Dobot Magician & Magician Lite

Overview

A multifunctional robotic arm built for practical training, teaching blockly and script programming, writing and drawing, laser engraving, 3D printing, etc.

Appearance

The robot contains a base, rear arm, forearm and an end-effector.

There are three main joints: J1 (red colour in Figure 2.1), J2 (dark blue colour in Figure 2.1) & J3 (orange colour in Figure 2.1). However, another joint, J4 (green colour in Figure 2.1), is activated when an end-effector with a servo (gripper or suction cup) is attached to the robot. All these joints can rotate.

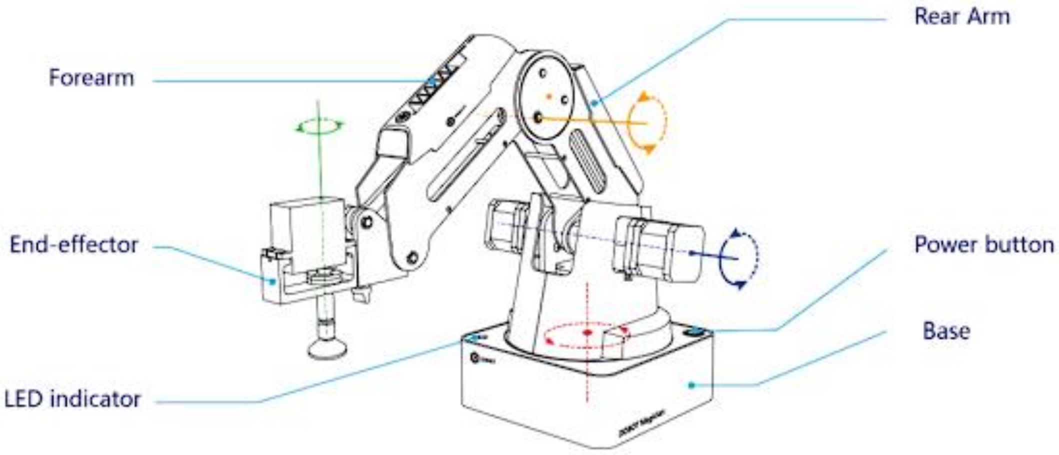
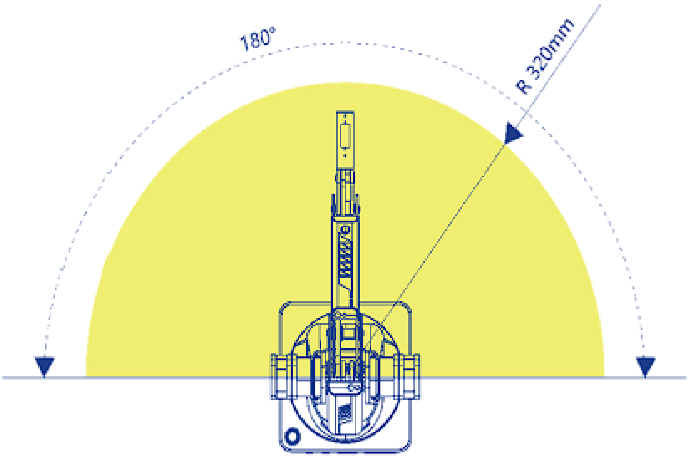


Figure 2.1 Appearance of Dobot Magician

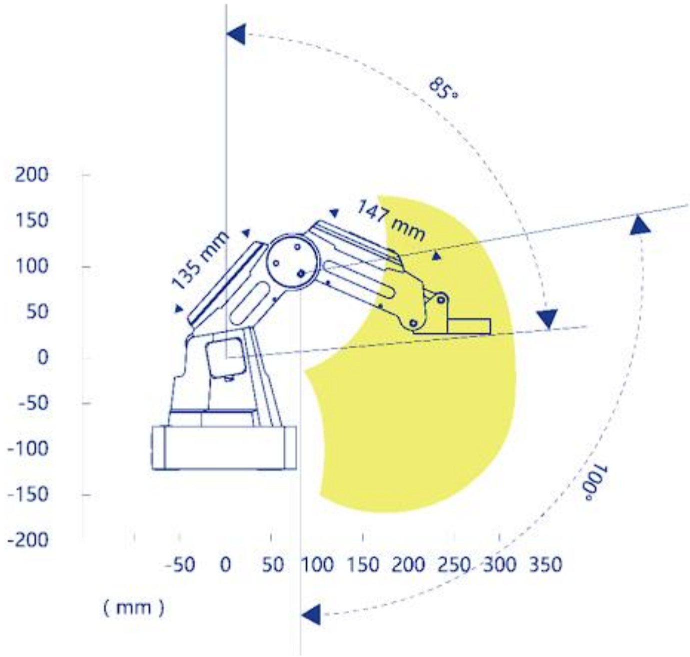
Robot Workspace

The workspace of the robot refers to the area within which its arm can operate, determined by the range of the motion allowed by the robot's joints.

