Lab 1: Getting Start with Franka Panda

Robot Autonomy

Prof. Oliver Kroemer

0 Prerequisites

In this lab, you will learn how to start the Franka Panda Robot, how the e-stop works, how to control the robot with Python on your computer, and how to use iam-interface.

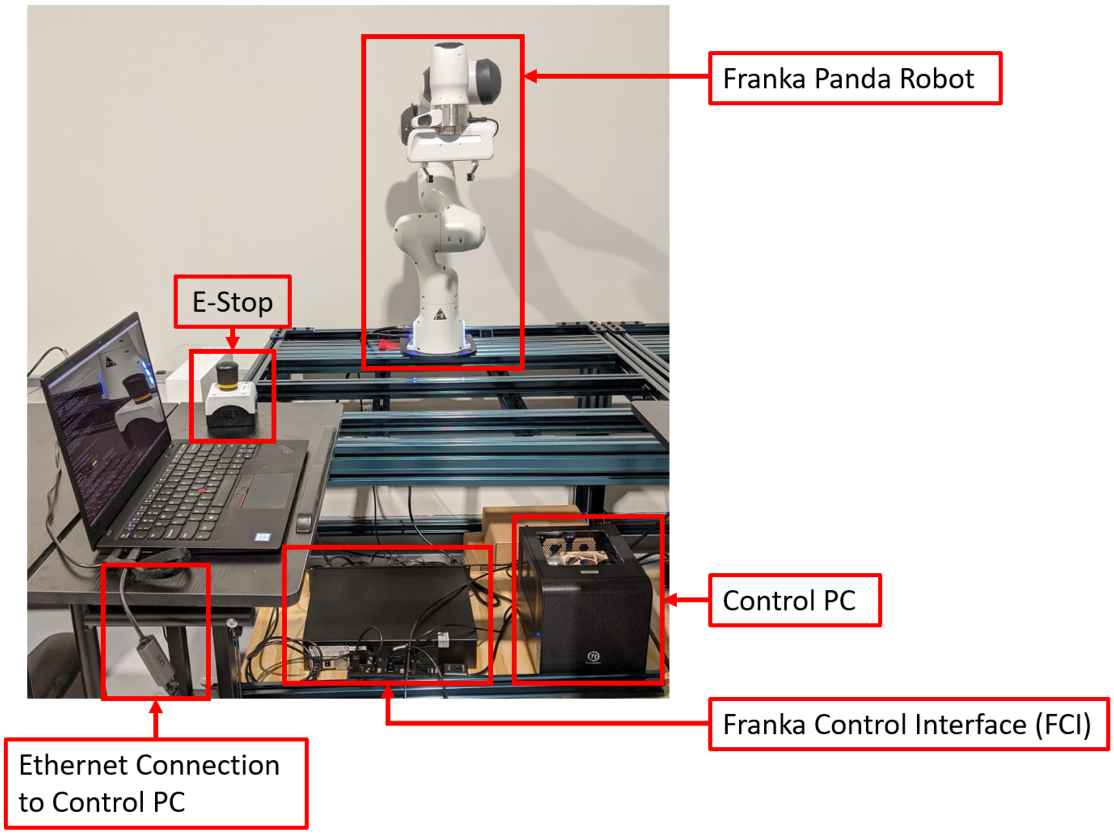


Figure 1: Lab Workstation

1 Turning on and Connecting to the Robot

1. Flip the on switch on the Franka Control Interface and the Intel Nuc

Figure 2: On button for the Franka

2. Unlock the joints on the robots. To do this you'd need to open up a webpage on the Control PC. Do this by:

   ```
   ssh -X student@iam-<name>
   ```

   Launch Firefox by running `firefox` in the terminal.

   (If running `firefox` or `-X` is giving errors, try restarting the terminal.)

   Then open the webpage `172.16.0.2` to unlock joints:

   (If it prompts for a login, the username is `admin`, and the password is `shavethepandas`)

   The joints are unlocked when the indicator lights are solid white.

1.0.1 Note on Safety

The robot has the following modes denoted by the indicator light on the side of the robot base:

1. Yellow - Lock Mode. The robot's joints are locked, and it cannot be moved. Locking and unlocking the joints are done in a webpage on the Control PC (you will do this in a later section). Once the joints are unlocked, the robot can be either in White or Blue.

