RBE 501 Week 2 Assignment

1st Arjan Gupta
Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
agupta11@wpi.edu

***Abstract*—This document provides an in-depth solution for Problems 3–2 and 3–5 described in the Robot Modeling and Control textbook. This is the assignment for the second week in RBE 501 (Robot Dynamics), Spring 2023 at Worcester Polytechnic Institute.**

***Index Terms*—robotics, forward kinematics, manipulator**

I. Introduction

We are asked to solve Problems 3–2 and 3–5 of the main textbook [1]. For Problem 3–2, the objective is to calculate the forward kinematic equations of the robot shown in Figure 1 below, *without* using the DH-convention.

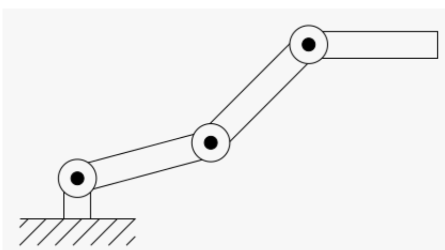

Fig. 1. Three-link planar arm of Problem 3–2

For Problem 3–5, the objective is to calculate the forward kinematic equations of the robot shown in Figure 2 below, using the DH-convention.

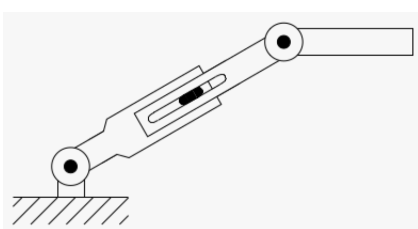

Fig. 2. Three-link planar arm with prismatic joint of Problem 3–5

II. Materials and Methods

A. Approach for Problem 3–2

1) Restate objective in technical terms: In technical terms, we must first assign frames 0 through 3 for each link of the manipulator. Then, we need to find the homogeneous transformation matrices between each frame, i.e. we need to find H_1^0, H_2^1, and H_3^2. Using these, we need to find H_3^0 to give us our final answer.

2) Assignment of frames: The first step toward solving our problem is to redraw our robot in symbolic form, and assign frames for links 0 through 3. This is shown in Figure 3. Frame 0 ($x_0y_0z_0$) is assigned at the first joint, frame 1 ($x_1y_1z_1$) is assigned at the second joint, and frame 2 ($x_2y_2z_2$) is assigned at third joint. Frame 3 ($x_3y_3z_3$) is assigned at the end of link 3.

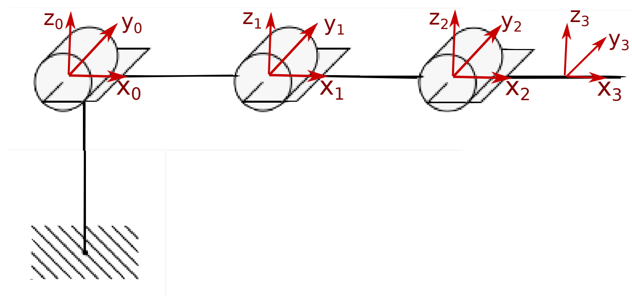

Fig. 3. Symbolic form of Figure 1, along with frame assignments

3) Variables and constants: Now that we have assigned the frames, we can come up with some constants for the link lengths. Assume that link 1 is of constant length L_1, link 2 is of constant length L_2, and link 3 is of constant length L_3. Additionally, we are dealing with 3 revolute joints, so when the first revolute joint moves, an angle subtended by x_1 from x_0 is the variable θ_1. Similarly, when the second and third revolute joints move, we get a variable angles of θ_2 and θ_3.

4) Find H matrices: We can now start formulating our homogeneous transformation matrices.

We first form the rotation matrices. The rotation matrices are given by the angle of rotation θ about the Y axis, given as the following.

