6ELEN018W - Tutorial 3 Exercises

Reminder: Setting up the Python Robotics Toolbox on your Personal Computer

To set up the Robotics toolbox on your personal machine you must have a working version of Python installed.

Once you have this run either of the following commands (the second is only for Anaconda installations of Python). You can run these inside a Python virtual environment (the way you create and activate this, depends on the operating system of your machine) or the main installation of Python on your computer:

- `pip install rvc3python`
- `conda install rvc3python`

The above installation provides a command line tool `rvctool` which can be started executing its name. The tool starts a Python command line and it automatically imports all necessary modules for the toolbox.

Alternatively, you can start your own Python command line or IDE (e.g. JupyterLab) and import all the necessary modules before using the toolbox:

```
import math
import numpy as np
from scipy import linalg, optimize
import matplotlib.pyplot as plt
from spatialmath import *
from spatialmath.base import *
from spatialmath.base import sym
from spatialgeometry import *
from roboticstoolbox import *
from machinevisiontoolbox import *
import machinevisiontoolbox.base as mvb
```

You will need to import these modules in every program you write.

Exercise 1

Using the Python Robotics toolbox compute the homogeneous transformation which calculates the pose of the end-effector for a robot manipulator having 4 revolute joints.

The calculation should be done in symbolic form. The joints angles are q_1, q_2, q_3, q_4 and the lengths of the corresponding links are a_1, a_2, a_3, a_4.

Exercise 2

Consider the problem of a 2-joint robot arm discussed in the lecture.

Write a Python function which accepts 4 arguments, the two joints angles and the lengths of the 2 links.

1. Implement your function so that it returns the homogeneous transformation which describes the end-effector pose without using the Python Robotics toolbox .
2. Call the appropriate Python Robotics toolbox to do the equivalent computations and check that your two implementations give the same results.

Exercise 3

Consider the problem of a 3-joint robot arm discussed in the lecture.

Write a Python function which accepts 6 arguments, the two joints angles and the lengths of the 3 links.

Your function should return the Jacobian matrix which can be used for the calculation of the velocity of the pose of the end-effector. Your implementation should use the functions included in the Python Robotics toolbox.

Exercise 4

Using Matlab and the live script demonstrated to you during the lecture and found in the following URL:

`https://dracopd.users.ecs.westminster.ac.uk/DOCUM/courses/6elen018w/lecture3_arm_demos.mlx`

1. You need to install some additional files for the Matlab Robotics toolbox: Download from `https://github.com/petercorke/RVC3-MATLAB` the zip file and unzip it in a location where you can use it (in the university labs this should be done in your `H:` drive).

   The `toolbox` directory in the directory you unzipped the above should be included in the Matlab path:

   - In the university labs and from inside Matlab execute `path(path,'H:\RVC3-MATLAB\toolbox')` assuming that `H:\RVC3-MATLAB` contains the extracted contents of the zip file. You have to execute this every time you restart Matlab in the labs.
   - In your personal computer where you have admin rights, you can use the `pathtool` command (from inside Matlab) to add the additional directory to the Matlab path permanently.
2. Run the script, section by section and try to understand every single line of Matlab code. Ask your seminar tutor if in any doubt.
3. Add a new section to the live script for demonstrating a 4-joint robot arm. The robot arm should have link lengths a_1, a_2, a_3, a_4 and joints angles $q_1 = \frac{\pi}{2}, q_2 = \frac{\pi}{4}, q_3 = \pi, q_4 = \frac{\pi}{8}$.