52-O: Robot Exercises

Instructions

- 5 robot exercises.
- Include code, images, and equations where appropriate.
- When you are finished, compile this document to a PDF and submit it directly to Gradescope.
- This assignment is **fixed length**, and the pages have been assigned for you in Gradescope. As a result, **please do NOT add any new pages**. We will provide ample room for you to answer the questions. If you *really* wish for more space, please add a page *at the end of the document.*
- **We do NOT expect you to fill up each page with your answer.** Some answers will only shorter, and that is okay.

1. Fundamentals

In this course, you will work with a tabletop robot arm called the **Mirobot**. This first exercise will help you learn the fundamentals of using this robot.

1A. Robot Use Agreement. Read carefully the Do's and Don'ts with handling the robot on this Google Form and submit the form. Once done, please attach a screenshot of your submission in the space provided.

⚠ <u>**Note**</u>: You will need to fill out this form **every single time** you use the robot.

1B. Install the latest version of the WLKATA Studio software here. For Windows users, additionally install the driver (more instructions can be found here.) Once done, please attach a screenshot of the WLKATA Studio software running on your computer in the space provided.

1C. Set up hardware: plug in the USB cable and power on the robot by pressing the on/off button on the base of robot, as shown in the image below. Please attach a photo of the robot switched on in the space provided.

Powering on the Mirobot. The power button is to the right of the one the person above is pressing.

1D. Check connection.

1. Open WLKATA Studio.

2. If the upper left corner of the WLKATA Studio interface displays CONNECTED, you are successfully connected. Otherwise, you may need to change the COM setting (more details).

3. When the robotic arm is powered on, or when the serial port connection is first established, each axis of the robotic arm is locked. To unlock the axes and perform any operations, the robotic must be homed. Click the HOMING button in the WLKATA Studio. Wait for the manipulator to be homed. Please attach a photo of your robot after the homing operation. It should look like the one below.

The correct position of the manipulator after a successful HOMING action

1E. Inverse Kinematics.
Set the robot mode to **COORD MODE** (more details) within the studio to move the robot arm to a target position marked `TARGET A` on the mat. Submit photo result in the space provided.

1F. Forward Kinematics.
Set the robot to **JOINT MODE** (more details) within the studio to move the robot arm to a target position marked `TARGET B` on the mat. Submit photo result in the space provided.

Answers. Please do not exceed the height provided for each answer image.

1A. Robot Use Agreement

YOUR_ANSWER.png

1B. WLKATA Studio

YOUR_ANSWER.png

1C. Set up hardware

YOUR_ANSWER.png

1D. Check Connection

YOUR_ANSWER.png

1E. Inverse Kinematics

YOUR_ANSWER.png

1F. Forward Kinematics

YOUR_ANSWER.png

Additional Space. Please do not exceed this page for this question.

2. Python API

For this section, you may reference the template code or the documentation. This handbook may also be useful for additional information. For all subsequent exercises, we will create a Python virtual environment using Anaconda. Please install Anaconda as instructed in the above link, then create a new environment:

```
conda create -n 52-O python=3.8
```

Then activate this virtual environment. You must do this for all subsequent exercises even if not instructed.

```
conda activate 52-O
```

2A. Install Python SDK: `pip install wlkata-mirobot-python`. Attach at screenshot that shows successful completion in the space provided.

2B. Create robot arm object and home. Attach a screenshot of successful command execution.

1. `arm = WlkataMirobot(portname="your port name")`
2. `arm.home()`

2C. Replicate **1E** and **1F** using the Python API. Consult the documentation to figure out the values for `joint_angles`, `x, y, z`, etc. Submit photo result and include your code in the space provided.

- `def set_joint_angle(self, joint_angles, is_relative=False, speed=None, wait_ok=None)`
- `def set_tool_pose(self, x=None, y=None, z=None, roll=0.0, pitch=0.0, yaw=0.0, mode='p2p', speed=None, is_relative=False, wait_ok=True)`

2D Utilize the below 4 interpolation functions to make the robot go from `TARGET A` to `TARGET B` on the mat. Submit photo result of the arm in **both** targets in the space provided. Please also include your code where appropriate.

- Point2Point: `def p2p_interpolation(self, x=None, y=None, z=None, a=None, b=None, c=None, speed=None, is_relative=False, wait_ok=None)`
- Linear: `def linear_interpolation(self, x=None, y=None, z=None, a=None, b=None, c=None, speed=None, is_relative=False, wait_ok=None)`

- Door: `def door_interpolation(self, x=None, y=None, z=None, a=None, b=None, c=None, speed=None, is_relative=False, wait_ok=None)`
- Circular: `def circular_interpolation(self, ex, ey, radius, is_cw=True, speed=None, wait_ok=None)`

Answers. Please do not exceed the height provided for each answer image.

2A. Install Python SDK

YOUR_ANSWER.png

2B. Create Python Object and Home

YOUR_ANSWER.png

2C. IK / FK using Python Include your code also.