The joint J1 can move an area of 180° (semi-circle) horizontally.



The joint J2 can move an area of 85° vertically. The joint J3 can move an area of 100° vertically.



Coordinate Systems

The robot can be controlled via the Joint Coordinate System or the Cartesian Coordinate System.

Joint Coordinate System: Uses the joints and when these joints are rotated in a positive direction on the controller, they move counterclockwise.

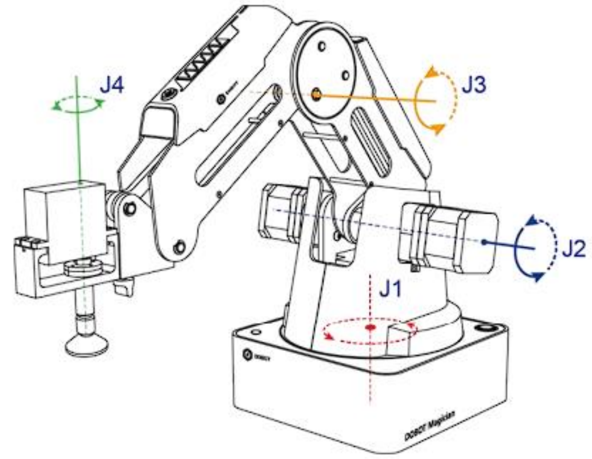


Figure 2.4 Joint coordinate system

Cartesian Coordinate System: Uses X, Y, Z coordinates. The starting position of the end-pointer is x=0, y=0, z=0. R-axis is activated when servo tool is attached.

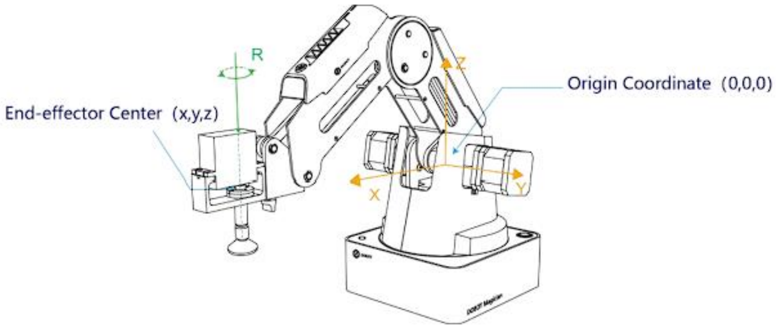


Figure 2.5 Cartesian coordinate system

Motion

The robot moves using the following motions: Jogging, & Point to Point.

Jogging: Moves the robot along Joint Coordinate System or Cartesian Coordinate System.

JCS: Moving the robot by pressing the J1+, J1-, J2+, J2-, J3+, J3- (and J4+, J4-) controls.

CCS: Moving the robot by pressing the X+, X-, Y+, Y-, Z+, Z- (and R+, R-) controls.

Point to Point (PTP): Moves the robot from point to point using the following modes: MOVJ, MOVL, JUMP & ARC.

MOVJ: Joint movement. Used when the trajectory is not important, and speed is required. Each joint moves independently.

MOVL: Linear movement. Used when the trajectory is important, and precision is required. All joints move together.

JUMP: Jump movement. Used when performing pick-and-place tasks. Picks an object from point A, moves horizontally, and places down the object at point B.

ARC: Arc movement. Used when drawing/tracing an arc.

Operating Magician Lite in Virtual Simulation Lab

Requirements

- Computer
- DobotLab Software | Download Link: https://www.dobot-robots.com/service/download-center

Set-up

- Open DobotLab on your computer

- Click and run the Virtual Simulation Lab with Python Lab.

- Select a scene of your choice and it will open the simulated environment.

- You can then connect to Magician Lite from the top right of the Virtual Reality panel.

- Once connected, you can change the end-effectors and the objects from the left of the Virtual Reality panel.

