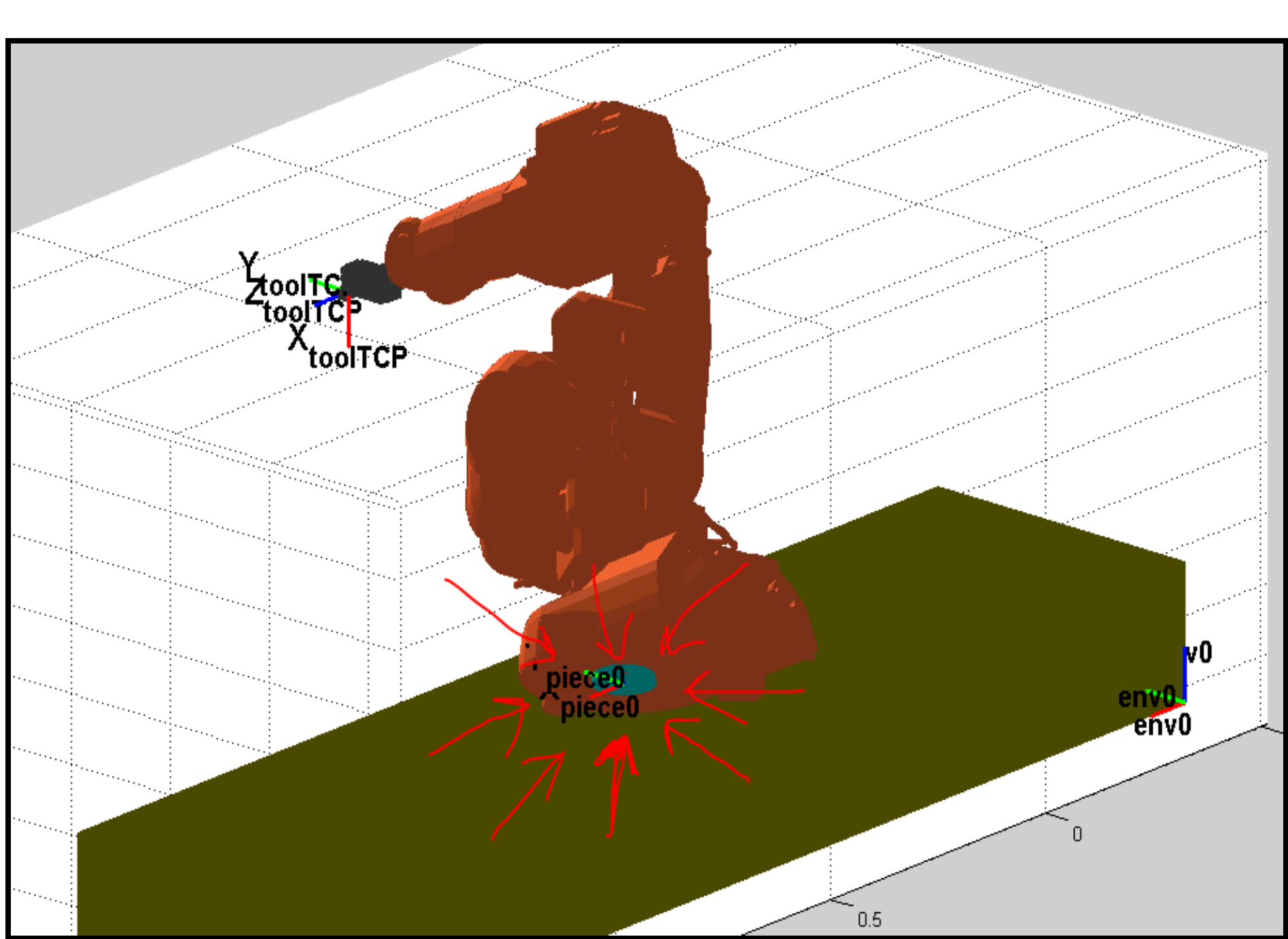
robot => equipment => tables =>
table_two_areas => parameters.m



برای آوردن یک قطعه کوچک که ربات بتواند آن را جابجا کند از گزینه load a piece استفاده میکنیم.





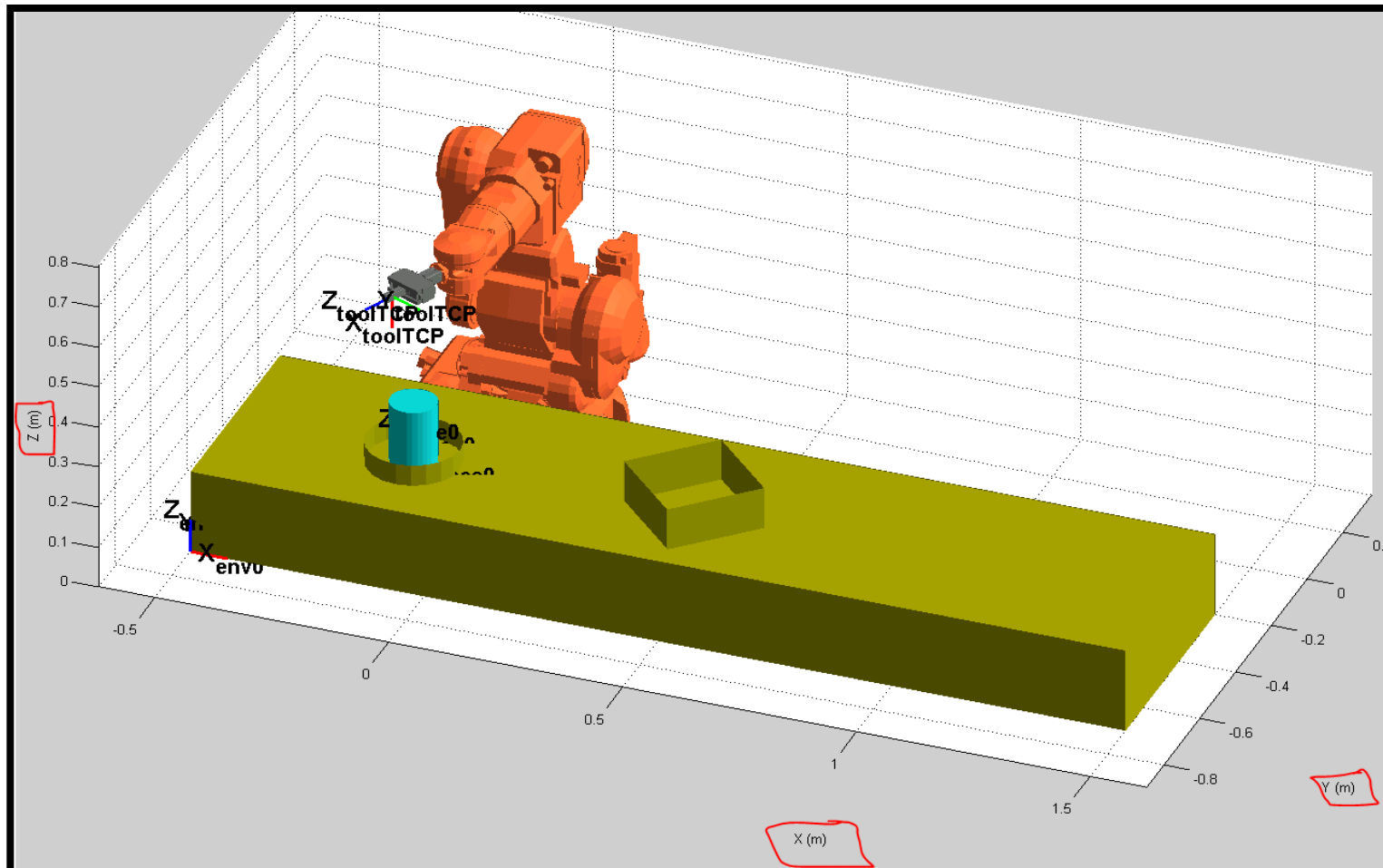


برای تغییر مختصات قطعه کار از دستور زیر استفاده میکنیم.

```
>> robot.piece.T0(1:3,4)=[-0.1 -0.5 0.2]';
```

Tool Center Point (TCP)

مختصاتی است که کارگیر باید در آن نقطه قطعه مورد نظر را بردارد.
برای مثال در یک ابزار نقطه جوش زن، TCP موقعیت و جهت نوک ابزار است که در آنجا جوش می زند.



