Robotic Manipulation: A Practical Application of Linear Algebra in Perception and Path Planning

Akhil Sankar, Mentor: Prof. Ariane Masuda

Department of Mathematics, New York City College of Technology of The City University of New York (CUNY)

Objective

In this project, we describe the mathematics that informs the decision making of a SCARA robotic manipulator by performing a simple pick-and-place operation within a controlled workspace. The intent is to showcase a practical implementation of concepts within linear algebra, differential equations and trigonometry as the basis of this type of robotic control algorithm, allowing the reader to build a more intuitive understanding of these disciplines.

Implementation: Target Localization

The system is comprised of a 3 joint SCARA assembly (the arm) with two revolute (rotating) joints and a prismatic (linearly translating) joint. The assembly is mounted to a baseboard that also serves as the 2D plane representing the workspace.

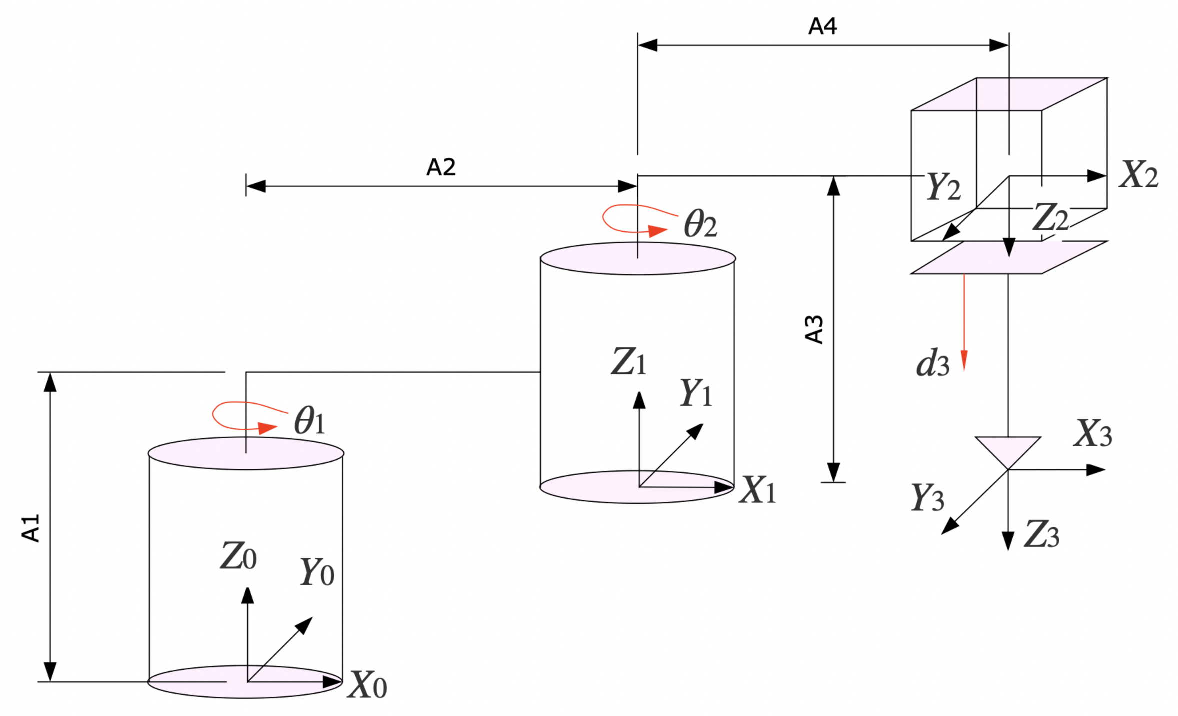

Figure 1:Denavit-Hartenburg representation of the SCARA manipulator.

Target objects are placed on the workspace and a combination of background subtraction and center-of-brightness computer vision techniques are used to determine the object coordinates in the physical space. The raw camera coordinates are then passed into a homogeneous transformation matrix (HTM), to account for the difference in camera and physical frame orientation and origin displacement [3].

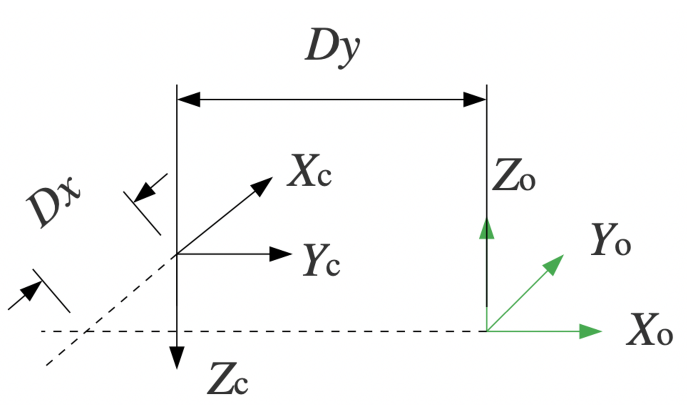

Figure 2:A graphical representation of the camera (C) frame and the physical (O) frame.

$$\begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix} = H_c^o \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_c^o & D \\ 0\ 0\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

The rotation matrix in the HTM is the result of two rotation operations performed consecutively on the O-frame [1, 4]:

$$\begin{aligned} R_c^o &= R(180^\circ, x)R(-90^\circ, z) \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \cos(180^\circ) & -\sin(180^\circ) \\ 1 & \sin(180^\circ) & \cos(180^\circ) \end{bmatrix} \begin{bmatrix} \cos(-90^\circ) & -\sin(-90^\circ) & 0 \\ \sin(-90^\circ) & \cos(-90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Implementation: Motion Control

The target coordinates (with respect to the O-frame) are used to determine the requisite joint angles using inverse kinematics following the Denavit-Hartenburg parameters for defining reference frames. It should be noted that the joint θ_2 is placed in the 'elbow-up' orientation for this assembly, therefore always moving clockwise.