YOUR_ANSWER.png

Inverse Kinematics to `TARGET A`

YOUR_ANSWER.png

Forward Kinematics to `TARGET A`

Answer for 2C.

```
var = 'YOU CODE GOES HERE'
```

2D. Interpolation Include your code also.

YOUR_ANSWER.png

Point2Point: `TARGET A`

YOUR_ANSWER.png

Point2Point: `TARGET B`

Answer for 2C.

```
var = 'YOU CODE GOES HERE'
```

YOUR_ANSWER.png

Linear: TARGET A

YOUR_ANSWER.png

Linear: TARGET B

Answer for 2C.

```
var = 'YOU CODE GOES HERE'
```

YOUR_ANSWER.png

Door: `TARGET A`

YOUR_ANSWER.png

Door: `TARGET B`

Answer for 2C.

```
var = 'YOU CODE GOES HERE'
```

YOUR_ANSWER.png

Circular: `TARGET A`

YOUR_ANSWER.png

Circular: `TARGET B`

Answer for 2C.

```
var = 'YOU CODE GOES HERE'
```

Additional Space. Please do not exceed this page for this question.

3. End Effectors

For this section, you may reference the template code or the documentation. You will perform the same task using 3 different end effectors.

Gripper

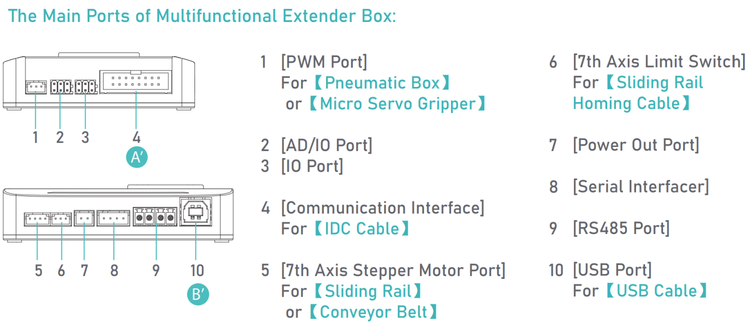
Flexible Claw

Suction Cup

Types of end effectors.

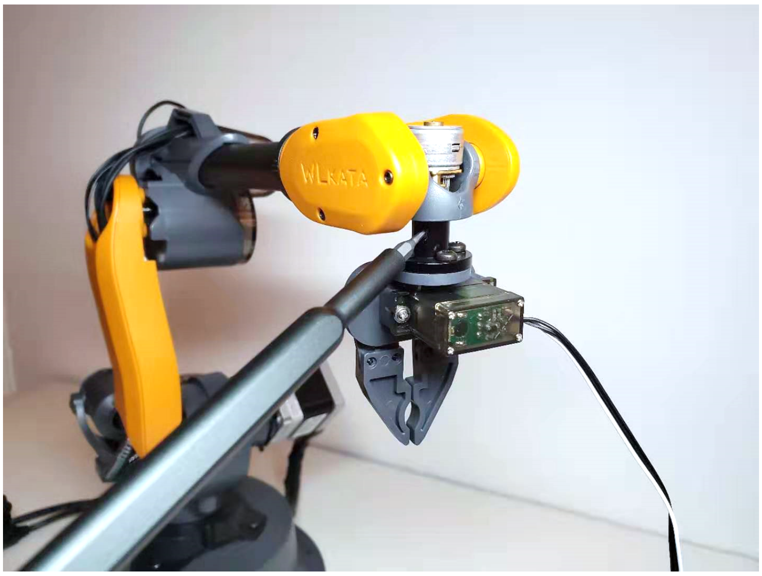
3A. Change end effectors. Attach a picture of each after successfully attaching each end effector.

1. Power off the robot
2. For the gripper, connect it to the correct port on the Multifunctional Extender Box (MEB). For the claw and suction cup, connect the pneumatic pump to the MEB. Make sure to connect the pump to the gripper as needed.



Use port 1 to connect the end effectors

3. Connect the MEB to the robot.
4. Screw on the end effector (an Allen wrench is the the tool box). Be careful to connect all wires before screwing on the end effector.



How to screw on the end effector. Use the Allen wrench in the toolbox.

3B. Use the gripper (2 finger) to grab any toy block from the table. Submit photo result of the robot successfully lifting a block in the space provided.

1. Complete the task using the studio software
2. Complete the task using Python API
    - `from wlkata_mirobot import WlkataMirobotTool`
    - `arm.set_tool_type(WlkataMirobotTool.GRIPPER)`
    - `arm.gripper_open()`
    - `arm.gripper_close()`
    - `arm.set_gripper_spacing(spacing_mm)`

3C. Use the flexible claw (3 finger soft gripper) to grab any block from the table. Submit photo result of the robot successfully lifting a block in the space provided.

