Advanced Model Predictive Control Fall 2023
Prof. M. Zeilinger
Dr. A. Carron
Dr. L. Hewing
Dr. J. Köhler

Python Setup

Alexandre Didier, Jérôme Sieber, and Rahel Rickenbach

1 Setup of Development Environment

1.1 Download Project and Check Python Version

As a first step, download the AMPyC project folder from Moodle. You can find it in the "Introduction" tile.

To work on the programming exercises, Python 3.8 needs to be installed on your computer. If you are working with Linux or Mac, you can check whether you already have a version of Python 3.8 installed by running

```
python3.8 --version
```

in your terminal. Or you run

```
python3 --version
```

in your powershell/cmd, when working with Windows. If you have a current installation of Python 3.8, the command will output its version, e.g.,

```
Python 3.8.18
```

Any version of Python 3.8 should be fine to work with. If there is no current installation shown, you have to install it for your operating system of choice.

1.2 Create and Activate Virtual Environment with Venv

Disclaimer *Please refrain from using Anaconda or one of its derivatives to install the virtual environment. Its usage typically creates unnecessary overhead and there are some known issues for active module development associated with Anaconda. If you decide to use Anaconda or one of its derivatives anyway, we won't provide any technical support.*

To ensure all students have the same development setup, you will test and run your programming exercises in a virtual environment. Hence, if not already done so in the past, we install the virtual environment package by running:

```
pip install virtualenv
```

Then, move into your AMPyC project folder with

```
cd ampyc
```

and create your virtual environment in Python 3.8. To increase clarity, we will call it "ampyc_env". To do so, run

```
python3.8 -m venv ampyc_env
```

Inspecting your AMPyC project, you should now find a new folder named "ampyc_env". If you use either Mac or Linux, the virtual environment can be activated using the following command,

```
source ampyc_env/bin/activate
```

On Windows the activation is done by running

```
ampyc_env/Scripts/activate.bat //In CMD
ampyc_env/Scripts/Activate.ps1 //In Powershell
```

After successful activation, your terminal path should contain the prefix "ampyc_env", which indicates that you are working in the specific virtual environment.
Activating your virtual environment for the first time it will be nearly empty. Hence, to test whether your installation and activation was successful, you can run

```
python -m pip list
```

which should only output the packages pip and setuptools.

1.3 Install Requirements

Before installing all requirements, we need to update pip using the following command

```
python -m pip install --upgrade pip
```

In the AMPyC project folder, we provide you with a requirements file (`requirements.txt`), which contains all the necessary packages. You can install them in your virtual environment, by running

```
python -m pip install -r requirements.txt
```

This might take a few seconds.

1.4 Install AMPyC Package

Next, we need to install the AMPyC package on which you will work in all programming exercises. You can do this by running

```
python -m pip install -e .
```

Please note that we used the flag "`-e .`". This is important, since it allows to modify the package on the fly, while working on the programming exercises!
Listing the installed packages with `python -m pip list` again, should now also show the AMPyC package.

1.5 Work on Programming Exercises

You can open and work on the source code with your editor of choice. The main files however are written as notebooks and are easiest to run in a jupyter notebook environment. You can open it with the command

```
python −m jupyter notebook
```

Alternatively, you can use VS Code to work on the Python scripts and the notebooks simultaneously. For more details, please visit https://code.visualstudio.com/docs/datascience/jupyter-notebooks.

1.6 Test your Installation

In the notebooks folder you find an implementation of a standard linear MPC controller. Run this notebook to verify your installation. If you encounter any errors when running this notebook, your installation is corrupted.

1.7 Deactivation of Virtual Environment

To deactivate your virtual environment, run

```
cd ampyc_env/bin
deactivate
```

This should remove the "ampyc_env" prefix of your terminal path. For more information on the usage of venv, you can visit https://docs.python.org/3/library/venv.html.