بعد از آوردن قطعات مورد نیاز به محیط شبیه سازی می بایست کارگیر ربات را به نقطه مورد نظر ببریم.

این کار با استفاده از منوهای move end effector و یا move joints انجام پذیرد.

The screenshot displays the 'teach' software interface for robot control, organized into several functional panels.

Move joints: This panel allows for individual joint movement. It features six horizontal sliders labeled q_1 through q_6, each with a 'Current value (deg, m)' field to its right, all currently set to 0. A 'Reset all' button is located at the bottom of this section.

Move end effector: This panel provides a visual representation of the end effector's position and movement parameters. It includes a 3D coordinate system with X-, X+, Y-, Y+, Z-, and Z+ directional buttons. Below this, three dropdown menus are present: 'Kind of movement' set to 'Line', 'Coordinates' set to 'Base', and 'Resolution' set to 'High'.

Direct/inverse kinematics: This panel is used for calculating and moving to specific target poses. It contains two sets of input fields: one for transformation matrix 'T' and another for joint angles 'Q'. Each set has a 'Move to' button. The 'T' matrix values are: [-0.000, -0.000, 1.000, 0.640; -0.000, 1.000, 0.000, 0.000; -1.000, -0.000, -0.000, 0.712]. The 'Q' values are: [0.707, -0.000, 0.707, 0.000]. Below these, a row for (Px, Py, Pz) shows values [0.640, 0.000, 0.712].

Target points: This panel manages saved target points. It includes buttons for 'Save current targ...', 'Show current tar...', and 'Save TPs to file'. A list box labeled 'Target' currently displays 'RT_tp1'.

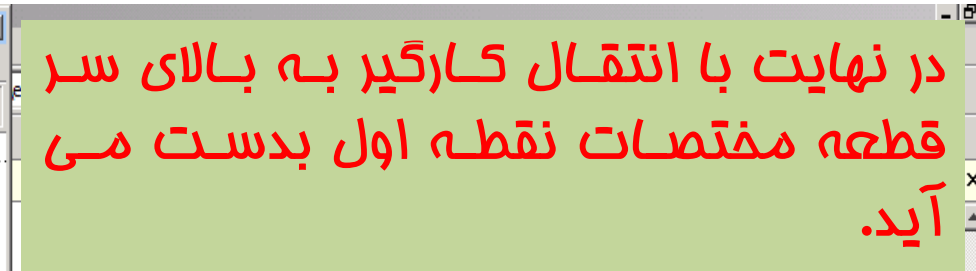
Bottom Controls: A row of four buttons at the bottom includes 'Load Robot', 'Load end tool', 'Load equip...', and 'Load a piece'. A 'Refresh' button is positioned to the right of these.

در منوی move end effector اگر گزینه kind of movment را liner انتخاب کنیم با کلیک روی گزینه های $x+, y+, z+, \dots$ تغییر جهت کارگیر به صورت قطعی خواهد بود.

اما اگر آن را movement انتخاب کنیم حرکات به صورت چرخشی خواهد بود.

علاوه بر گزینه هایی که در بالا شرح داده شد میتوان از قسمت direct / inverce kinematic هم استفاده نمود. بدین صورت که مختصات جایی که کارگیر باید برود را در قسمت px, py, pz می نویسیم و سپس دکمه move to را میزنیم. میبینیم که بازوهای ربات به صورت خودکار به سمت آن مختصات حرکت می کند و در نقطه مورد نظر می ایستند.

دقت شود چنانچه مختصات درج شده صحیح نباشند ربات هیچگونه حرکتی نخواهد کرد.



The screenshot displays the 'teach' interface with the following components:

- Joint Positions:** A table showing the current values for joints q_1 through q_6 in degrees (deg) and meters (m).

Joint	Current value (deg, m)
q_1	-102
q_2	15.6
q_3	9.09
q_4	-178
q_5	-67.3
q_6	-7.2972
- Kinematics:**
 - Direct/Inverse kinematics:** A section with input fields for q_1 and q_2, and a 'Reset all' button.
 - T= (Transformation Matrix):**

