MAE 145: Robot Planning and Estimation

A Quick Guide to Python, anaconda and IDEs: In this course, we will be using Python to write programs. If you are not familiar with Python but you have programming experience (e.g. MATLAB), it should be easy to get up to speed with it. Please spend time to get yourself familiar with python in the first two weeks of the course. Here is some information on how to work with it and install it.

Python. Python is an open source language, which allows for intuitive, fast writing. Apart from being free, the main strength of Python is that it has a very large user community, who offer free support and develop useful packages for many scientific areas. Because of this, we will be using Python in this course.

(a) A brief tutorial on Python, by David J. Pine, and its use in Science is available in `https://physics.nyu.edu/pine/pymanual/html/pymanMaster.html`

(b) Learning about Python. One can find many tutorials and materials online to learn about it. For example:

- Apart from the tutorial above, you can check out the Python tutorial from Python.org `https://docs.python.org/3.3/tutorial/`.
- A source to learn about Python for someone without programming experience is `https://runestone.academy/ns/books/published/thinkcspy/index.html`, based on the book "How to think like a computer scientist" by Allen B. Downey.
- Links to all documentation are available in the Anaconda Navigator (learning tab), as well as access to forums (click on the community tab) such as Stack Overflow: Python. When in doubt about Python, you can look for similar questions others had in this forum.
- In particular, documentation on python and scientific computing is available here `http://scipy-cookbook.readthedocs.io/` and, on matplotlib, `matplotlib.org`

(c) Although not required for this course, you can also take some time to learn about command lines for terminals, as in Linux and MacOS. A game you can play to do this is "Terminus" developed by MIT students. For more info, check out the file `http://www.mprat.org/projects/terminus/`

Anaconda. (This part is optional) Anaconda is a tool that allows you to manage different virtual Python environments, access to a collection of popular packages related to data science, as well as IDEs (see below). Anaconda can be installed in different OS (windows, mac, and linux). To install anaconda just google "anaconda installation", and you will get directed to it. For this course, you do not need to install the full anaconda, but just miniconda. You will be able to access anaconda through the command line or via the anaconda navigator.

Your Python programming projects may require different libraries, packages or modules. You do not have to install all the packages in one computer, and moreover, sometimes you may need different, conflicting versions of non-compatible packages for different projects. For example, old packages may require a python 2.7, while newer data science modules may require a python 3.6. Anaconda offers you the possibility of having non-compatible environments in the same computer, making your life easier. Once anaconda is installed in your computer, you can set up a virtual environment through the command line, or, even more easily, through the anaconda navigator.

To create a new environment using the command line, just type `conda create --name project1`, then choose the [y] option. In addition, you need to activate your environment `conda activate project1`. It will show in the command line as (project1) and the environment should have python in it. Under this environment, you can install packages with `conda install`. For example, `conda install pandas` will install `pandas`, a popular module to handle data. To exit the enviroment type `conda deactivate project1`. To create an environment with python3, we can just type `conda create -name env2 python=3.7`. You will find these environments in your home directory, under the `anaconda3>envs` folder. To remove an environment, just type `conda env remove -n project1`.

The anaconda navigator allows you to do the above operations as well as install python packages via an interface that looks like a browser. Just look for and click on “Environments”, find icons to “Create”, “Clone”, “Import”, or “Remove”, these environments.

IDEs: An IDE or Integrated Development Environment, is a collection of tools and features that allows you to write, and debug Python programs. It usually includes a graphical user interface that is similar to that of Matlab’s and also a few commands that allows you to debug a program following shorthand commands. Among the most popular IDEs are “Spyder,” “PyCharm”, or “VS Code”. You can either install one of the above to work in this course. For example, Spyder, already comes with the Anaconda navigator. For PyCharm google “download PyCharm” and will get to the official PyCharm website (at jetbrains.com, similarly for VS Code. Download the community version, which is free.) For each of the IDE above, there are quick tutorials that you can follow (please take a moment to do so). Finally, another free option is to use the Emacs or Vim classic text editors to work with Python. You can also work with Python via jupyter (which is a web-based notebook environment), or with JupyterLab (based on jupyter).