5ELEN018W - Tutorial 3 Exercises

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 *
```

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

Exercise 1

1. Write a Python function which accepts a single argument representing an angle θ in radians and returns the corresponding 2D rotation matrix by calling the appropriate function in the Python Robotics toolbox.
2. What is the type of the returned matrix (use the `type` Python function to figure it out)

3. Extend your function so that it accepts a second argument which is a string. If the string is `'deg'` it calculates the rotation matrix in degrees, otherwise if the string is `'rad'` it calculates the rotation matrix in radians. For any other value of the second argument the function returns an error.

Exercise 2

Study the documentation of `numpy` arrays from the following link:

`https://numpy.org/doc/stable/reference/generated/numpy.array.html`

Create a 1-dimensional numpy array from a Python list. Create two different 2-dimensional numpy arrays and multiply them together. Confirm that the result of the matrix multiplication is correct by doing the multiplication manually as well.

Exercise 3

Write a Python function which accepts 2 arguments corresponding to angles (in degrees) and plots the reference frames after applying rotation with respect to the given angles. The first frame should be in blue colour and the second in red.

Hint: Use the `trplot2` function of the toolbox.

Exercise 4

Without using the Robotics Toolbox write a Python function which accepts 2 arguments, an angle θ and a list $(t_x t_y)$ corresponding to a translation vector. The function should return the 3×3 homogeneous transformation for that angle and translation as a `numpy` array.

Exercise 5

A frame B is created by rotating a 3D frame A about the z axis by 70 degrees and translating frame A by 20 units in axis x and 10 units in axis y.

1. Write some Python code which returns a `numpy` array corresponding to the homogeneous transformation describing the above.
2. A point P has the following coordinates with respect to frame B: (3.1, 8, 2). Calculate the coordinates of point P with respect to frame A using the homogeneous transformation you calculated previously.

   Hint: Use the `math.radians` function to convert degrees to radians (see `help(math.radians)`).