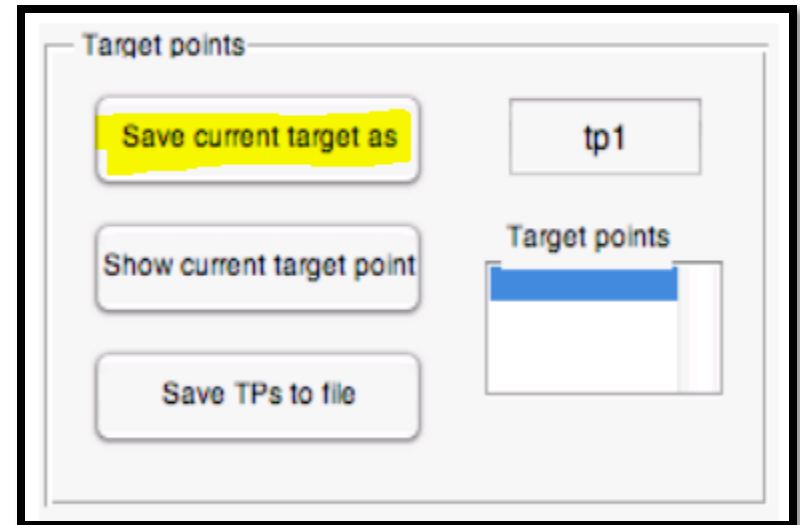
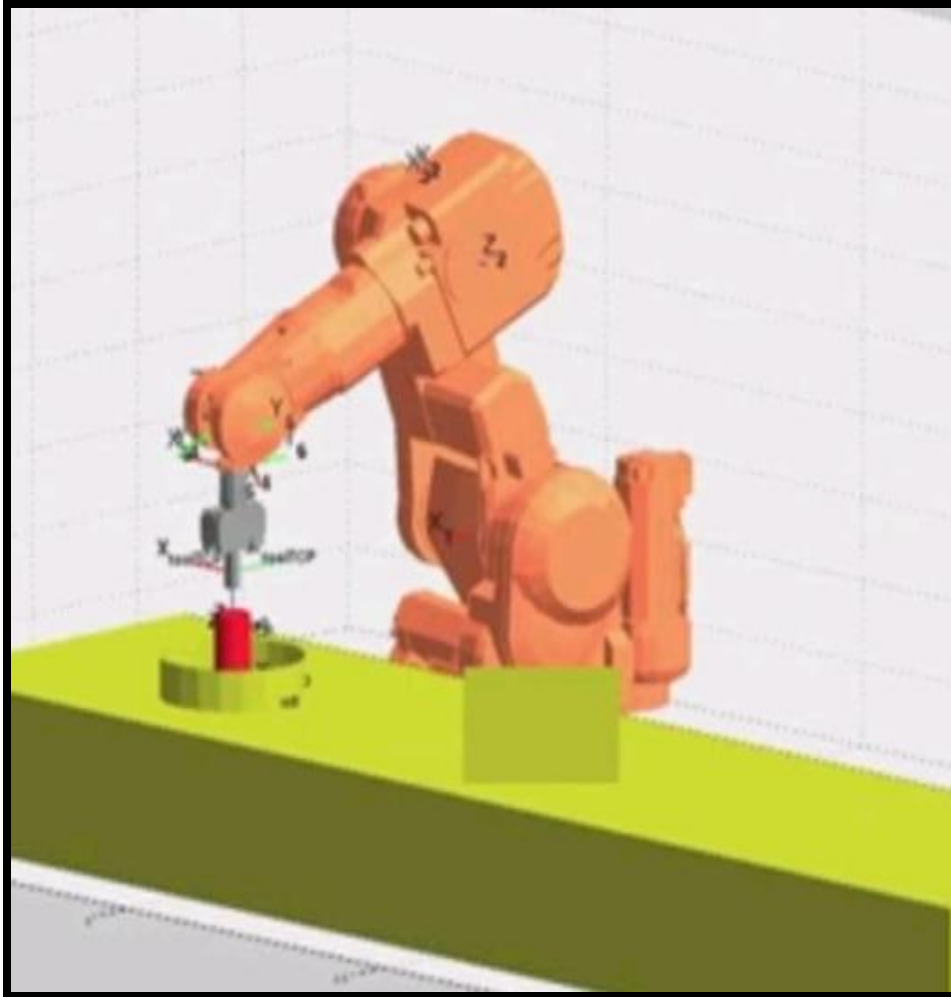
-0.090	-0.995	0.043	-0.095
-0.996	0.091	0.026	-0.497
-0.030	-0.040	-0.999	0.350
 - Q= (Joint Positions):**

0.025	-0.674	0.738	-0.005
-------	--------	-------	--------
 - (Px, Py, Pz) (End Effector Position):**

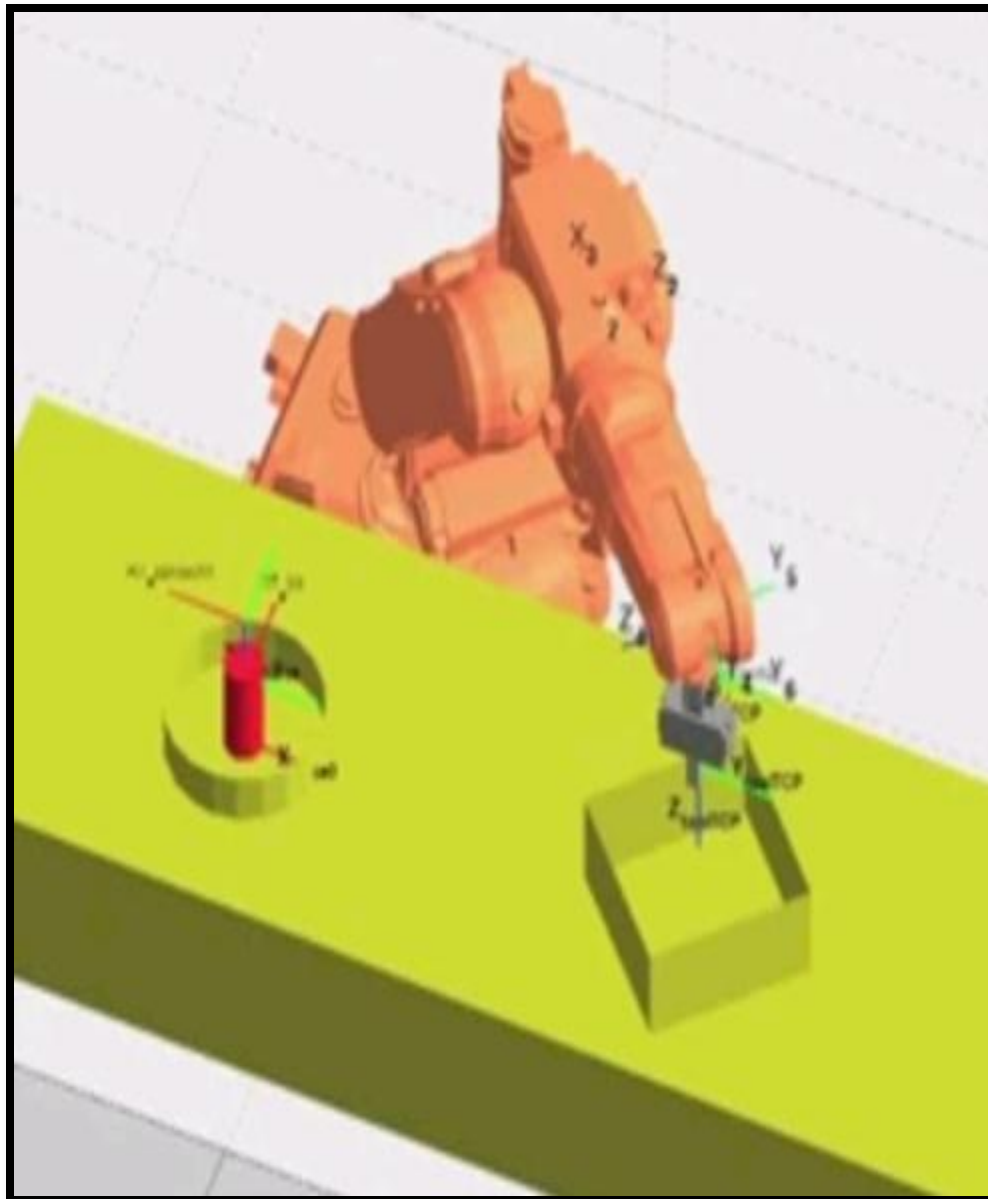
-0.095	-0.497	0.350
--------	--------	-------
- Movement Controls:**
 - Move joints:** A section with a 'Reset all' button and a 'Move to' button.
 - Kind of movement:** A dropdown menu set to 'Reorient'.
 - Coordinates:** A section with a 'Reset all' button and a 'Move to' button.
 - Resolution:** A dropdown menu set to 'Low'.
 - Target points:** A section with a 'Reset all' button and a 'Move to' button.
 - Save current target:** A button labeled 'RT_tp1'.
 - Show current target:** A button.
 - Save TP's to file:** A button.
 - Refresh:** A button.

بعد از اینکه کارگیر را به نقطه مورد نظر رساندیم روی گزینه “Save current target as” کلیک میکنیم.