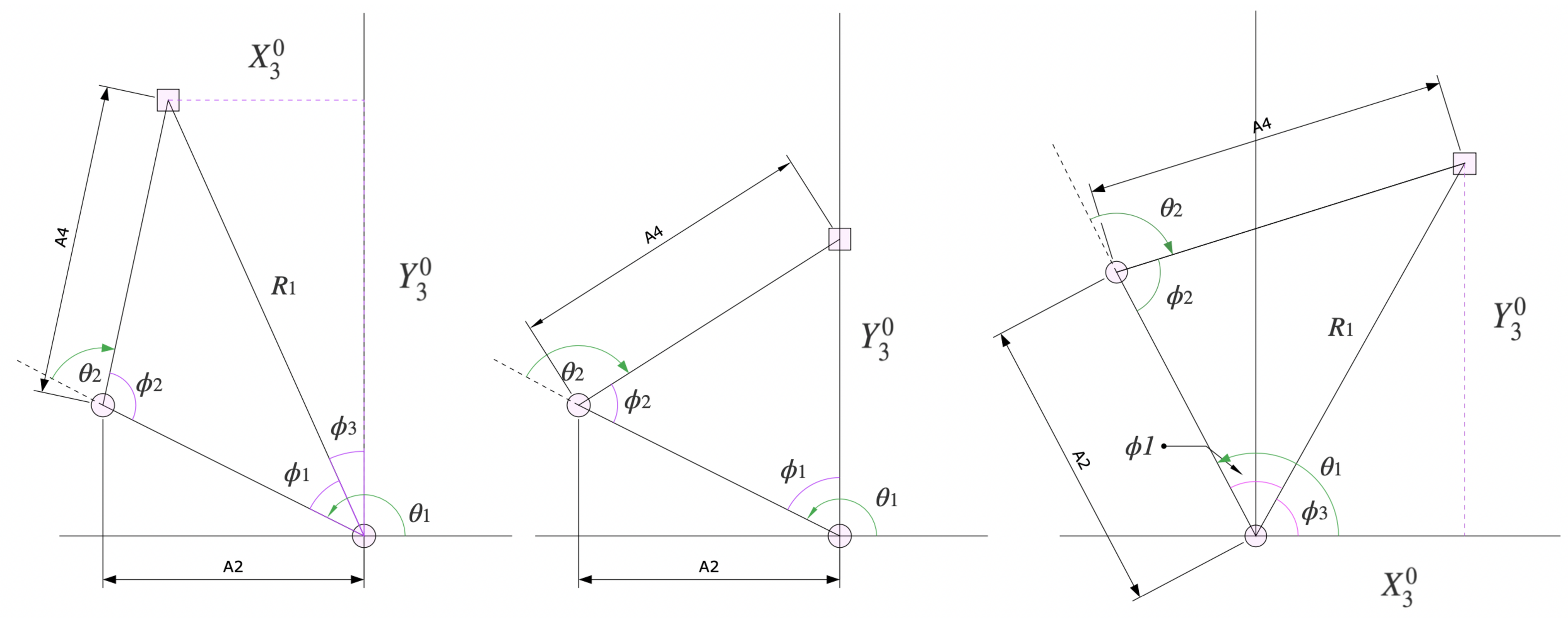

Figure 3:(from right to left): Top-down schematics describing the assembly orientation to navigate to target points on the first and second quadrants, respectively.

The joint angles θ_1 and θ_2 can therefore be calculated [2], given the target coordinates X_3^0 and Y_3^0, as shown below.

$$R_1 = \sqrt{(X_3^0)^2 + (Y_3^0)^2}$$

$$\theta_1 = \tan^{-1}\left(\frac{(A_2)^2 + (R_1)^2 - (A_4)^2}{2A_2R_1}\right)$$

$$\theta_2 = 180^\circ - \cos^{-1}\left(\frac{(A_2)^2 + (A_4)^2 - (R_1)^2)}{2A_2A_4}\right)$$

Results and Conclusion

The object localizer was relatively accurate, locating the target object within approximately 0.3 cm in the X- and Y-axes. Likewise, the motion control algorithm produced repeatable results, although this approach was less accurate, placing the end-effector only within around 2 cm of the target coordinates. This error can mostly be attributed to being unable to accurately calibrate the requisite duty cycle to achieve a specific servo angle, an issue outside the scope of this project.

Many improvements could be made to both the localization and motion control approaches. For example, a next iteration of the inverse kinematic implementation could include numerical approaches to solve non-linear equations generated for larger workspaces or to include elbow-down orientations as well.

References

[1] P. E. Red. 1 homogeneous transformations. URL `http://www.et.byu.edu/~ered/ME537/Notes/ch1-537.pdf`.

[2] P. A. Sodemann. Robotics 1: Inverse kinematics for position, . URL `http://robogrok.com/1-1-6_Inverse_Kinematics_of_a_SCARA_Manipulator.php`.

[3] P. A. Sodemann. Robotics 1: Image subtraction), . URL `http://robogrok.com/1-2-4_Background_Subtraction_and_Object_Localization.php`.

[4] P. A. Sodemann. Robotics 1: Camera coordinates), . URL `http://robogrok.com/1-2-5_Camera_Coordinates.php`.

More Information

- Contact: Akhil.Sankar@mail.citytech.cuny.edu
- Live demonstrations of the implementation @ YouTube
- Project codebase @ GitHub