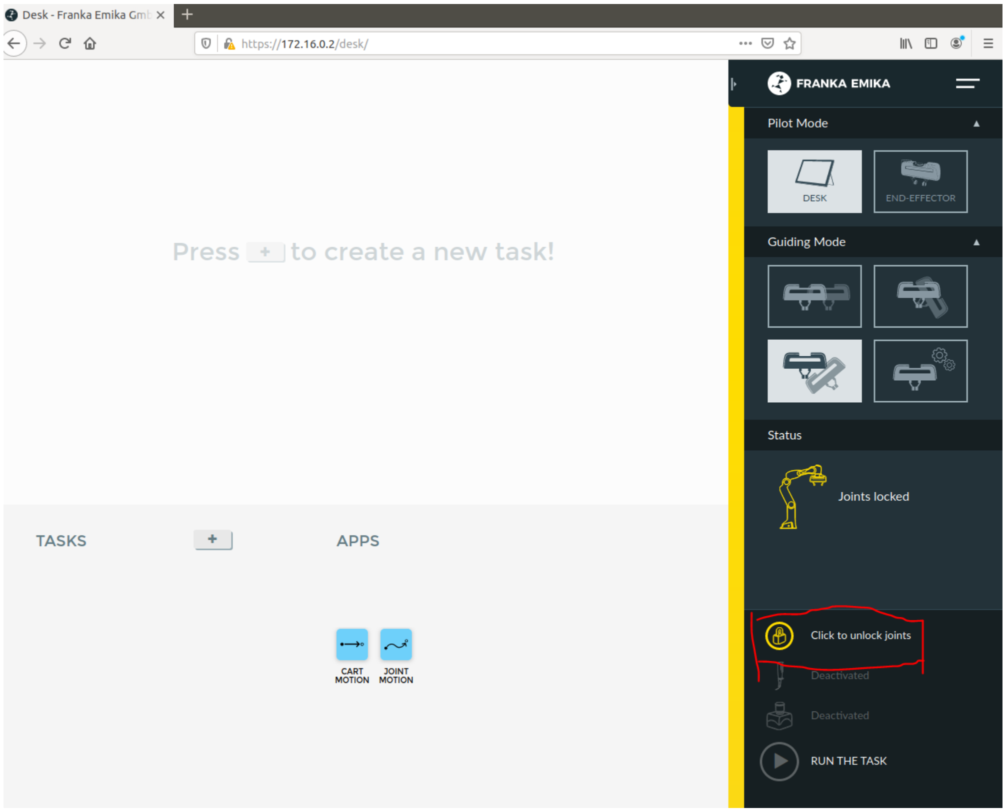


Figure 3: Unlock Joints

2. White - Manual Mode. The robot may be moved by a human pressing down on the gray buttons near the robot hand. However, when the robot is in manual mode, it cannot be commanded via a program. White is achieved when the e-stop is pressed down.

3. Blue - Program Mode. The robot can be commanded with a program, but it cannot be moved by a human pressing down on the gray buttons. To achieve Blue, twist and release the e-stop.

4. Pink - Small error. Someone attempted to move the robot manually while it was in Blue. When in Pink, press down on the e-stop to return the robot to White.

5. Red - Serious error. This happens when the robot incurs a significant collision. A full restart is required to move the robot again.

When running a program that commands the robot, please:

1. Stay outside of the robot's workspace (defined by the edges of the table).

2. Always keep a hand on the e-stop. Under no circumstances should the e-stop be out of reach.

2 Controlling the Robot

1. Physically move the arm to get it out of the initial position. Note, you need a light touch otherwise the robot will lock up.

2. Note, you should be holding the estop whenever the robot could be moving

3. Whenever you hit the estop you will need to restart the control pc. You can do this by running `cd ~/Prog/frankapy`
   `bash bash_scripts/start_control_pc.sh -u student -i iam-<name>`

   This should launch 3 new terminals indicating the server is running on the Control PC.

4. Now you need to enter the IamEnv virtual environment.Use this command to start the virtual env.
   `senv`

5. Run the example script in the `~/Prog/frankapy/scripts` folder to reset the robot's position:

   `python scripts/reset_arm.py`

   This should reset the robot to the initial position. Please do this whenever you are done with the robot.

3 Camera Calibration

3.0.1 Running Instructions

1. Start the Azure Kinect Camera
`roslaunch azure_kinect_ros_driver driver.launch`

2. Make sure the camera is on. You should see a small white light by the lens. You might need to unplug/replug in the camera if you don't see it.

3. Open the iam-interface/camera-calibration/calib/azure_kinect.intr file the text editor of your choice.

4. Rostopic echo /rgb/camera_info once.

5. Copy the first number in K: to after "_fx"

6. Copy the third number in K: to after "_cx"

7. Copy the fifth number in K: to after "_fy"

8. Copy the sixth number in K: to after "_cy"

9. Then start franka-interface and use the following command
(in `~/Prog/iam-interface/camera-calibration`):
`python scripts/register_camera_using_franka.py`
Note: the script will give you 10 seconds to manually move the franka such that the checkerboard is directly under the camera. The Franka should be in roughly this orientation

10. Afterwards, check to make sure the calib/azure_kinect_overhead/azure_kinect_overhead_to_world.tf file is similar to calib/example_azure_kinect_overhead_to_world.tf.