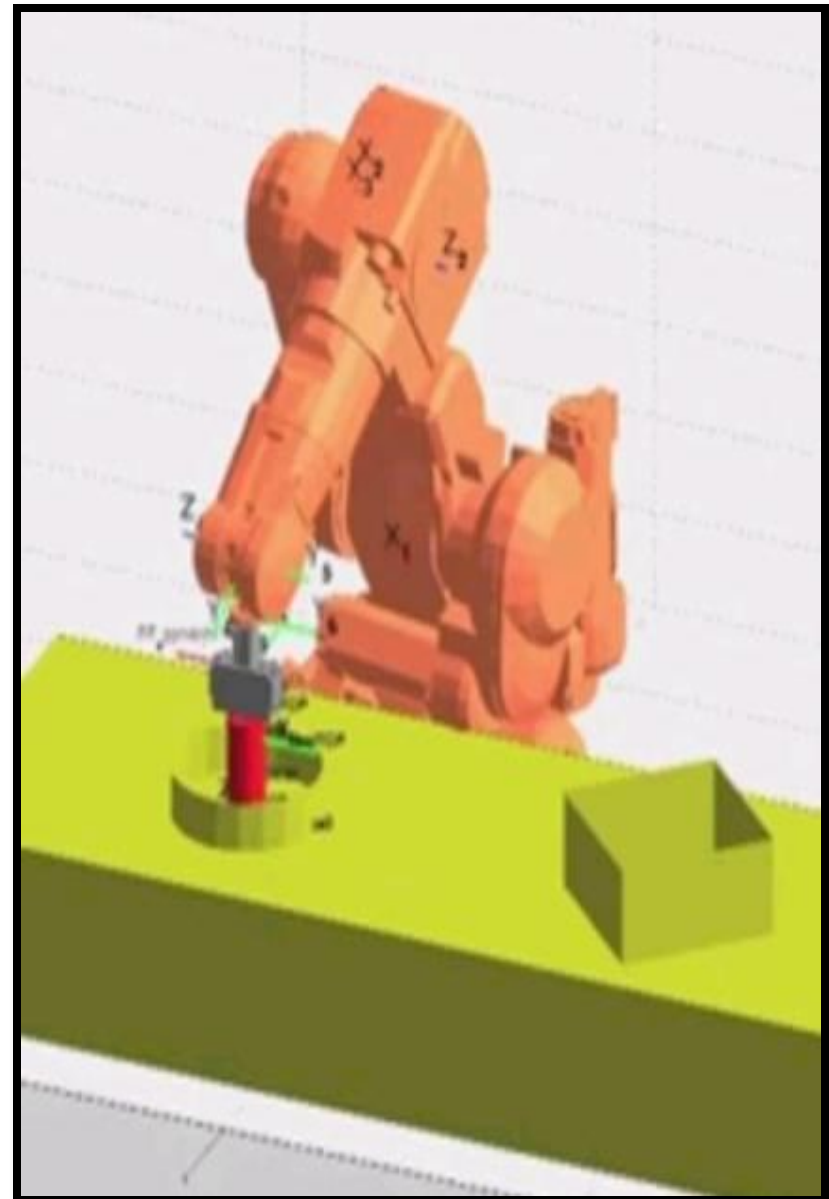
برای هر نقطه می بایست این کار را انجام دهیم و مختصات آن را ذخیره کنیم.
مختصات نقطه اول را در موقعیت زیر در نظر میگیریم.



