RVAL 2023 Assignment 1

In this assignment, you are required to finish two task:

1. **Rotation functions**: You will create a simple toolbox `myquaternion` for handling rotations using `numpy`.
2. **Toy forward kinematics**: Given a 2-joint robotic arm, you need to compute the position and orientation of its end effector.

Task A: Rotation Functions

Detailed descriptions of each function can be found in `myquaternion.py`

Basic Operations

1. `normalize()`
2. `multiply()`
3. `conjugate()`
4. `rotate()`
5. `relative_angle()`
6. `interpolate_quaternions()`

Conversion

1. `quaternion_to_matrix()`
2. `matrix_to_quaternion()`
3. `quaternion_to_rotvec()`
4. `rotvec_to_quaternion()`
5. `rotvec_to_matrix()`
6. `matrix_to_rotvec()`

Random Sampling

1. `generate_random_quaternion()`

Files

- `myquaternion.py`: Your rotation processing libray. You need to implement all the functions in it.
- `eval_myquaternion.py`: Scoring script for self-evaluation.
- `eval_data.pkl`: Validation set used by the scoring script.

Task B: Forward kinematics for a toy robot arm

In this task, you are required to implement the forward kinematics function for a two-joint robotic arm. The function takes the joint angles and link lengths as inputs and outputs the position and orientation of the end effector.

Robots

We define a toy robotic arm that has only two joints, each associated with a corresponding link.

- Joint 1: The first joint rotates about an axis perpendicular to the ground, connecting the base to Link1.
- Joint 2: The second joint rotates about an axis perpendicular to Link1, connecting Link1 to Link2.

An illustation of the robot arm can be found in `robot_arm.png`.

Function and file

Please implement the `kinematics_forward` function in the `toy_robot_arm.py` file. The joint angles should be represented in radians, and the link lengths should be represented in meters.

Grading

- The assignment will be evaluated on a held-out test set to check the correctness. If you only manage to achieve part of the objectives, you will receive partial score.
- It is not necessary to import extra libraries. You will also lose points if you use extra libraries like `scipy` and `transform3d` (*i.e.* you need to write the calculations by yourself). Late submission will also lose points.

Turning it in

- The deadline of assignment 1 is October 27, 11:59 p.m.
- Submit `myquaternion.py`, `toy_robot_arm.py` and a (very simple) PDF document with self-evaluation results in a single `.zip` file to the school course website.

Hints

> - We use the (w, x, y, z) convention for quaternions (as in the slides).
> - If you can not pass some test data, you can just check them with the scoring script.
> - Think about the singularity of your functions.
> - If you have questions, please post them to the discussion board on the school course website.