11. If the transformations are similar, then test the calibration using a flat surface and an alphabet block and run the following command: `python scripts/run_pick_up_using_camera.py` This will work by launching a screen, where you place a block in the frame. Then click on the center of the block and press enter. If you calibration is good then it should be able to pick up the block.

12. If it is not similar, uncomment the transformation matrix (in the run_pick_up_using_camera.py file) that makes it match the other transform

4 Basic Motion

4.1 Getting Set Up

1. Go to iam-construct folder: `~/iam-construct`
2. Run the startup script:

   `./iam-construct-run.sh {name of robot}`
3. Navigate to nuc_ip:9080/javascript/example/simple.html in your web browser. We would suggest doing on your phone or another computer, not the nuc

This will allow you to open up the basic interface.

4.2 Navigating to Points

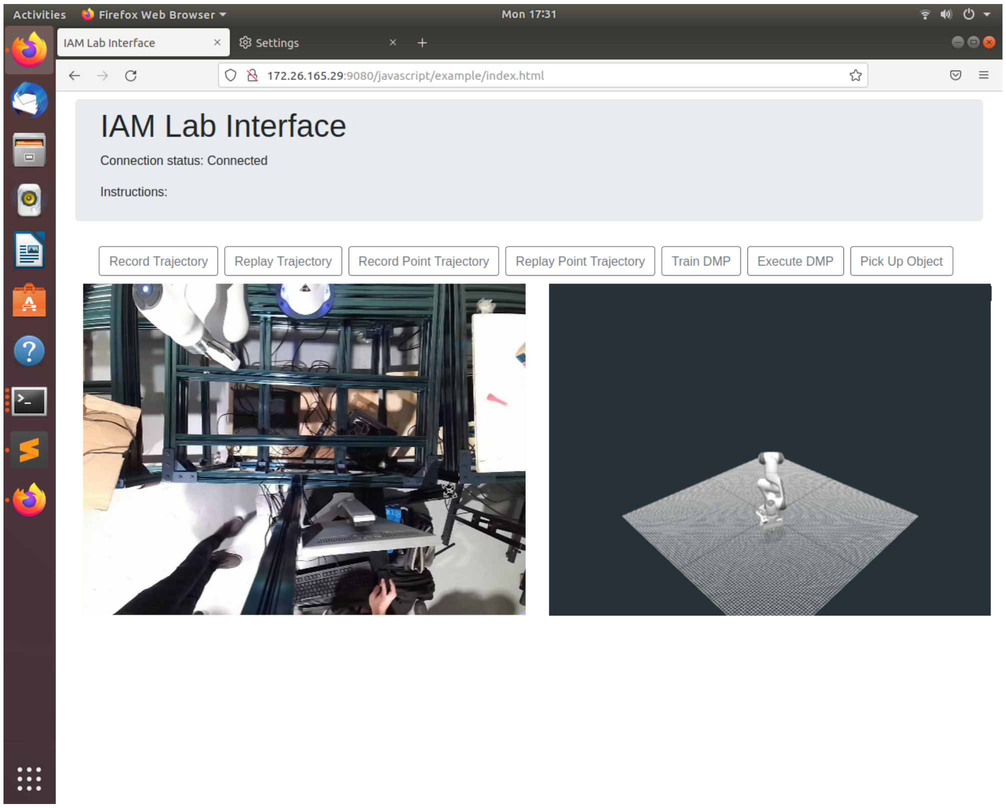
1. With everything running, navigate to the following directory:
   `~/Prog/iam-interface/iam-domain-handler/examples/`
   and launch the point-to-point navigation skill by running the following command:
   `python robot_autonomy_lab_1.py`

