Robotics Control Instruction

Jungjae Lee1

1 Boston University College of Engineering

Table of Contents

[**1. Download & Installation** 2](#_Toc139275278)

[**2. MuJoCo** 3](#_Toc139275279)

[2.1 Installation 3](#_Toc139275280)

[2.2 Notes 3](#_Toc139275281)

[2.2.1 mujoco-py (Does Not Support) 3](#_Toc139275282)

[2.2.2 MuJoCo Gym (Does Not Support) 3](#_Toc139275283)

[2.2.3 MuJoCo Python Bindings 4](#_Toc139275284)

[2.2.4 DeepMind Control Software 4](#_Toc139275285)

[2.2.5 Gymnasium 4](#_Toc139275286)

[**3. Gazebo** 4](#_Toc139275287)

[**4. ROS 2 (Humble) with moveit2 package** 4](#_Toc139275288)

[**5. Errors** 7](#_Toc139275289)

[5.1 Error 4.2.1 7](#_Toc139275290)

[5.2 Error 4.3.1 8](#_Toc139275291)

# **1. Download & Installation**

* Download Visual Studio Code (<https://code.visualstudio.com/>)
* Install Python (<https://www.python.org/downloads/>)

Install Python

python –version

// Check Python’s version as most ubuntu come with Python pre-installed

Continue if Python is not installed

sudo apt update

sudo apt install software-properties-common

// Allows you to easily manage your distribution and independent software sources

sudo add-apt-repository ppa:deadsnakes/ppa

// Allows you to install multiple Python versions on Ubuntu System

sudo apt update

sudo apt install python3.11.4

python –version

Source: <https://phoenixnap.com/kb/how-to-install-python-3-ubuntu>

* Install pip (<https://pip.pypa.io/en/stable/installation/>)

sudo apt install python3-pip

# **2. MuJoCo**

## 2.1 Installation

* Install MuJoCo (<https://github.com/deepmind/mujoco/releases>)

pip install mujoco==2.3.6

* Install mediapy
* Install UR5e (On email)

## 2.2 Notes

### 2.2.1 mujoco-py (Not Supported)

* Mujoco-py allows using MuJoCo from Python 3, maintained by the OpenAI Robotics team.
* mujoco-py does not support any version after MuJoCo 2.1.0, according to GitHub 1. Instead, you can use [MuJoCo Python Bindings 2](#_2.2.3_MuJoCo_Python) or [DeepMind Control Software 3](#_2.2.4_DeepMind_Control).
* Install mujoco-py

Install mujoco-py

pip install mujoco-py

Source: <https://pypi.org/project/mujoco-py/>

* Sources:
  + 1 “mujoco-py” (GitHub, 2022): <https://github.com/openai/mujoco-py>
  + 2 “MuJoCo Python Bindings” (GitHub, 2023): <https://github.com/deepmind/mujoco/blob/main/python/README.md>
  + 3 “DeepMind Control Software” (GitHub, 2023): <https://github.com/deepmind/dm_control>
  + Installing & Using MuJoCo 2.1.5 with OpenAI Gym (Reddit, 2022): <https://www.reddit.com/r/reinforcementlearning/comments/usaigw/installing_using_mujoco_215_with_openai_gym/>

### 2.2.2 MuJoCo Gym (Not Supported)

* MuJoCo Gym is a standard API for reinforcement learning, and a diverse collection of reference environments. 1
* The newest OpenAI gym does not work with MuJoCo 2.0, which you will have to install MuJoCo 1.50 binaries. Alternatively, if you need to use MuJoCo 2.0, you can download the MuJoCo 2.0 binaries and install the newest mujoco-py. Then, you can install the latest Gym that supports MuJoCo 2.0 using the command below. 2

Install MuJoCo Gym

pip install -U gym[all]==0.15.3

Source: <https://neptune.ai/blog/installing-mujoco-to-work-with-openai-gym-environments>

