

METR4202 / METR7202 -- Robotics

Tutorial 1 – Week 1: Robotics Toolbox

The objective of this tutorial is to familiarize yourself with the MATLAB robotics toolbox developed by Peter Corke as well as the OpenCV software for computer vision.

1. For a team of 5 (i.e., four others and yourself). We may allow a few teams of 6.

2. Please visit and download:

- MATLAB robotics toolbox:

<http://robotics.itee.uq.edu.au/~metr4202/software/index.html>

Cached from: http://petercorke.com/Robotics_Toolbox.html

- OpenCV (Downloads are listed on the 4th section of the page):

<http://robotics.itee.uq.edu.au/~metr4202/software/index.html>

Cached from: <http://opencv.org/>

(Both cached on the METR4202 page)

3. To install the Matlab Robotics Toolbox, unzip the contents (the folder rvctools) to a the MATLAB toolbox folder (e.g., D:\MATLAB\toolbox\rvctools).

4. Then restart Matlab

5. Then try the command `startup_rvc`

6. Play with the `rpy2tr` and `rpy2r` functions.

```
doc rpy2tr
```

```
doc rpy2r
```

After this, try playing with the `rtbdemo`

Representing Position and Orientation

7. We consider a 2D world reference frame denoted by $\{0\}$ which has its x-axis pointing East and its y-axis pointing North. There is a robot with an attached body-fixed coordinate frame $\{A\}$ whose origin is in the centre of the robot, and whose x-axis points in the robot's forward direction.

The robot's centre is at a distance of 123m in the x-direction and -74.6m in the y-direction (with respect to the world coordinate frame). The robot is facing SSW.

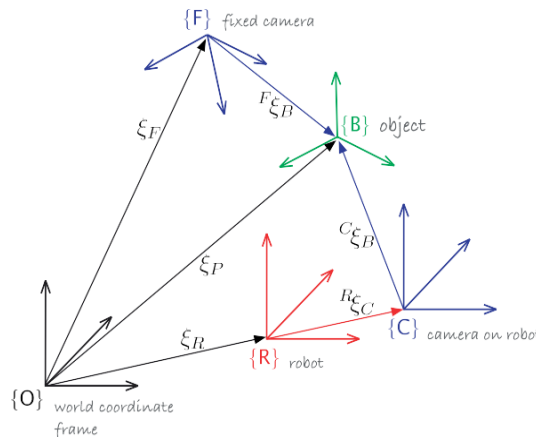
(a) Write a 3x3 matrix homogeneous transformation matrix that expresses the pose of the robot frame $\{A\}$ with respect to the world frame $\{O\}$.

(b) Express the point ${}^0p = (7.2, 11.8)$ in the robots coordinate frame (so find Ap).

8. Five 3D (relative) poses ξ_F , ${}^F\xi_B$, ξ_R , ${}^R\xi_B$ and ${}^R\xi_C$ (see figure below) are given in the file:

<http://robotics.itee.uq.edu.au/~metr4202/exercise8.mat>

Write a matlab script that calculates ${}^C\xi_B$, the relative pose between the camera's coordinate frame $\{C\}$ and the object's coordinate frame $\{B\}$.



9. Now repeat 8, but use a vector-unit quaternion representation of pose. You are not use any existing Matlab tools or functions for handling quaternions. Write everything you need yourself.

The same relative poses as above given as (rotation-axis \mathbf{v} , rotation angle Θ (theta), position \mathbf{t}) can be found in:

<http://robotics.itee.uq.edu.au/~metr4202/exercise9.mat>

(Does the answer here agree with the one you got in 8?)

10. **(optional)** Write a function that calculates the composition of two poses in 2D, when they are given by an (angle, 2-vector) representation (see slide 21 lecture 2).