4.3 Using the website

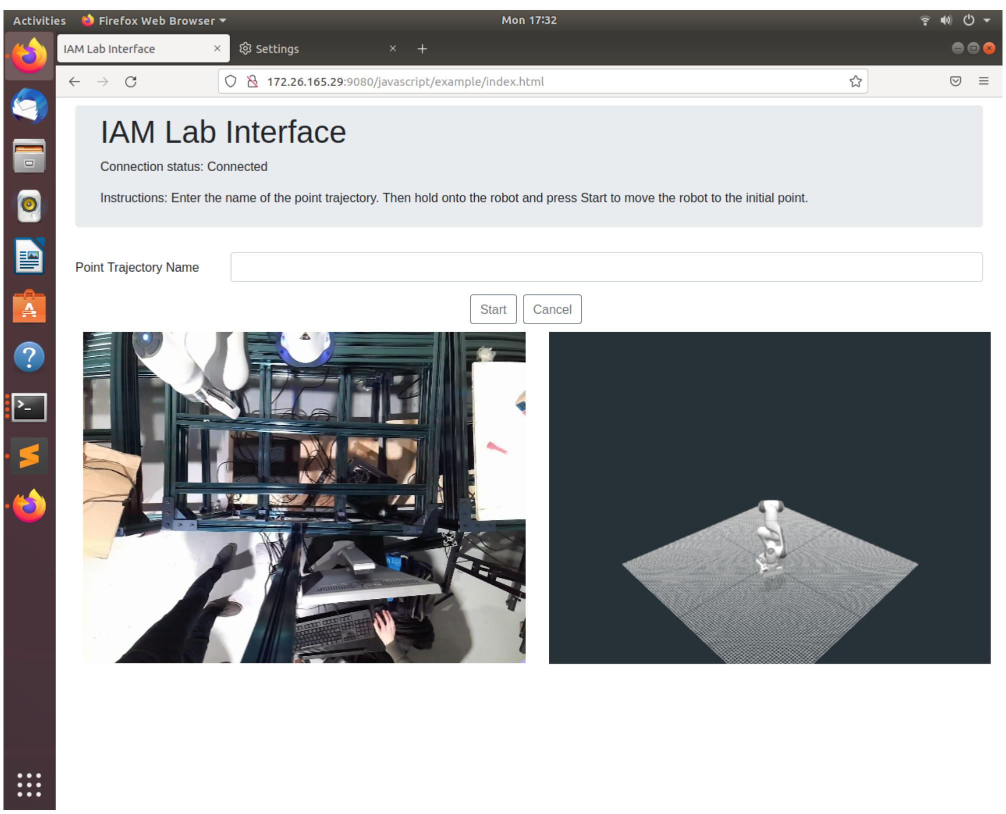
Note: You will want to have the website loaded on your phone before running the python script

You will be using the website to move along a point trajectory

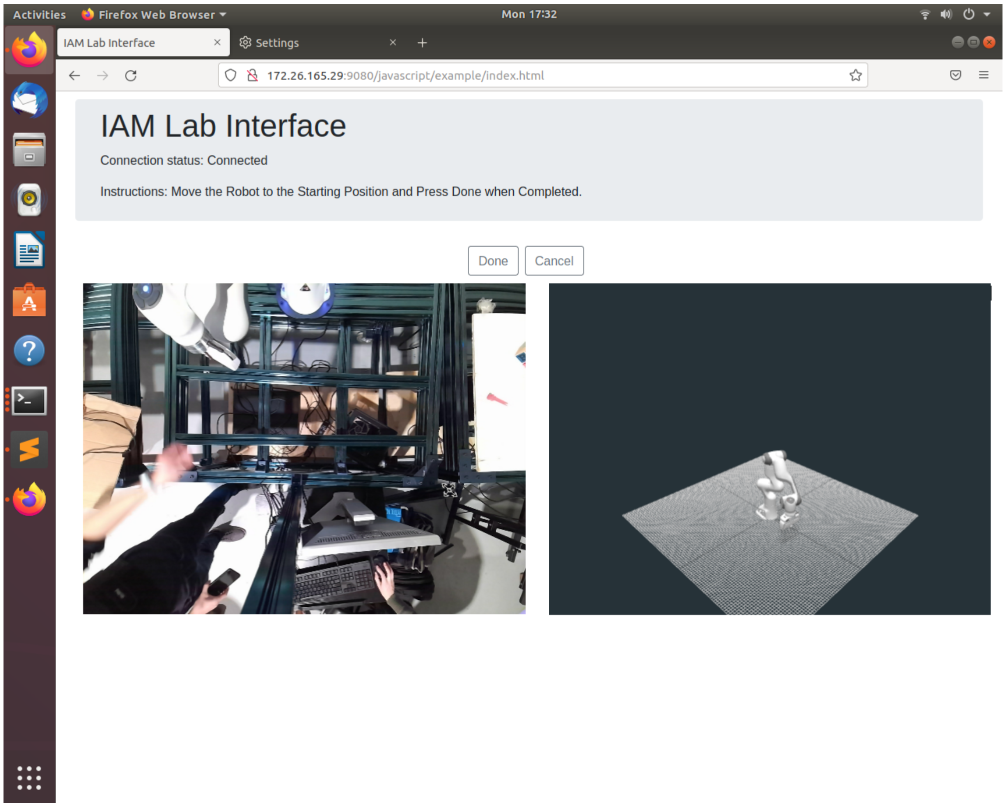
Click the record point trajectory button