* DeepMind does not support MuJoCo Gym anymore. Alternatively, you can use [Gymnasium 3](#_2.2.5_Gymnasium).
* Sources:
  + 1 “MuJoCo” (Gym Documentation): <https://www.gymlibrary.dev/environments/mujoco/index.html>
  + 2 “Installing MuJoCo to Work With OpenAI Gym Environments” (Neptune.ai, 2023): <https://neptune.ai/blog/installing-mujoco-to-work-with-openai-gym-environments>
  + 3 “Gymnasium” (Farama Foundation - Gymnasium Documentation): <https://gymnasium.farama.org/>

### 2.2.3 MuJoCo Python Bindings

### 2.2.4 DeepMind Control Software

### 2.2.5 Gymnasium

* Gymnasium is a standard API for reinforcement learning, and a diverse collection of reference environments, maintained fork of OpenAI’s Gym library. 1
* Install Gymnasium

Install Gymnasium

pip install gymnasium[classic-control]

Source: <https://gymnasium.farama.org/environments/classic_control/>

* Source:
  + 1 “Gymnasium” (Farama Foundation - Gymnasium Documentation): <https://gymnasium.farama.org/>

# **3. Gazebo**

* Install Gazebo Sim 7 (Ubuntu Software)

# **4. ROS 2 (Humble) with moveit2 package**

* Install ROS 2 (Humble)

Install ROS 2 (Humble)

locale // Check for UTF-8

Continue if it is not set as UTF-8. If it is already set, skip to the next step

sudo apt update && sudo apt install locales

sudo locale-gen en\_US en\_US.UTF-8

sudo update-locale LC\_ALL=en\_US.UTF-8 LANG=en\_US.UTF-8

export LANG=en\_US.UTF-8

locale // Verify the setting

-----

sudo apt install software-properties-common

sudo add-apt-repository universe

sudo apt update && sudo apt install curl -y

sudo curl -sSL [https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg](https://raw.githubusercontent.com/ros/rosdistro/master/ros.key%20-o%20/usr/share/keyrings/ros-archive-keyring.gpg)

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null

sudo apt update

sudo apt upgrade

sudo apt install ros-humble-desktop

source /opt/ros/humble/setup.bash

To run ROS, you have to run “source /opt/ros/humble/setup.bash” to configurate the environment. Instead, you can do this to skip this step

gedit ~/.bashrc // This would open up the “.bashrc” file

source /opt/ros/humble/setup.bash

// Type this code at the very bottom of the file, which most likely will be on line 119

Source: <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>

* Install MoveIt 2 Source

sudo apt update

sudo apt dist-upgrade

rosdep update

sudo apt install -y \

build-essential \

cmake \

git \

libbullet-dev \

python3-colcon-common-extensions \

python3-flake8 \

python3-pip \

python3-pytest-cov \

python3-rosdep \

python3-setuptools \

python3-vcstool \

wget && \

python3 -m pip install -U \

argcomplete \

flake8-blind-except \

flake8-builtins \

flake8-class-newline \

flake8-comprehensions \

flake8-deprecated \

flake8-docstrings \

flake8-import-order \

flake8-quotes \

pytest-repeat \

pytest-rerunfailures \

pytest

sudo apt remove ros-$ROS\_DISTRO-moveit\*

// remove any pre-existing MoveIt debians

export COLCON\_WS=~/ws\_moveit2/

mkdir -p $COLCON\_WS/src

cd $COLCON\_WS/src

git clone https://github.com/ros-planning/moveit2.git -b $ROS\_DISTRO

for repo in moveit2/moveit2.repos $(f="moveit2/moveit2\_$ROS\_DISTRO.repos"; test -r $f && echo $f); do vcs import < "$repo"; done

rosdep install -r --from-paths . --ignore-src --rosdistro $ROS\_DISTRO -y

sudo apt install ros-$ROS\_DISTRO-rmw-cyclonedds-cpp

export RMW\_IMPLEMENTATION=rmw\_cyclonedds\_cpp

cd $COLCON\_WS

colcon build --event-handlers desktop\_notification- status- --cmake-args -DCMAKE\_BUILD\_TYPE=Release

