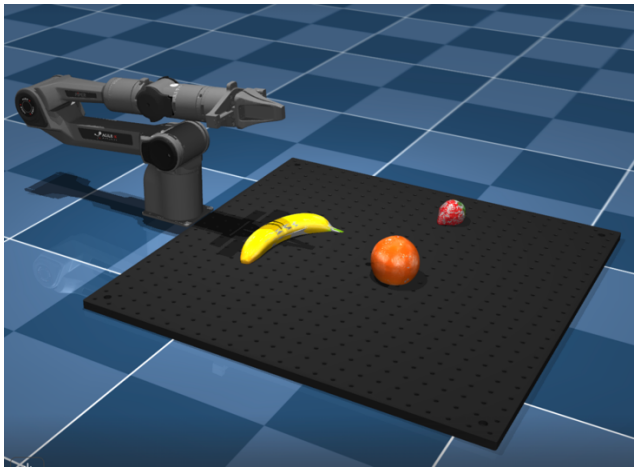
MINI PROJECT 0: SIMULATOR SETUP

Due: Wednesday, September 3 11:59pm EST

The objective of MP 0 is to setup the simulator environment for the robotic manipulator. In this course, we will use MuJoCo to simulate the AgileX Piper robotic arm.



A. Linux Environment Setup

The majority of today's robotics research and industry development happens in Linux (especially Ubuntu) environments. Some of the reasons for this include but not are not limited to:

- Strong support for open-source tools and libraries.
- Easier access to low-level hardware, e.g, serial ports, CAN buses, and custom drivers.
- Large developer and research community
- Frameworks such as ROS (Robot Operating System), Gazebo, MoveIt, and many hardware drivers are designed primarily for Linux.

Accordingly, to become a competent roboticist, you need to get familiar with working in a Linux environment. The hardware drivers and SDK (Software Development Kit) for AgileX Piper robotic arm that we will utilize in the lab are also designed for Linux. Therefore, the desktop computer connected to the robotic arm at each station runs Linux (Ubuntu). To minimize discrepancies between working in a simulator environment and at a real robot arm station, we strongly encourage you to setup a Linux environment on your computer.

If you currently have not setup a Linux environment on your computer, please choose from the two options below.

Note: Some instructions may refer to other versions of Ubuntu. However, we strongly recommend

that students use either Ubuntu 22.04 or 24.04.

- Windows Users:
    1. Dual Boot
    2. WSL (Windows Subsystem for Linux)
    3. VM (Virtual Machine) – Virtual Box
       Note: Running MuJoCo on a VM can be very slow
- Mac Users:
  Currently, there is no consistent and stable way to install Linux on a Mac. Ubuntu does not officially support native install for Macs with Apple M-series chips (ARM64 architecture). Users of older Macs with Intel chips (x86_64 architecture) can setup dual boot, but this may be somewhat difficult.
    1. Keep using Mac OS (seems to be the best option for now, but may face complications in the future)
    2. VM (Virtual Machine) – Virtual Box
       Note: Running MuJoCo on a VM can be very slow
    3. Dual Boot (Intel Macs only)

B. Docker Installation

Note: Mac users can skip this step. They can still install Docker on their machine, but they will not be able to see GUI from the container. Please follow instructions at the end of Section C. for local MuJoCo installation.

In this class, we will also heavily rely on Docker. Docker is a software platform that lets you package an application and all of its dependencies (OS, libraries, configuration, environment, etc) in a lightweight unit called a container. A container is just like a complete snapshot of a working instance of an application. The main advantages of using Docker are:

- Containerized software runs consistently across different environments, avoiding "it works on my machine" problem. If a container runs on my machine, it will run on your machine as well, if Docker is installed.
- Containers are lightweight. Unlike a virtual machine, containers share the host machine's operating system kernel. Therefore, they are much lighter and faster to run.

Note: Docker is an instance of a container engine / runtime. There are alternatives including Podman and containerd, but Docker is the most popular one.

Note: The instructions below are terminal commands for a Linux (Ubuntu) system.

- Installing Docker:

  # Docker installation using the convenience script

```
curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh
```

Add user to the docker group

```
sudo usermod -aG docker $USER
```

Note: This allows you to run docker command without 'sudo.' However, this will not take effect until you restar.

Verify Docker

```
docker run hello-world
```

- Installing Nvidia container toolkit (only for computers with Nvidia GPU for faster simulation):

Configure the repository

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg
--dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \

    && curl -s -L https://nvidia.github.io/libnvidia-
container/stable/deb/nvidia-container-toolkit.list | \

    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-
container-toolkit-keyring.gpg] https://#g' | \

    sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list \

    && \

    sudo apt-get update
```

Install the NVIDIA Container Toolkit packages

```
sudo apt-get install -y nvidia-container-toolkit
```

```
sudo systemctl restart docker
```

Configure the container runtime

```
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

Verify NVIDIA Container Toolkit

```
docker run --rm --runtime=nvidia --gpus all ubuntu nvidia-smi
```

C. MuJoCo Installation:

In this course, we provide a preconfigured docker container installed with MuJoCo 3.3.5.

Clone the class repo to your desired directory

```
git clone https://github.gatech.edu/CS4803ARM/CS4803ARM_Fall2025.git
```

Download and unzip MP0.zip from Canvas and place it under

```
CS4803ARM_Fall2025/user_data
```

Note: Any files placed in this directory will be accessible from the Docker container

Change the directory

```
cd CS4803ARM_Fall2025/docker
```

Important: You will not see GUI if you don't run this command before running the shell script.

```
xhost +
```

Run the convenience script to run the container

```
./docker_run.sh
```

or

```
./docker_run_gpu.sh
```

(If your computer has an Nvidia GPU and you have installed Nvidia container toolkit. MuJoCo will be much smoother and faster.)

Change the directory to MP0 inside the Docker container.

```
cd user_data/MP0
```

Test run MuJoCo. This will load an AgileX Piper arm and an optical breadboard. You can control each joint by clicking/dragging each bar under the 'Control' tab on the top right.

```
python3 -m mujoco.viewer -mjcf=./scene.xml
```

Test run a Python demo. After a few seconds, a Piper arm will follow a pre-defined trajectory.

```
python3 test.py
```

Local MuJoCo installation for Mac users:

Open 'Ternimal' app

Install dependencies. Please make sure you have Python 3.10

```
brew install python@3.10 glfw
```

Install MuJoCo

```
pip3 install mujoco
```

Clone the class repo into your desired directory

```
git clone https://github.gatech.edu/CS4803ARM/CS4803ARM_Fall2025.git
```

Download and unzip MP0.zip from Canvas and place it under

```
CS4803ARM_Fall2025/user_data
```

Change the directory to the MP0 folder

```
cd CS4803ARM_Fall2025/user_data/MP0
```

Test run MuJoCo. This will load an AgileX Piper arm and an optical breadboard. You can control each joint by clicking/dragging each bar under the 'Control' tab on the top right.

```
python3 -m mujoco.viewer -mjcf=./scene.xml
```

Run a Python test file. After a few seconds, a Piper arm will follow a pre-defined trajectory.

```
mjpython test.py
```