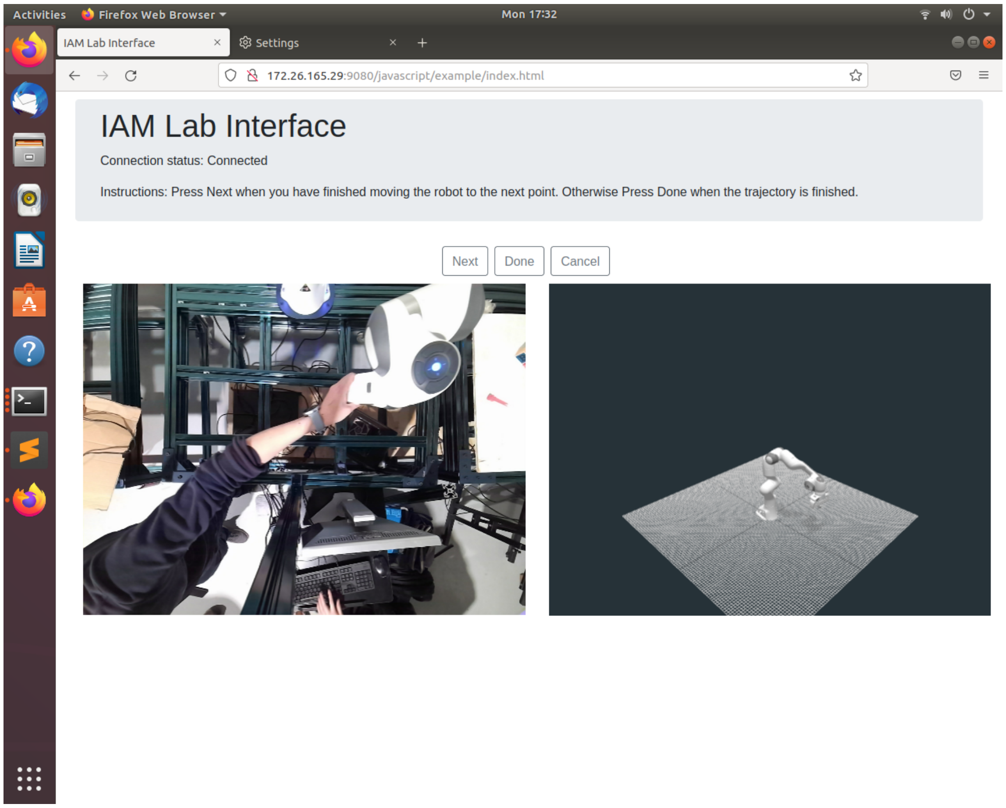
Create a name for the trajectory. You will need to remember it when you replay it.



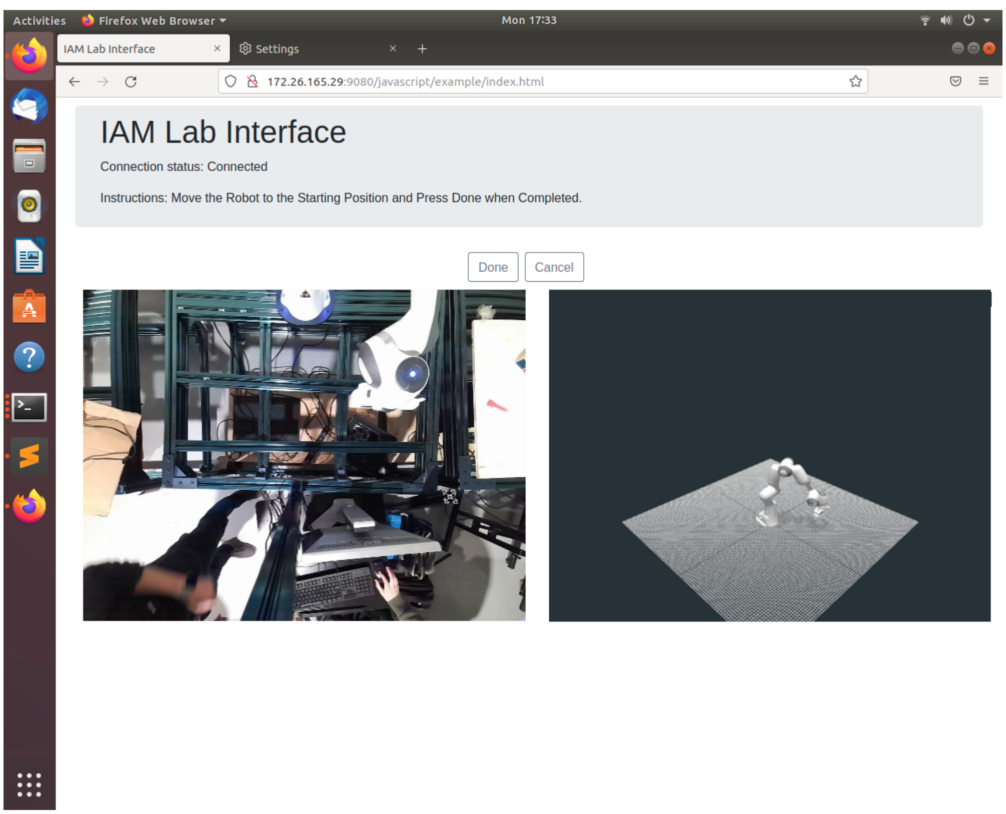
Manually move the robot to the desired position and click done when complete. Do not press the buttons on the side of the end effector. You should be able to move the robot.