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

In our LiveScript, we use the following MATLAB code to formulate R_y.

```
function Ry = formulate_Ry(theta)
    Ry = [cos(theta) 0 sin(theta); 0 1 0; ...
        -sin(theta) 0 cos(theta)];
end
```

Since all our link lengths are constants along the x-axis of the respective frames, they take on the following form.

$$d = \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}$$

Furthermore, we use another handy function to compute the homogeneous transformation matrices.

```
function H = compute_H(R,d)
    H = [R, d; zeros(1,3), 1];
end
```

Using this function and the previous one, we found the following H matrices for the frame-to-frame transformations.

$$H_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & L_1 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} R_2^1 & d_2^1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & L_2 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = \begin{bmatrix} R_3^2 & d_3^2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & L_3 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_3 & 0 & \cos\theta_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now we are ready to multiply these matrices to obtain the final result of our objective.

B. Compute all rotation matrices

As shown in the figure, there are no rotational differences between frames 0, 1, and 2. Therefore, the corresponding rotational matrices are the 3×3 identity matrix. So,

$$R_1^0 = R_2^0 = I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

By visually inspecting the figure, we also determine that the rotation for frame 3 can be obtained by a $-180°$ rotation around the x-axis, and a $-90°$ rotation around the z-axis. Using our MATLAB Live Script, we can determine this as,

$$R_3^0 = R_{x,-\pi} R_{z,-\frac{\pi}{2}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Similarly, the rotation for frame 2 with respect to frame 3 can be obtained by a $180°$ rotation around the y-axis, and a $90°$ rotation around the z-axis. This can be stated as,

$$R_2^3 = R_{y,\pi} R_{z,\frac{\pi}{2}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

C. Compute all the translations

By visually inspecting the figure, it is easy to see that,

$$d_1^0 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

In order to find d_2^0 and d_3^0, we need to consider the geometric distance from frame 0 to the very center of the cube. In the XY plane, we consider the table top as a square ABCD of side 1, as shown in Figure 2. Using the geometric properties of a square, we determine that $AE = EB = 0.5$ because $\triangle AEO \cong \triangle EOB$.

For the Z component of the translations, we need to consider half of the height of the cube, which is 10 cm, or 0.1 m. Therefore, we can write,

$$d_2^0 = \begin{bmatrix} -0.5 \\ 1.5 \\ 1.1 \end{bmatrix}, d_3^0 = \begin{bmatrix} -0.5 \\ 1.5 \\ 3 \end{bmatrix}$$

Also, since the cube is directly under the camera, we can state,

$$d_2^3 = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

III. Results

Given the fact that we have found all of the rotation matrices and rigid transformations, we can now form the final results of our objective. These are as follows,

$$H_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^0 = \begin{bmatrix} R_2^0 & d_2^0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 1.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^0 = \begin{bmatrix} R_3^0 & d_3^0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -0.5 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^3 = \begin{bmatrix} R_2^3 & d_2^3 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Furthermore, a MATLAB Live Script has been submitted alongside this report to show the code used to compute these results.

IV. Discussion

In the opinion of the author, this problem is very practical in real-world robotics. If a 3-DOF robotic arm is being operated (which seems to be the case in Figure 1), then one of the modern ways to monitor the arm would be to use computer vision via a camera. An overhead camera provides a good perspective in such a case, where the robot perhaps needs to perform tasks on the table-top. Some examples of table-top tasks could be moving chess pieces, manufacturing assembly, and other pick-and-place operations. In the case of this problem, the robot is doing something with a cube.

Generally, in such a setup, it is imperative to know the relative transformations between the frames of the robot, table, and the camera in order for a computer to properly command the controllers of the robot to perform accurate and precise tasks. For a simple example, if the computer wants to command the robot to go to a location on the table, first the computer needs to compute that location using H_1^3. Next, the computer needs to use H_1^0 to help the robot know about that location.

Calibration and frame transformations are very important in order for a robot to perform any real-world tasks. As for students of robotics, our spatial and geometric understanding of robotics is tested by this type of problem.

References

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, “Robot modeling and control,” 2006.