- The script panel is in the middle of the screen, and this is where you can write your scripts.

- Import DobotEDU library by writing the following code:
  from DobotEDU import *

- To enable the end-effector suction cup, write the following code:
  m_lite.set_endeffector_suctioncup(enable=True, on=True)

  enable: Specify whether the suction cup is enabled or not. True is enable, False is disabled.
  on: Specify whether the suction cup is gripping or releasing. True is grip, False is release.

- To enable the end-effector gripper, write the following code:
  m_lite.set_endeffector_gripper(enable=True, on=True)

  enable: Specify whether the gripper is enabled or not. True is enable, False is disable.

on: Specify whether the gripper is gripping or releasing. True is grip, False is release.

Motion

- To move the robot using PTP method, write the following code: m_lite.set_ptpcmd(ptp_mode=1, x=200, y=20, z=10, r=0)

  For the mode, values 0-9 can be entered and for x,y,z,r enter your own values:

ptp_mode	Motion Type	Coord Type	System	Description
0	JUMP	Absolute	Cartesian	Jump to target point
1	MOVJ	Absolute	Cartesian	Joint motion to point
2	MOVL	Absolute	Cartesian	Linear motion to point
3	JUMP	Absolute	Cartesian	Same as 0
4	MOVJ	Absolute	Cartesian	Same as 1
5	MOVL	Absolute	Cartesian	Same as 2
6	MOVJ	Incremental	Joint	Rotate each joint by an amount
7	MOVL	Incremental	Cartesian	Move in a straight line by offset
8	MOVJ	Incremental	Cartesian	Move in a curved path by offset
9	JUMP/MOVL	Incremental	Cartesian	Jump up then linear motion

- To reset the robot's position, write the following code:
  m_lite.set_homecmd()

Get Methods

- To get the end-effector type, write the following code:
  type = m_lite.get_end_effector_type()
  print(type)

Return Value	End-effector Type
0	No end tool
1	Suction Cup
2	Gripper
3	Pen

- To get the real-time position of the robot, write the following code:
  result = m_lite.get_pose()
  print(result)

- To get the speed of the robot arm, write the following code:
  result = m_lite.get_armspeed_ratio(get_type=1)
  print(result)

  get_type: Could be either 0 or 1. 0 is for Jogging motion, 1 is for PTP motion

- To check if the robot lost any steps, write the following code:
  result = m_lite.get_lost_step_result()

```
print(result)
```

Return Value	Description
0	No steps lost
1 or higher	Steps lost

Set Methods

- To set a threshold for lost-step, write the following code:
  ```
  m_lite.set_lost_step_params(value=10)
  ```

- To set the speed of the arm, write the following code:
  ```
  m_lite.set_armspeed_ratio(set_type=1, set_value=50)
  ```

  set_type: Could be either 0, 1, 2 or 3. 0 is Jogging, 1 is PTP, 2 is continuous path and 3 is circular path.
  set_value: Could be from 0-100. Sets the speed percentage of the arm.

- To set the parameters for JUMP movement, write the following code:
  ```
  m_lite.set_ptpjump_params(z_limit=100, jump_height=100)
  ```

  z_limit: Specify the maximum lifting height.
  jump_height: Specify the lifting height.

- To set a waiting time, write the following code:
  ```
  m_lite.wait(delay=10)
  ```

- To perform lost-step detection, write the following code:
  ```
  m_lite.set_lost_step_cmd()
  ```

Activity: Sorting Blocks (Basic Scene)