Move the robot again to a new position

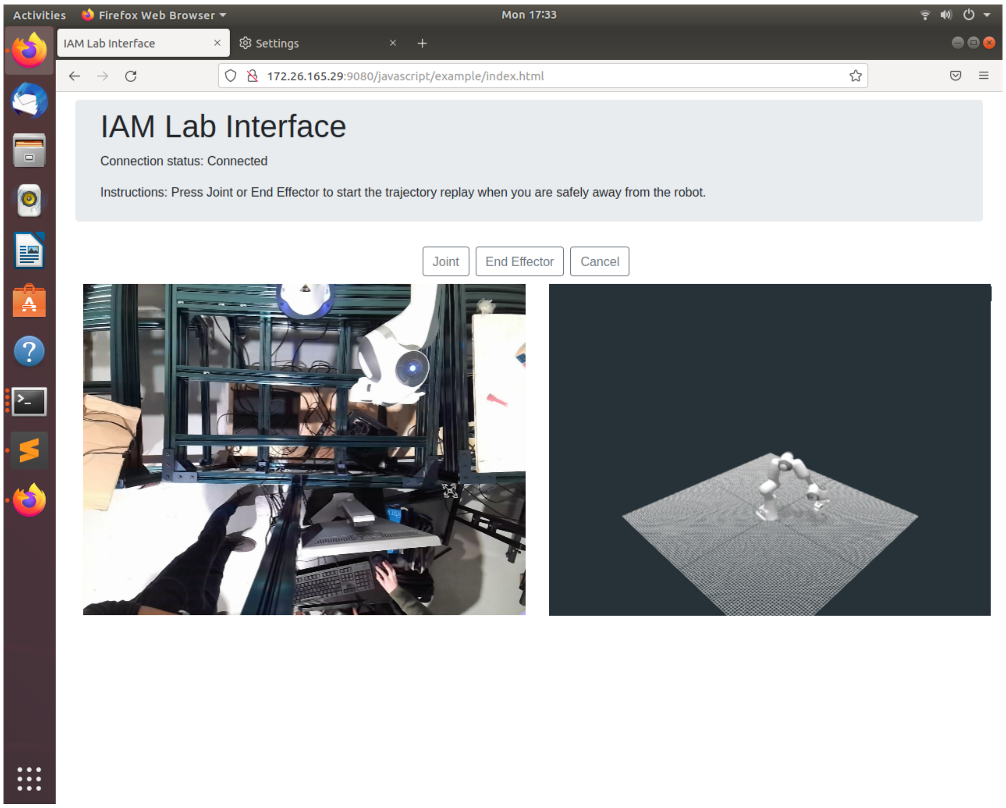


Click done when it is in the proper location

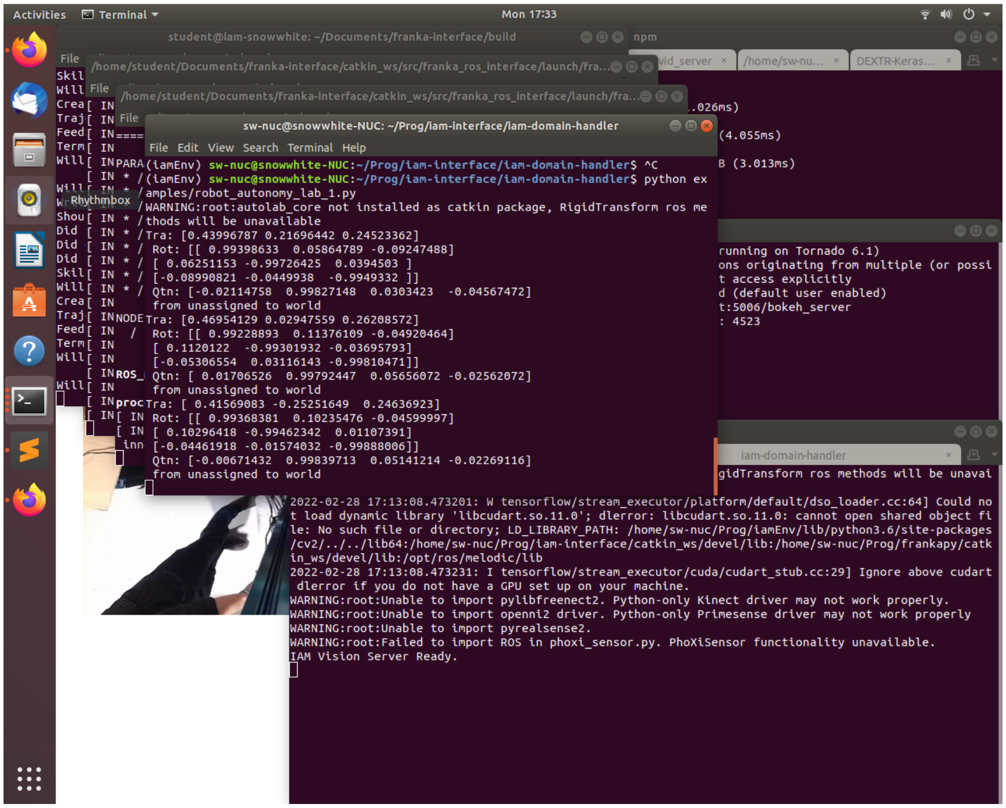


Now we need to replay the point trajectory you created.

