

2.12: Introduction to Robotics

Lab 1:

Software Installation*

Spring 2023

Assigned on: 9th February 2023 Due by: 15th February 2023

Instructions:

1. **You will be working on this installation alone for your individual computer, but if you face any issues, contact a TA. Feel free to work alongside a fellow classmate, but turn in your own work.**
2. **Check off with TA or submit a PDF of the collection of screenshots to Canvas.**

1 Introduction

In this lab you will install several different software packages that will be used for the remainder of the semester.

1. MATLAB 2022a: Make sure you use the latest build of MATLAB to ensure that all the packages we will be using for this class will work. It is recommended that you get the full installation of MATLAB (with all the packages) from MIT.
2. Required Toolboxes from Mathworks: In addition to base MATLAB, we'll be using a number of add-ons through this class. This includes a robotics toolbox that'll handle a lot of the lower level details for future robotics simulations.
3. Python 3.9: You may be able to use other versions of Python 3, however there will likely be some syntax issues when using Python 2.
4. Pip: Python's package installer. You'll use this for all the packages that you don't have on your machine that we use.
5. OpenCV: An open source computer vision package that contains a lot of image processing tools.

*

1. Version 1 - 2020: Jerry Ng, Rachel Hoffman, Steven Yeung, Kamal Youcef-Toumi
2. Version 2 - 2021: Cormac O'Neill
3. Version 3 - 2023: Cormac O'Neill
4. Version 4 - 2023: Ravi Tejawani

2 Matlab and Required Toolboxes (fresh install)

You likely already know this, but as an MIT student, you have access to MATLAB. We ask that you install the newest version as there have been updates to the package we'll be using for the class since going to the 2022a update.

2.1 MATLAB download link

Specifically, you'll be going to the following website: <https://ist.mit.edu/matlab/all> if you need the download. You'll also need to install Simulink as the following toolbox may require it for some exercises. Note that these downloads can take a significant amount of time to complete (upwards of 2 hours), so start it early!

We recommend that you perform a full installation of MATLAB (install all of the available toolboxes), however here is a list of toolboxes that you must have for this class:

1. Optimization toolbox
2. Robotics System toolbox
3. Control System toolbox
4. Computer Vision toolbox
5. Deep Learning toolbox
6. Stateflow
7. Simulink
8. ROS toolbox
9. Navigation toolbox
10. Image Processing toolbox
11. Image Acquisition toolbox
12. Signal Processing toolbox

2.2 Required Toolboxes

Assuming you already have installed MATLAB on your computer, you can install the required toolboxes if you didn't do so during the main installation process.

You can install the required toolboxes in the typical way, by opening MATLAB, navigating to the "Home" tab and clicking on the add-ons button.

Regardless of how you installed the toolboxes, verify that you at the very least have successfully installed the Required Toolboxes listed above by clicking on the add-ons button, and then selecting manage toolboxes to see a list.

After installing the toolboxes, we'd like you to run through the example listed here: <https://www.mathworks.com/help/robotics/ug/2d-inverse-kinematics-example.html>

After going through the example, take a screenshot of the robot completing the circular trajectory.

3 Python and Python packages

3.1 Python

We ask that you install Python 3.9 on your home operating system (approx. 120MB).

You can find Python 3.9 here: <https://www.python.org/downloads/release/python-3910/>

IF YOU HAVE A VERSION OF PYTHON 2 INSTALLED ALREADY, it's okay.

Usually the installation just makes it so that the default terminal command

```
python file.py #Example of a command that could be run
```

will use that previously installed version.

To use a different version of python, you can type

```
python3 file.py #Example
```

and it will use version 3.9.

You can check which version of python you have installed by running the command:

```
python -V
```

3.2 PIP

Pip is a python package installer. You can find the link for installation here: <https://pip.pypa.io/en/stable/installation/> Or alternatively install pip from the terminal.

You'll need to run the following commands in the terminal:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

To make sure that the installation completed successfully, run the following in the terminal:

```
pip help
```

and you will see a list of functions that can be run using pip.

If you already have pip installed, please make sure that it is up to date by following the instructions here:

<https://pip.pypa.io/en/stable/installation/#upgrading-pip>

3.3 OpenCV

OpenCV is an open source computer vision library that we'll be using for the rest of the semester. To install OpenCV, we'll be using pip. Run the following command in the terminal to do so:

```
pip install opencv-python
```

3.4 Git

The last thing you'll install to your computer is Git. Git helps you manage and keep track of your source code history. You may have heard of GitHub, a cloud-based service that allows you to manage Git repositories. To install Git locally on your computer, go to the following page and follow the instructions: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

There are different installation instructions depending on OS, so make sure to read it carefully.

3.5 Testing

To complete the lab, you'll need to test if all these tools work. Run the following commands to clone from the github:

```
git clone https://github.com/mit212/lab1_2023
```

This should have cloned a folder from 2.12's Git repository on GitHub, to a folder on your own computer.

3.5.1 Test 1: Webcam

The first script you will run is using OpenCV to access your webcam. To do this, run the command:

```
cd lab1_2023/lab1_2023-master/scripts
python webcamtest.py
```

This **SHOULD** pull up two windows of webcam footage. Take a screenshot of these for turning in later.

3.5.2 Test 2: Image Processing

The second script you'll run is using OpenCV to process an image. To do this, run the command:

```
python process.py
```

Similar to before, this **SHOULD** pull up four windows of images. Take a screenshot of these for turning in as well.

4 Turn in items

Upon completion of this lab, turn in the screenshots you took throughout the process - or show your screen to a TA in person. There should be a total of three different screen shots:

1. Section 2.2: Robot trajectory motion in a circle
2. Section 3.5.1: Webcam Test
3. Section 3.5.2: Image Processing Test

On a separate note, we recommend that you install some text editor in the future, as it will make your life easier in future labs. We personally recommend Sublime Text or Visual Studio. You can find the install links for those here: <https://www.sublimetext.com/> and <https://visualstudio.microsoft.com/>