CSE4421/5324: Assignment 2

Burton Ma

Posted: Tue Feb 01, 2011
Due: Before noon on Tue Feb 15, 2011

You can find a picture of the A255 base coordinate frame and end-effector coordinate frame in Chapter 1 of the User Guide. You can find the range of motion and dimensional specifications of the A255 robotic arm in Chapter 2 of the User Guide:

`http://www.cse.yorku.ca/course/4421/doc/a255/A255UserGuide.pdf`

(also under the Documentation page of the course web site).

1. Derive the table of Denavit-Hartenberg parameters for the A255 robot (using inches for the units because this allows you to use the same solution for the A150). Place frame 0 as shown in the figure on page 1 of Chapter 1 of the User Guide. Place frame 3 at the wrist center o_c, and frame 5 on the tool flange (see the figure at the end of this document). Document your work by showing a schematic sketch of the A255 (using cylinders for the joints and single lines for the links) labelled with the DH coordinate frames and parameters.

2. Derive the matrices T_3^0 and T_5^3.

3. Solve the inverse kinematics problem for the wrist; i.e., given T_5^3 solve for the values of θ_4 and θ_5.

4. Solve the inverse kinematics problem for the first three joints given the wrist center $o_c^0 = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^T$; i.e., given T_3^0 and o_c^0 solve for the values of θ_1, θ_2, and θ_3. Try to find all of the possible solutions, and then indicate which set applies to the A255 arm.

5. In Java implement the method with signature `move(double x0, double y0, double z0)` that takes as input a location (expressed in the base frame of the robot); the function should then move the origin of frame 5 to the input location, or output a message indicating that the position is not reachable. You should assume that

$$R_5^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

   For this assignment it is sufficient to modify `ArmSimulator.java`. Consider adding a method that solves the inverse kinematics problem for the arm, rather than putting all of the inverse kinematics code inside of `move`; see the final part of this assignment.

6. In Java implement a method with signature `moveLinear(double x0, double y0, double z0)` that takes as input a location (expressed in the base frame of the robot); the function should then move the origin of frame 5 to the input location *in a straight line from the current position*, or output

a message indicating that the position is not reachable. If the location is reachable, but a straight line path is impossible, then the arm should move along the line connecting the start and finish points as far as possible (within reason—say within an inch or so) before completing the motion using `move`. Assume the same value of R_5^0 as in Part 5.

Hand in paper copies of Parts 1–4. Submit `ArmSimulator.java` using the command

```
submit 4421 a2 ArmSimulator.java
```

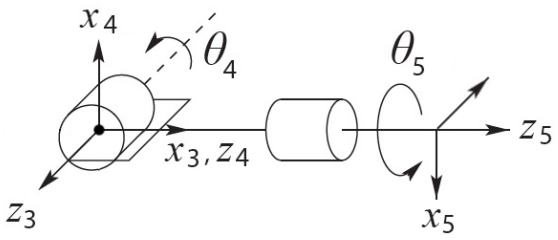

Figure 1: The wrist of the A150 and A255 robots. z_3 points out of the page and y_5 points into the page.