Code

To sort the blocks in the VSL, write the following code:

```
1) from DobotEDU import *


2) m_lite.set_homecmd()


3) m_lite.set_endeffector_suctioncup(enable=True, on=False)
4) m_lite.set_ptpcmd(ptp_mode=2, x=336, y=30, z=-41.91, r=0)
5) m_lite.set_endeffector_suctioncup(enable=True, on=True)
6) m_lite.wait(delay=2)
7) m_lite.set_ptpcmd(ptp_mode=0, x=336, y=30, z=10, r=0)
8) m_lite.set_ptpcmd(ptp_mode=2, x=336, y=70, z=10, r=0)
```

```
9)   m_lite.set_ptpcmd(ptp_mode=0, x=336, y=70, z=-40.22, r=0)
10) m_lite.set_endeffector_suctioncup(enable=True, on=False)
11) m_lite.set_ptpcmd(ptp_mode=2, x=336, y=70, z=10, r=0)


12) m_lite.set_endeffector_suctioncup(enable=True, on=False)
13) m_lite.set_ptpcmd(ptp_mode=2, x=317.65, y=30, z=-41.91, r=0)
14) m_lite.set_endeffector_suctioncup(enable=True, on=True)
15) m_lite.wait(delay=2)
16) m_lite.set_ptpcmd(ptp_mode=0, x=317.65, y=30, z=10, r=0)
17) m_lite.set_ptpcmd(ptp_mode=2, x=336, y=89.64, z=10, r=0)
18) m_lite.set_ptpcmd(ptp_mode=0, x=336, y=89.64, z=-40.22, r=0)
19) m_lite.set_endeffector_suctioncup(enable=True, on=False)
20) m_lite.set_ptpcmd(ptp_mode=2, x=336, y=89.64, z=10, r=0)


21) m_lite.set_endeffector_suctioncup(enable=True, on=False)
22) m_lite.set_ptpcmd(ptp_mode=2, x=299.30, y=30, z=-41.91, r=0)
23) m_lite.set_endeffector_suctioncup(enable=True, on=True)
24) m_lite.wait(delay=2)
25) m_lite.set_ptpcmd(ptp_mode=0, x=299.30, y=30, z=10, r=0)
26) m_lite.set_ptpcmd(ptp_mode=2, x=319.10, y=70, z=10, r=0)
27) m_lite.set_ptpcmd(ptp_mode=0, x=319.10, y=70, z=-41.91, r=0)
28) m_lite.set_endeffector_suctioncup(enable=True, on=False)
29) m_lite.set_ptpcmd(ptp_mode=2, x=319.10, y=70, z=10, r=0)


30) m_lite.set_endeffector_suctioncup(enable=True, on=False)
31) m_lite.set_ptpcmd(ptp_mode=2, x=280.95, y=30, z=-41.91, r=0)
32) m_lite.set_endeffector_suctioncup(enable=True, on=True)
33) m_lite.wait(delay=2)
34) m_lite.set_ptpcmd(ptp_mode=0, x=280.95, y=30, z=10, r=0)
35) m_lite.set_ptpcmd(ptp_mode=2, x=318.97, y=89.64, z=10, r=0)
36) m_lite.set_ptpcmd(ptp_mode=0, x=318.97, y=89.64, z=-41.91, r=0)
37) m_lite.set_endeffector_suctioncup(enable=True, on=False)
38) m_lite.set_ptpcmd(ptp_mode=2, x=318.97, y=89.64, z=10, r=0)


39) m_lite.set_homecmd()
```

Explanation

Line 1 imports the library required to control the robot

Line 2 resets the arm position

Line 3 activates the suction cup, but it stays off

Line 4 moves the arm to the position of the red block

Line 5 turns the suction cup on to hold the block

Line 6 sets a delay time

Line 7 moves the arm in the upward direction while it’s holding the block

Line 8 hovers the arm to new a direction with the block

Line 9 moves the arm in the downward direction with the block

Line 10 turns the suction cup off to release the block

Line 11 moves the empty arm in the upward direction

The remaining lines perform the same tasks on different blocks at different coordinates.

Activity: Stacking Blocks (Basic Scene)

Code

To stack the blocks in the VSL, write the following code:

```
1)  from DobotEDU import *


2)  m_lite.set_homecmd()


3)  m_lite.set_endeffector_suctioncup(enable=True, on=False)
4)  m_lite.set_ptpcmd(ptp_mode=2, x=336, y=-30, z=-42.38, r=0)
5)  m_lite.set_endeffector_suctioncup(enable=True, on=True)
6)  m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-30, z=20, r=0)
7)  m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=20, r=0)
8)  m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=-42.38, r=0)
9)  m_lite.set_endeffector_suctioncup(enable=True, on=False)


10) m_lite.set_homecmd()
11) m_lite.set_ptpcmd(ptp_mode=2, x=317.65, y=-30, z=-42.38, r=0)
12) m_lite.set_endeffector_suctioncup(enable=True, on=True)
13) m_lite.set_ptpcmd(ptp_mode=9, x=317.65, y=-30, z=20, r=0)
14) m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=20, r=0)
15) m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=-30, r=0)
16) m_lite.set_endeffector_suctioncup(enable=True, on=False)


17) m_lite.set_homecmd()
18) m_lite.set_ptpcmd(ptp_mode=2, x=299.30, y=-30, z=-42.38, r=0)
19) m_lite.set_endeffector_suctioncup(enable=True, on=True)
20) m_lite.set_ptpcmd(ptp_mode=9, x=299.30, y=-30, z=20, r=0)
21) m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=20, r=0)
22) m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=-20, r=0)
```

```
23) m_lite.set_endeffector_suctioncup(enable=True, on=False)


24) m_lite.set_homecmd()
25) m_lite.set_ptpcmd(ptp_mode=2, x=285, y=-30, z=-42.38, r=0)
26) m_lite.set_endeffector_suctioncup(enable=True, on=True)
27) m_lite.set_ptpcmd(ptp_mode=9, x=285, y=-30, z=20, r=0)
28) m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=20, r=0)
29) m_lite.set_ptpcmd(ptp_mode=9, x=336, y=-69, z=-10, r=0)
30) m_lite.set_endeffector_suctioncup(enable=True, on=False)


31) m_lite.set_homecmd()
```