1. Complete the task using the studio software
2. Complete the task using Python API
    - For flexible claws, air pump suction represents opening and blowing represents closing.
    - `arm.set_tool_type(WlkataMirobotTool.FLEXIBLE_CLAW)`
    - `arm.pump_suction()`
    - `arm.pump_blowing()`

3D. Use the suction cup to grab any block from the table. Submit photo result of the robot successfully lifting a block in the space provided.

1. Complete the task using the studio software
2. Complete the task using Python API
   - `arm.set_tool_type(WlkataMirobotTool.FLEXIBLE_CLAW)`
   - `arm.pump_suction()`
   - `arm.pump_blowing()`

Image showing how to use the pneumatic pump.

Answers. Please do not exceed the height provided for each answer image.

YOUR_ANSWER.png	YOUR_ANSWER.png	YOUR_ANSWER.png
Mirobot with Gripper.	Mirobot with Flexible Claw.	Mirobot with Suction Cup.

Answer for 3A.

3A. Change End Effectors

```
var = 'YOU CODE GOES HERE'
```

YOUR_ANSWER.png

Gripper lifting a block (WLKATA Studio).

YOUR_ANSWER.png

Gripper lifting a block (Python).

Answer for 3B.

3B. Using Gripper.

```
var = 'YOU CODE GOES HERE'
```

YOUR_ANSWER.png

Flexible Claw lifting a block (WLKATA Studio).

YOUR_ANSWER.png

Flexible Claw lifting a block (Python).

Answer for 3C.

3C. Using Flexible Claw.

```
var = 'YOU CODE GOES HERE'
```

YOUR_ANSWER.png

Suction Cup lifting a block (WLKATA Studio).

YOUR_ANSWER.png

Suction Cup lifting a block (Python).

Answer for 3D.

3D. Using Suction Cup.

```
var = 'YOU CODE GOES HERE'
```

Additional Space. Please do not exceed this page for this question.

4.ROS Integration

4A. Install ROS kinetic here. Be sure to select the "kinetic" tab after clicking on the platform option. Make sure to activate the `52-O` conda environment we created earlier. Attach screenshot of successful installation.

4B. Install relevant ROS packages. Make sure to activate the `52-O` conda environment we created earlier. Attach screenshot of successful installation.

- `ros-kinetic-serial`
- `ros-kinetic-ros-control`
- `ros-kinetic-ros-controllers`
- `ros-kinetic-moveit`
- `ros-kinetic-gazebo-ros-pkgs`
- `ros-kinetic-gazebo-ros-control`

4C. Replicate **1E** and **1F** using ROS. Reference the wlkata documentation and github repo. Submit photo result in the space provided.

Answers. Please do not exceed the height provided for each answer image.

4A. Install ROS kinetic.

YOUR_ANSWER.png

4B. Install ROS Packages.

YOUR_ANSWER.png

4C. Inverse Kinematics

YOUR_ANSWER.png

4C. Forward Kinematics

YOUR_ANSWER.png

Additional Space. Please do not exceed this page for this question.

5. Cameras & Mirobot

In addition to the Mirobot, you will also work with the RealSense D400 series of sensors. These sensors have an RGB camera as well as a 3D depth sensor. In this exercise you will learn how to interface with these cameras and calibrate them relative to the robot.

5A. Install OpenCV and RealSense API as instructed below. Attach a final screenshot of your prompt after step 3.

1. Activate your conda environment: `conda activate 52-O`
2. Install OpenCV: `conda install -c conda-forge opencv`
3. Install the `pyrealsense2` library: `pip install pyrealsense2`

5B. Connect the camera to your machine.

The camera uses a USB 3 connection and a USB-C connector. Please connect **both** the camera and the robot into your machine. Please use the provided USB hub, if needed. Attach a photo of the camera and robot attached to your machine.

5C. Visualize camera output using OpenCV.

1. Run the Python script `code/q5c_1.py`. Attach a screenshot of what you see when you run this code.
2. Run the Python script `code/q5c_2.py`. Attach a screenshot of what you see when you run this code.

5D. Camera calibration. Follow this tutorial to understand camera intrinsics and extrinsics calibration. Use the `calibrateCamera` function in OpenCV and implement calibration of both intrinsics and extrinsics. Please attach your code and the intrinsics matrix (`cameraMatrix`) for the RGB camera in Intel RealSense.

You may want to combine code from 5C for this part. A printed 9x6 checkerboard pattern is available in your workspace. You can also find it here.

Answers. Please do not exceed the height provided for each answer image.

5A. Install OpenCV and RealSense API.

YOUR_ANSWER.png

5B. Connect Camera to your machine.

YOUR_ANSWER.png

5C. Visualize using OpenCV.

YOUR_ANSWER.png

5D. Camera Calibration.

```
var = 'YOU CODE GOES HERE'
```

Please also report the matrix stored in the `cameraMatrix` return variable of the `calibrateCamera` function.

Additional Space. Please do not exceed this page for this question.

Feedback

Please help us make the course better. If you have any feedback for this assignment, we'd love to hear it!