1. Click the point trajectory button
2. type in the name of your point trajectory
3. click end effector to move the whole end effector.
4. Watch it move!



Hint for next section: The x,y,z coordinates of your trajectory will be printing out on the terminal like so:



4.4 Creating your own behavior

The goal of this section will be to modify and create your own behavior. The goal is to pick up a cube and place it back. You will need to manually move the

1. Manually move the robot such that it would be picking up the cube
2. Use the interface to get the x,y,z position of the end effector. Hint: This should be printed out in the terminal
3. now that you have the final x,y,z of your trajectory, we need to add it into the new behavior. Please change
`~/Prog/iam-interface/iam-domain-handler/examples/lab1_custom_behavior.py`
Such that it will have the robot move the gripper to the correct location.
4. run this file and have it pick up the cube

Now that you have altered the x, y, z. The next step is have it actually pick up the block. Please add in the behavior to close the gripper on the cube, pick up the cube by returning to the original position, then going back and placing the cube.

Please show the TAs once you have completed this task.

5 Turning Off the Robot

It's important to return the robot to its home position and turn the robot off when it's not being used.

1. Reset the robot to its home position by running the following script in the frankapy-public folder:

   ```
   cd ~/Proc/frankapy
   python scripts/reset_arm.py
   ```

2. Close the terminals related to the Control PC to shut off the remote server.

3. Set the robot to manual mode by pushing down on the e-stop. This should return the indicator light to solid white.

4. Turn off the robot by returning to the webpage on the Control PC and clicking shutdown:

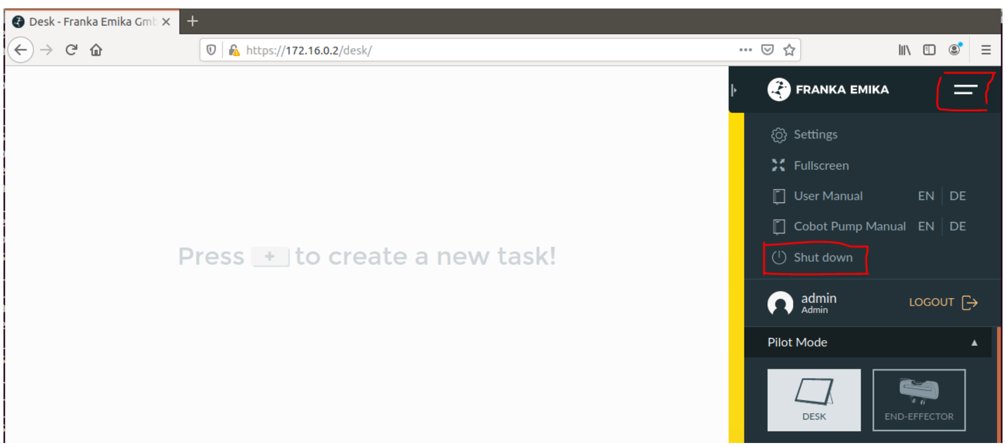


Figure 4: shutdown

Important! Wait for a minute for the robot to shutdown, then flip the physical switch on the FCI (do not flip the switch before the robot shutdown has finished).

6 Appendix

6.1 Ethernet setup

1. Flip on the power switch on the FCI. The robot should now boot up. It will take a couple minutes. The robot has finished booting up when the indicator light is solid yellow.

2. Connect the Ethernet cable from the Control PC to your laptop. Use the provided Ethernet-USB converters if needed.

3. Do this step if you're using the VM, otherwise skip to the next step.

   Before starting the VM, add a "Bridged Adapter" to the second networking interface, and set the attachment to the newly connected Ethernet interface.

   Do this by selecting the VM > Settings > Network > Adapter 2. Check "Enable Network Adapter" > Choose Bridged Adapter > Select the name of the Ethernet interface that's connected to the Control PC.

   The exact name of the Ethernet interface you need to choose may be different from the one shown in the screenshot.