Explanation

Line 1 imports the library required to control the robot

Line 2 resets the arm position

Line 3 activates the suction cup, but it stays off

Line 4 moves the arm to the position of the green block

Line 5 turns the suction cup on to hold the block

Line 6 moves the arm in the upward direction while it’s holding the block

Line 7 hovers the arm to new a direction with the block

Line 8 moves the arm in the downward direction with the block

Line 9 turns the suction cup off to release the block

Line 10 resets the arm position

The remaining lines perform the same tasks on the other green blocks and places them on top of each other.

Activity: Sorting (Factory Scene)

Code

To move boxes in the VSL, write the following code:

```
1.  from DobotEDU import *


2.  m_lite.set_homecmd()


3.  m_lite.set_endeffector_gripper(enable=True, on=False)
4.  m_lite.set_ptpcmd(ptp_mode=2, x=225.47, y=-40.26, z=19.24, r=0)
5.  m_lite.set_endeffector_gripper(enable=True, on=True)
```

```
6.  m_lite.set_ptpcmd(ptp_mode=9, x=225.47, y=-40.26, z=130, r=0)
7.  m_lite.set_ptpcmd(ptp_mode=9, x=69.20, y=-209.92, z=130, r=0)
8.  m_lite.set_ptpcmd(ptp_mode=9, x=69.20, y=-209.92, z=20, r=0)
9.  m_lite.set_endeffector_gripper(enable=True, on=False)


10. m_lite.set_homecmd()
11. m_lite.set_endeffector_gripper(enable=True, on=False)
12. m_lite.set_ptpcmd(ptp_mode=2, x=225.90, y=-120, z=19.24, r=0)
13. m_lite.set_endeffector_gripper(enable=True, on=True)
14. m_lite.set_ptpcmd(ptp_mode=9, x=225.47, y=-120, z=130, r=0)
15. m_lite.set_ptpcmd(ptp_mode=9, x=69.20, y=-209.92, z=130, r=0)
16. m_lite.set_ptpcmd(ptp_mode=9, x=69.20, y=-209.92, z=40, r=0)
17. m_lite.set_endeffector_gripper(enable=True, on=False)


18. m_lite.set_homecmd()
```

Explanation

Line 1 imports the library required to control the robot

Line 2 resets the arm position

Line 3 activates the gripper, but it stays off

Line 4 moves the arm to the position of the white box

Line 5 turns the gripper on to close the claw and grab the box

Line 6 lifts the arm up

Line 7 moves the arm to a new position

Line 8 moves the arm down

Line 9 turns the gripper off, so the box is placed on the floor

The remaining lines perform the same tasks on the brown box and places it on top of the white box.

Operating Magician Robot in Real Life

Requirements

- Computer
- Dobot Magician
- Power Cable
- Data Cable

Set-up

- Connect the power cable to the robot and a power outlet.

- Connect the data cable to the robot and your computer.

- Turn the robot on.

- Open DobotLab on your computer.

Activity: Writing

1) Attach the pen end-effector to the robot.

2) Select Writing and Drawing Lab on DobotLab.

3) Connect the robot by selecting ‘Magician’ on the top right of the screen.

4) Enter a word in the text box labelled 'Click here to insert’.

5) Press the ‘Add’ button.

6) Place the word in the workspace shown on the screen.
   You can resize the word by dragging the corners of the text.

7) Place some papers or a notebook under the end-effector.

8) Press ‘Run program’ to make the robot write your given text.