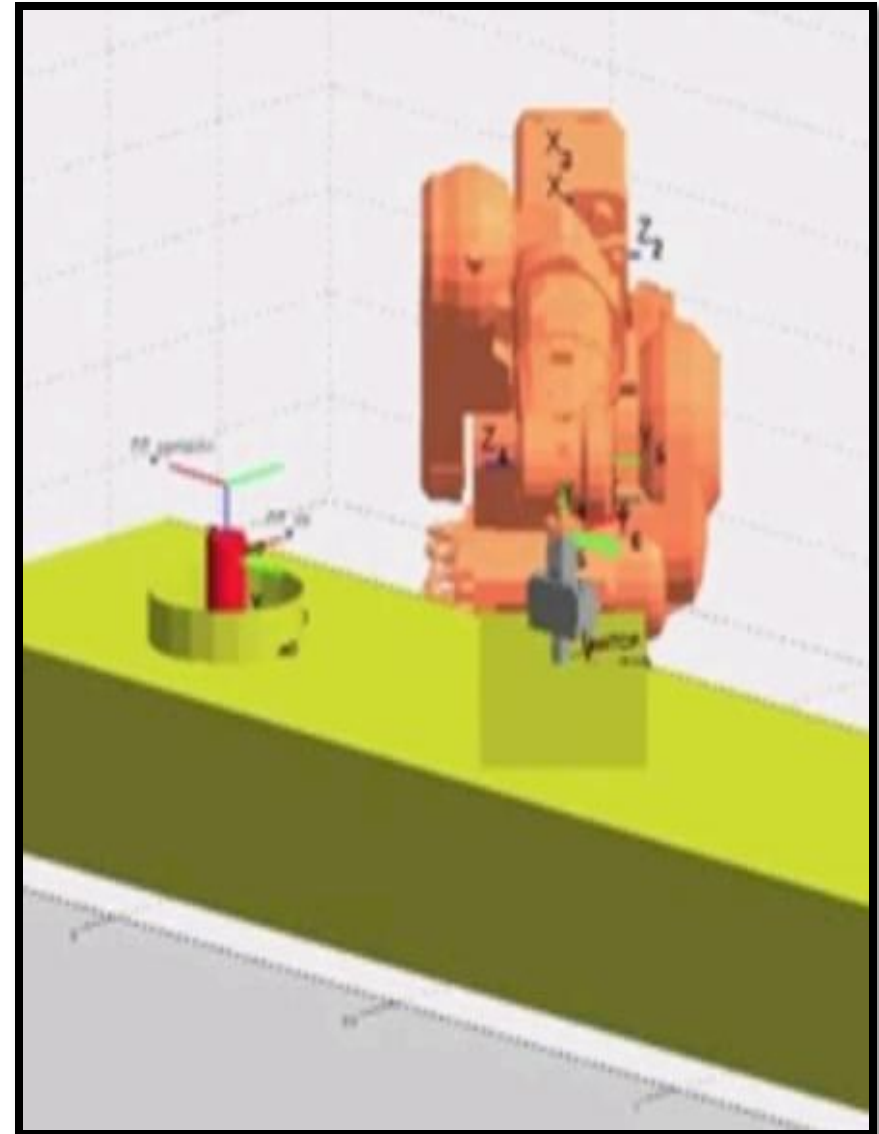
مختصات نقطه سوم



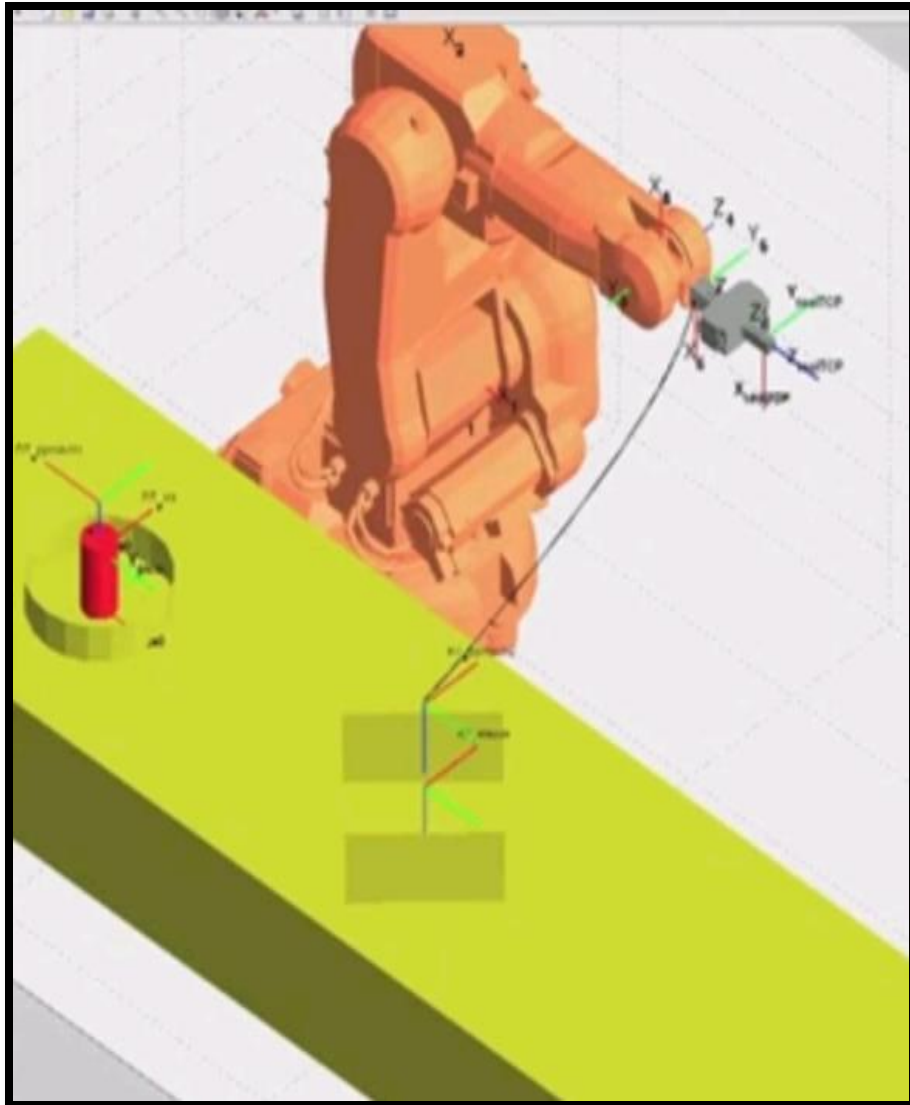
مختصات نقطه دوم



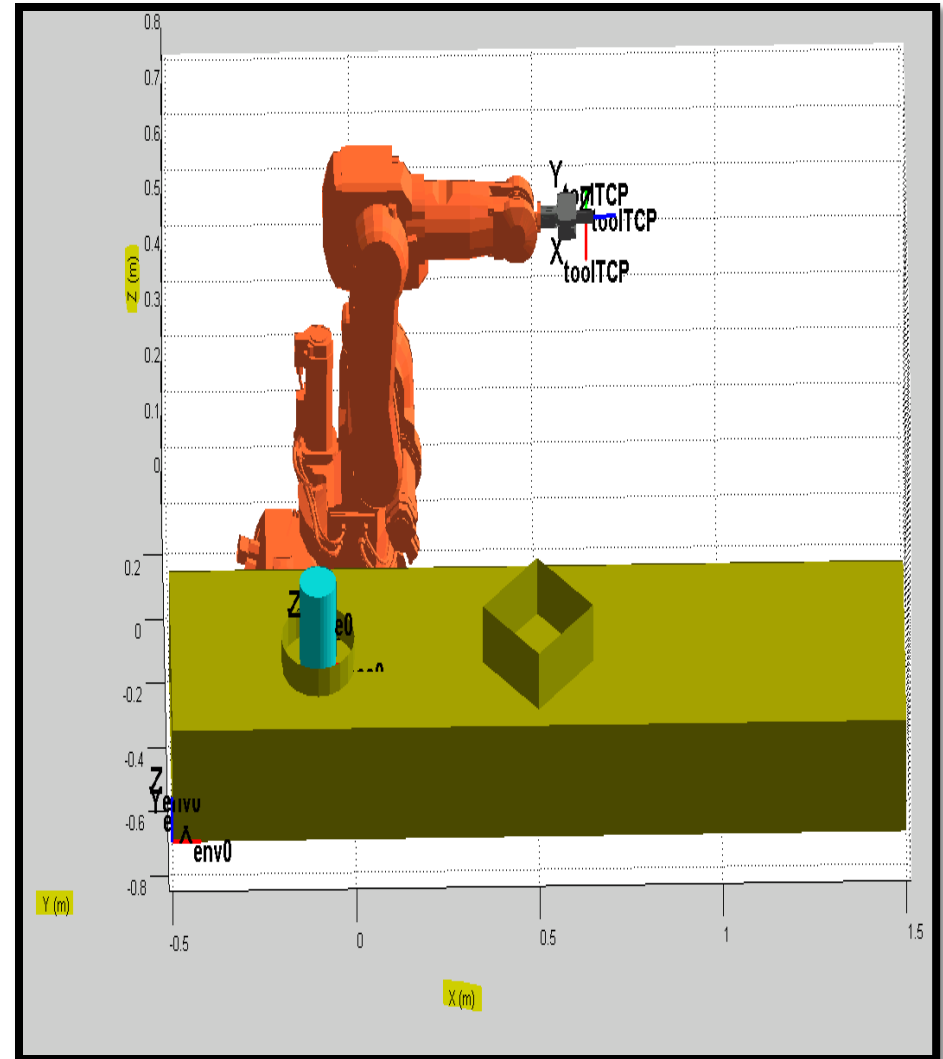
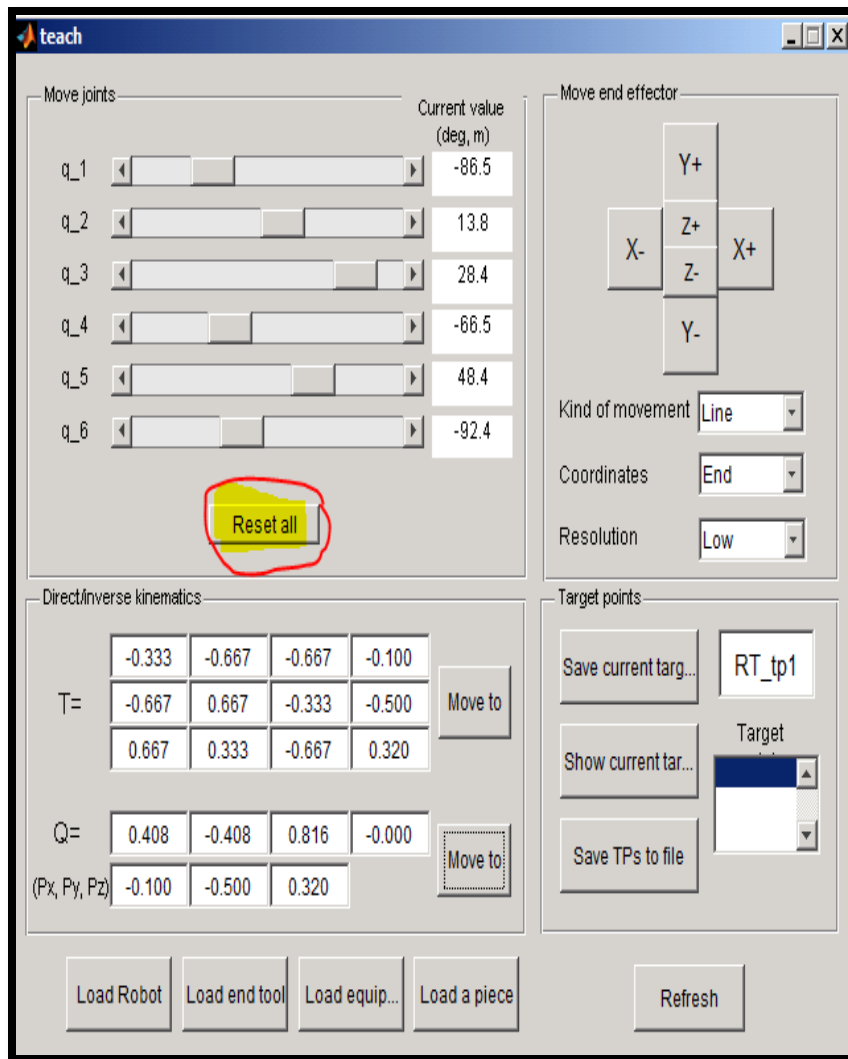
مختصات نقطه چهارم