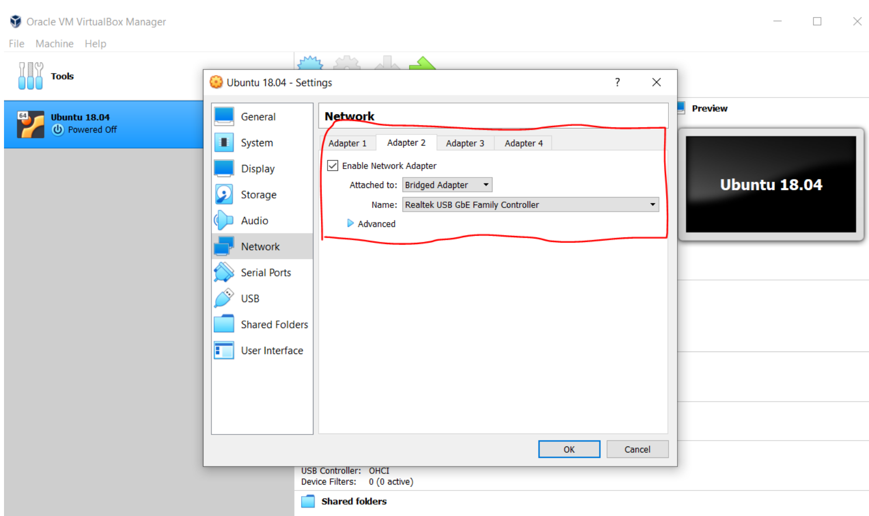
   

   Figure 5: VirtualBox Network Setting

   Now launch the VM, and it'll have access to this Ethernet connection to the Control PC.

   If you're using the VM, the following steps are all done **inside** the VM.

4. Set the IP of your newly connected Ethernet Interface to have a static IPv4 address of `192.168.1.3`. See Figure 6 for details.

5. Add the name and IP of the Control PC to your network host file. The name depends on which robot your workstation has, and it should be in the format of `iam-<name>`. The Control PC IP is `192.168.1.2`.

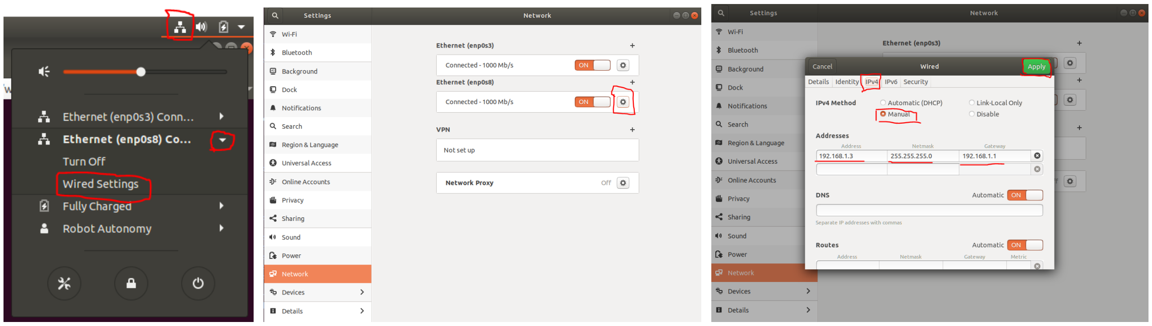


Figure 6: Steps (from left to right) of Setting the IP of the Local Machine in the Ethernet Connection to the Control PC

To do this on Ubuntu, add the following line to the end of `/etc/hosts` (editing this file requires sudo):

```
192.168.1.2          iam-<name>
```

6. Now we will set up SSH keys on the Control PC so that you can SSH into the Control PC without a password. This also allows us to easily remote launch the robot-interface code on the Control PC.

   First we will generate the SSH keys on your machine. Run:

   ```
   ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
   [Press enter]
   [Press enter]
   [Press enter]
   eval "$(ssh-agent -s)"
   ssh-add ~/.ssh/id_rsa
   ```

   You can view the newly generated SSH key:

   ```
   cat ~/.ssh/id_rsa.pub
   ```

   Now we will copy and paste this SSH key to the Control PC. Open a new terminal and ssh to the robot using the command:

   ```
   ssh student@iam-<name>
   ```

   Password is 16-662

   Open this file on the Control PC with a text editor:

   ```
   vim ~/.ssh/authorized_keys
   ```

   Press the letter `i`.

   Press the `enter` button on your computer to create a new line.

   Press your up arrow key.

   Copy the entirety of your machine's `id_rsa.pub` file that was printed out earlier.

Press the keys `Ctrl`, `Shift`, `v` in order to paste your key into the `~/.ssh/authorized_keys` file.

Press the key `Esc`. Then type `:wq` and press your `enter` button to close the file.

Now you should be able to SSH into the Control PC without typing the password.

7. If you're using the VM, skip this step.

   We also need to add your computer's hostname to the Control PC's `/etc/hosts` file. Use the previous instructions in order to modify the file using vim or your own text editor.

   SSH into the Control PC with the `student` account and add the following line to the Control PC's `/etc/hosts` file:

   ```
   192.168.1.3          <your computer name>
   ```

6.2 Learning more

The github for this project can be found at
https://github.com/iamlab-cmu
https://github.com/iamlab-cmu/iam-interface