[<ERROR 4.2.1>](#_Error_4.2.1)

source $COLCON\_WS/install/setup.bash

Source: <https://moveit.ros.org/install-moveit2/source/>

* Install Colcon

sudo apt install python3-colcon-common-extensions

sudo apt install python3-colcon-mixin

colcon mixin add default <https://raw.githubusercontent.com/colcon/colcon-mixin-repository/master/index.yaml>

colcon mixin update default

sudo apt install python3-vcstool // install vcstool

mkdir -p ~/ws\_moveit/src

cd ~/ws\_moveit/src

git clone [https://github.com/ros-planning/moveit2\_tutorials -b main --depth 1](https://github.com/ros-planning/moveit2_tutorials%20-b%20main%20--depth%201)

vcs import < moveit2\_tutorials/moveit2\_tutorials.repos

sudo apt update && rosdep install -r --from-paths . --ignore-src --rosdistro $ROS\_DISTRO -y

cd ~/ws\_moveit

colcon build --mixin release

[<ERROR 4.3.1>](#_Error_4.3.1)

# **5. Errors**

## 5.1 Error 4.2.1

* Error Message:

--- stderr: moveit\_setup\_controllers

c++: fatal error: Killed signal terminated program cc1plus

compilation terminated.

gmake[2]: \*\*\* [CMakeFiles/moveit\_setup\_controllers.dir/build.make:298: CMakeFiles/moveit\_setup\_controllers.dir/include/moveit\_setup\_controllers/moc\_urdf\_modifications\_widget.cpp.o] Error 1

gmake[2]: \*\*\* Waiting for unfinished jobs....

gmake[1]: \*\*\* [CMakeFiles/Makefile2:161: CMakeFiles/moveit\_setup\_controllers.dir/all] Error 2

gmake: \*\*\* [Makefile:146: all] Error 2

---

Failed   <<< moveit\_setup\_controllers [3min 43s, exited with code 2]

Aborted  <<< moveit\_setup\_core\_plugins [3min 43s]

Aborted  <<< moveit\_setup\_app\_plugins [3min 48s]

Aborted  <<< moveit\_setup\_srdf\_plugins [3min 48s]

Summary: 48 packages finished [12min 27s]

1 package failed: moveit\_setup\_controllers

3 packages aborted: moveit\_setup\_app\_plugins moveit\_setup\_core\_plugins moveit\_setup\_srdf\_plugins

5 packages had stderr output: controller\_manager moveit\_configs\_utils moveit\_setup\_controllers moveit\_setup\_srdf\_plugins ros2controlcli

1. packages not processed

* Solution:
  + Just re-run the code: “colcon build --event-handlers desktop\_notification- status- --cmake-args -DCMAKE\_BUILD\_TYPE=Release”

## 5.2 Error 4.3.1

* Error Message:

--- stderr: moveit\_configs\_utils

/usr/lib/python3/dist-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.py install is deprecated. Use build and pip and other standards-based tools.

  warnings.warn(

---

* Solution:

python3 // check the python version

import setuptools

print(setuptools.\_\_version\_\_) // check the setuptools version

if the setup tools version is above 58.2.0, follow the instructions below

pip install setuptools==58.2.0 // my version was 59.6.0 (updated to 68.0.0 due to “DeprecationWarning: The distutils package is deprecated and slated for removal in Python 3.12. Use setuptools or check PEP 632 for potential alternatives”)

Source: <https://answers.ros.org/question/396439/setuptoolsdeprecationwarning-setuppy-install-is-deprecated-use-build-and-pip-and-other-standards-based-tools/>