مختصات نقطه پنجم

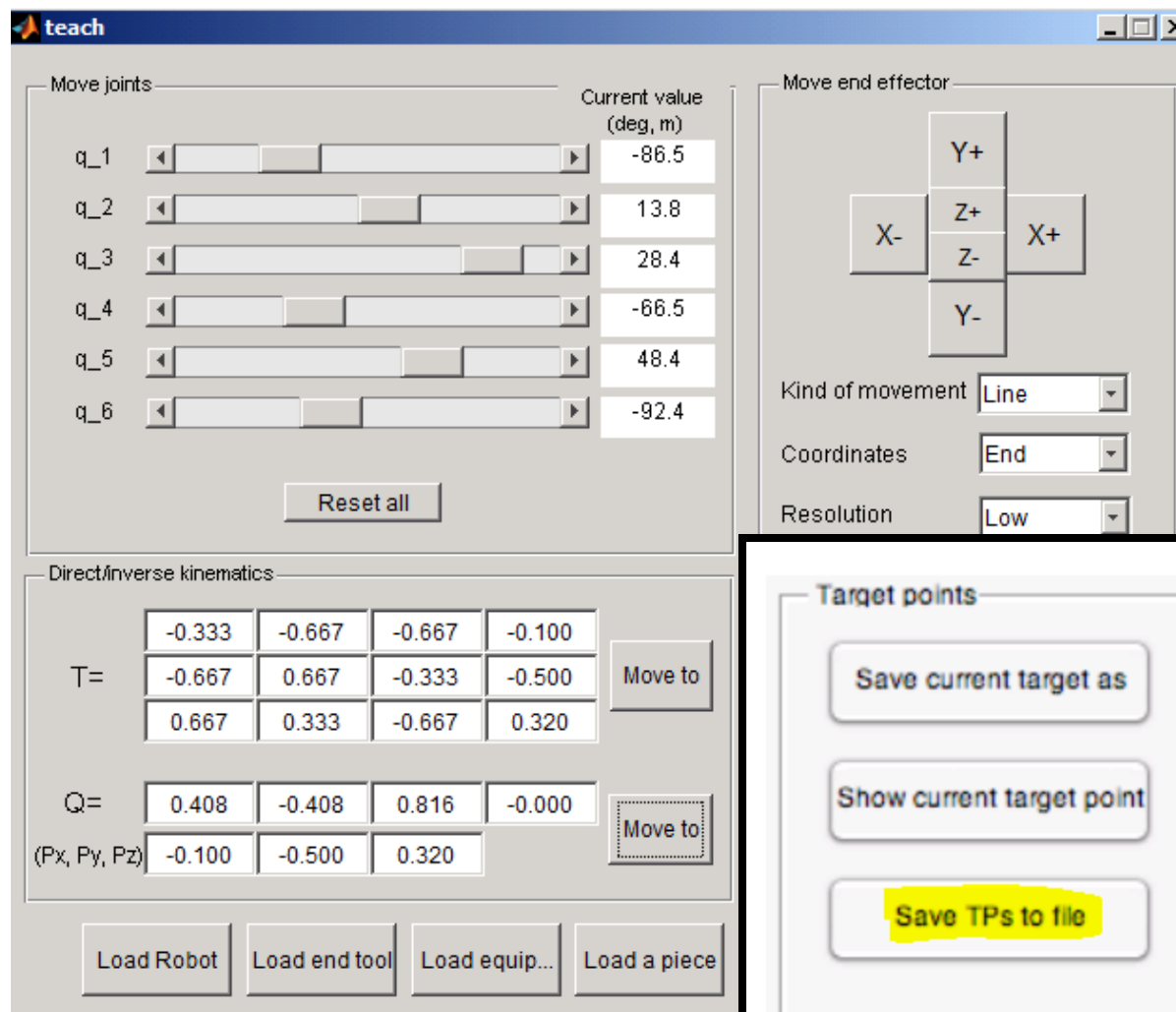


مختصات نقطه پنجم همان نقطه شروع است که از آنجا حرکت ربات شروع می شود. برای رسیدن به این نقطه می توانیم دکمه reset all را بزنیم.
با کلیک روی این گزینه بازوهای ربات در هر کجا که باشند به حالت اولیه خود باز می گردند.



در نهایت بعد از اینکه مختصات نقاط مختلف بدست آمد روی گزینه **save tps to file** کلیک می کنیم.

در صفحه جدید پوشه ای را جهت ذخیره کردن این مختصات انتخاب می کنیم. دقت شود که فرمت فایل ذخیره شده باید به صورت **m** باشد. یعنی یک **ام** فایل ایجاد شود.



بعد از ذخیره سازی می توانیم آن فایل را باز کنیم و مختصات درج شده در آن را ببینیم. مختصات ذخیره شده همان نقاطی هستند که در قسمت target point آنها را ذخیره کردیم.

```
1  
2- RT_approach1=[[-0.1000, -0.5000, 0.3500],[0.0000, -0.0000, 1.0000, 0.0000], [-2  
3 RT_pick=[[-0.1000, -0.5000, 0.2800],[0.0000, 0.7071, 0.7071, 0.0000], [-2, -1,  
4- RT_approach2=[[0.5000, -0.5000, 0.4000],[0.0000, 0.7071, 0.7071, -0.0000], [-1,  
5 RT_release=[[0.5000, -0.5000, 0.2800],[0.0000, 0.7071, 0.7071, -0.0000], [-1, -  
6- RT_initial=[[0.6400, 0.0000, 0.7120],[0.7071, -0.0000, 0.7071, -0.0000], [0, -1,
```

در مرحله بعد این مختصات را در فایل practice_1_programming.m واقع در مسیر
arte\RAPID\programs\

برای فراخوانی فایل بالا کافیسست در متلب قطعه کد زیر را تایپ کنیم.

```
>> edit practice_1_programming.m
```

در تصاویر موجود در اسلاید بعدی مکان کپی کدها مشخص شده است.

```

22 % To grip the piece, call simulation_grip_piece
23 % simulation_release_piece to release it.
24 % The call to each function must be correct, th
25 % sequence is:
26
27 % simulation_open_tool;
28 % approach the piece to grab.
29 % simulation_close_tool;
30 % simulation_grip_piece; --> the piece will be
31 % move to a different place
32 % simulation_open_tool;
33 % simulation_release_piece
34
35 function practice_1_programming
36 global robot
37
38 %init the position of the piece at the beginnin
39 robot.piece.T0(1:3,4)=[-0.1 -0.5 0.2]';
40 %robot.tool.piece_gripped=0;
41 drawrobot3d(robot, robot.q);
42
43 %close the tool
44 simulation_close_tool;
45
46 %define the tool
47 %In RAPID this is done by means of the tooldata
48 TD_gripper=[1,[0,0,0.125],[1,0,0,0]],[0.1,[0,0
49
50 %define target points
51 RT_initial=[[0.5,-0.4,0.5],[0.427269,0.289111,0
52 RT_approach1=[[-0.1, -0.5, 0.4],[0.0, 0.70711,
53 RT_grip=[[-0.1, -0.5, 0.28],[0.00000, 0.70711,

```

code Here

```

35 function practice_1_programming
36 global robot
37
38 %init the position of the piece at the beginnin
39 robot.piece.T0(1:3,4)=[-0.1 -0.5 0.2]';
40 %robot.tool.piece_gripped=0;
41 drawrobot3d(robot, robot.q);
42
43 %close the tool
44 simulation_close_tool;
45
46
47 RT_approach1=[[-0.1000, -0.5000, 0.3500],[0.000
48 RT_pick=[[-0.1000, -0.5000, 0.2800],[0.0000, 0.
49 RT_approach2=[[0.5000, -0.5000, 0.4000],[0.0000
50 RT_release=[[0.5000, -0.5000, 0.2800],[0.0000,
51 RT_initial=[[0.6400, 0.0000, 0.7120],[0.7071, -
52
53 %define the tool
54 %In RAPID this is done by means of the tooldata
55 TD_gripper=[1,[0,0,0.125],[1,0,0,0]],[0.1,[0,0
56
57 %define target points
58 RT_initial=[[0.5,-0.4,0.5],[0.427269,0.289111,0
59 RT_approach1=[[-0.1, -0.5, 0.4],[0.0, 0.70711,
60 RT_grip=[[-0.1, -0.5, 0.28],[0.00000, 0.70711,
61
62 RT_approach2=[[0.5, -0.5, 0.4],[0.00000, 0.7071
63 RT_release=[[0.5, -0.5, 0.28],[0.00000, 0.70711
64
65 %move to the initial point
66 MoveJ(RT_initial, 'vmax', 'fine', TD_gripper,

```

غیر از راه بالا که مختصات نقاط هدف را در یک فایل از پیش تعریف شده کپی می‌کردیم می‌توانیم خودمان آن فایل را ایجاد کنیم.

قطعه کد که در اسلاید بعدی آورده شده است را باید در یک `function` نوشت. برای ایجاد `function` باید مسیر `file\new\ function` را برویم.

کدهایی که با رنگ زرد های لایت شده اند همان مختصات نقاط کارگیر هستند که در مرحله قبل آنها را ذخیره کردیم.

در واقع به غیر از کدهای های لایت شده بقیه کدها همواره ثابت هستند و آن‌ها را تغییری نمی‌دهیم.

```

function test_1
global RT_tp1 RT_tp2 TD_tool0

RT_tp1=[[0.4000, 0.3000, 0.7000],[1.0000, -0.0000, -0.0000, -0.0000], [0, -1,
-1, 0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
RT_tp2=[[0.1000, -0.4000, 0.7000],[0.0000, -0.0000, 0.0000, 1.0000], [-1, -1,
-2, 0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

TD_tool0=[1,[[0,0,0],[1,0,0,0]],[0,[0,0,0],[1,0,0,0],0,0,0]];

main;
end

function main()
global RT_tp1 RT_tp2 TD_tool0

    MoveJ(RT_tp1, 'vmax' , 'fine' , TD_tool0, 'wobj0');
    MoveJ(RT_tp2, 'vmax' , 'fine' , TD_tool0, 'wobj0');
    MoveL(RT_tp1, 'vmax' , 'fine' , TD_tool0, 'wobj0');
    MoveL(RT_tp2, 'vmax' , 'fine' , TD_tool0, 'wobj0');
end

```

بعد از نوشتن کدها فایل را ذخیره می کنیم. (نام انتخابی برای این فایل را test_1.m می گذاریم).

حال برای اینکه کدهای بدست آمده به برنامه ای قابل خوانش توسط ربات (در اینجا ربات های شرکت abb) تبدیل شود می بایست با استفاده از دستور زیر کدها را به زبان rapid تبدیل کرد.

>> matlab2RAPID

با تایپ این کد در محیط متلب پنجره ای باز می شود که می بایست فایل test_1.m را انتخاب کرد.

بعد از انتخاب فایل متون زیر در صفحه متلب نمایش داده می شوند.

Converting Matlab into RAPID lenguaje...

Sink_name =

test_1.prg

فایل test_1.prg بدست آمده به زبان rapid می باشد و می توانیم به وسیله نرم افزار های FTP آن را به کنترلر ربات انتقال دهیم.

توجه: فایل test_1.prg در همان مسیر و پوشه ای ذخیره می شود که فایل test_1.m را انتخاب کردیم.

علاوه بر اینکه با استفاده از محیط گرافیکی می توان ابزارها و کارگیر و... را به محیط شبیه سازی آورد با استفاده از دستورات زیر هم میتوان اینکارها را انجام داد.

```
% A) بارگذاری یک ربات
% robot = load_robot('abb','irb140');
% OR
% robot = load_robot('abb','irb52');
% B) بارگذاری تجهیزات
% robot.equipment = load_robot('equipment','tables/table_two_areas');
%
% C) بارگذاری کارگیر
% robot.tool= load_robot('equipment','end_tools/parallel_gripper_0');
% D) بارگذاری قطعه کار
% robot.piece=load_robot('equipment','cylinders/cylinder_tiny');
%
% اگر لازم بود تغییر موقعیت قطعه کار
%
% robot.piece.T0= [1 0 0 -0.1;
%                  0 1 0 -0.5;
%                  0 0 1 0.2;
%                  0 0 0 1];
```

در مین اجرای شبیه سازی می توان از دستورات زیر استفاده کرد.

```
% during the simulation, call simulation_open_tool; to open the tool and  
% simulation_close_tool; to close it.  
%  
% To grip the piece, call simulation_grip_piece; and  
% simulation_release_piece to release it.  
%  
% The call to each function must be in a correct, otherwise, the result may be  
% undesired, thus, typically the correct sequence is:  
% simulation_open_tool;  
% (approach the piece to grab with MoveL or MoveJ)  
% simulation_close_tool;  
% simulation_grip_piece; --> the piece will be drawn with the robot  
%  
% move to a different place with MoveJ or MoveL  
% simulation_open_tool;  
% simulation_release_piece  
%  
% In RAPID, consider the use of ConfJ on, or ConfJ off, or ConfL on, or ConfL off
```

به طور کلی اگر شما بخواهید تمامی این مراحل گفته شده برای شبیه سازی حرکت ربات جهت برداشتن یک قطعه از یک محل و گذاشتن آن در محلی دیگر را شبیه سازی کنید کارهای زیر را انجام دهید.

۱- فایل زیر را باز میکنیم

arte/RAPID/programs/practice_1_programming.m.

۲- با استفاده از دستوری که در اسلاید های قبلی بود ربات IRB 140 را فراخوانی می کنیم. سپس با استفاده از دستور teach محیط گرافیکی را لود می کنیم.

```
>>init_lib;
```

```
>>robot = load_robot('abb','irb140');
```

```
>>teach
```

۳- کارگیر را از طریق گزینه Load end Tool انتخاب میکنیم. دقت شود مسیر کارگیر به صورت زیر است.


arte/robots/equipment/end_tools/parallel_gripper_0

۴- با کلیک روی گزینه Load equipment میز را از مسیر زیر به محیط شبیه ساز می آوریم.

arte/robots/equipment/tables/table_two_areas

۵- قطعه کار را نیز با استفاده از گزینه Load a piece و مسیر زیر می آوریم.

arte/robots/equipment/cylinders/cilynder_tiny

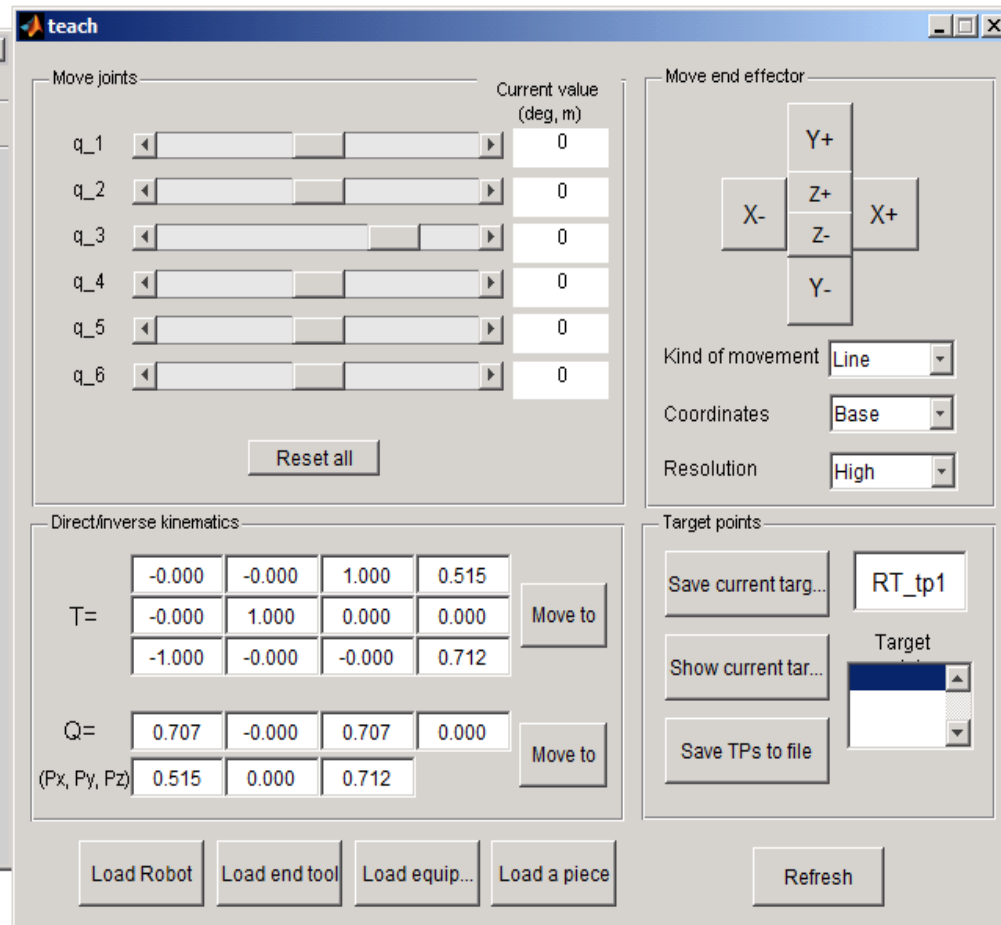
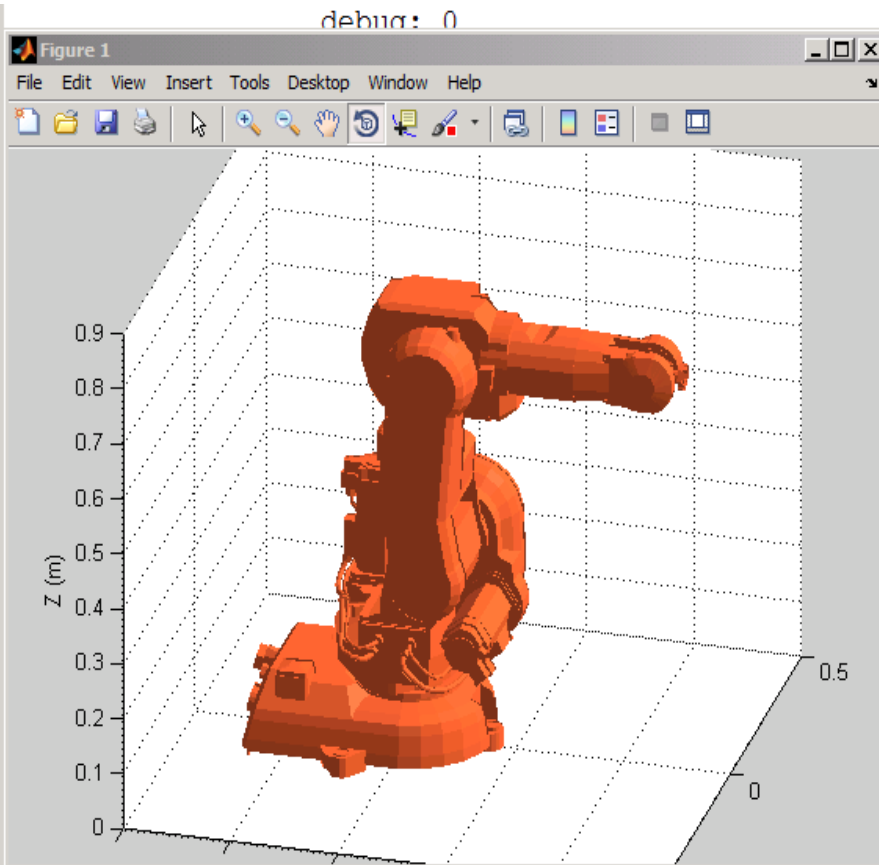
۶- سرانجام با کلیک روی گزینه  در نوار ابزار می توانیم شاهد حرکت شبیه سازی شده ربات برای برداشتن قطعه از موقعیت الف و گذاشتن آن در موقعیت ب می باشد.

برای حذف محورهای مختصات برای اینکه شلوغی کار را کاهش دهیم با استفاده از دستورات زیر
منحنی ها را حذف میکنیم.

```
>> robot.graphical.draw_axes=0;
```

```
>> drawrobot3d(robot,[0 0 0 0 0 0])
```

```
>> teach
```



```
>> robot.graphical.draw_axes=0;  
>> drawrobot3d(robot,[0 0 0 0 0 0])  
>> teach
```

سپاسی از توجه شما...