David Amano
UID: 504464970
EE 183DA
Prof. Mehta

Laboratory 1: Forward and Inverse Kinematics

1 Introduction

A week ago, I visited Universal Studios, where the popular Harry Potter World attraction lies. The featured ride in the section of the park is the Forbidden Journey ride, which simulates the experience of flying through Hogwarts. In this lab, I will analyze the ride mechanism, which is a type of kinematic linkage. The ride mechanism resembles a robotic arm, with the seats of the passengers being the end effector. The base of the arm is able to rotate, and there are three additional joints along the arm. A picture of the arm is shown below.

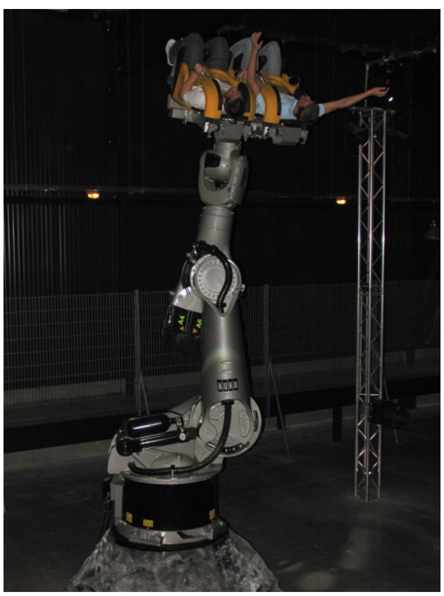

Figure 1: Robocoaster G2 manufactured by KUKA and Dynamic Attractions

Drawing a simplified version of the mechanism, a schematic of the arm is shown to contain four 1-DOF joints, comprising of four revolute joints. The schematic is shown below.

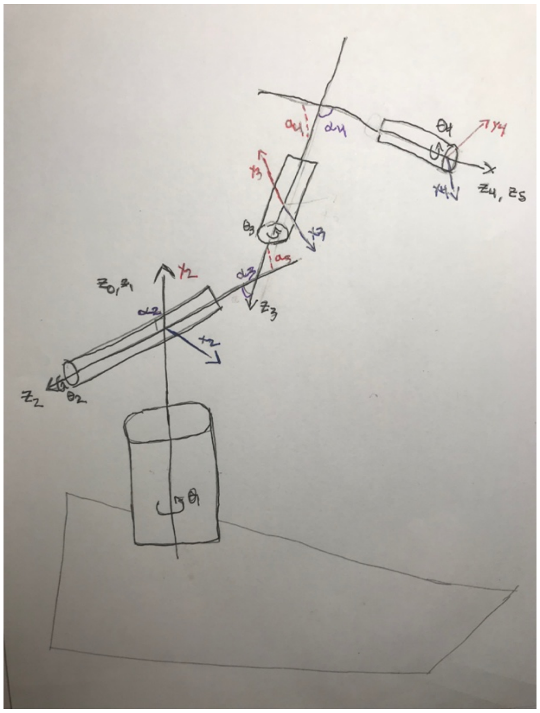

Figure 2: Simplified Schematic of Robocoaster

As depicted in the figure above, there are four revolute joints, represented by cylinders. The base frame is chosen to be the base of the arm, where z_0 points upward. This axis coincides with the z_1 axis, which points upward out of the first revolute joint, acting as the axis of rotation. This first joint is used to spin the base of the arm. The second joint rests on top of the first, but has a z_2 axis that is rotated to be perpendicular to the first. This joint allows for the lower part of the arm to swing back and forth. The third joint swings the upper part of the arm. The fourth joint spins the chairs (which we define as the end effector, or z_5).

The operational space of the robocoaster can be defined as the space in which the chairs can be moved. This is mostly limited to forward/backward motion, as well as the rotational motion of the chairs.

We can analyze the difference between two positions of the robocoaster. The default position will be defined as the position when the arm is straight and the chair is upright. A beginning ride position will be defined at a point where the robocoaster is stretched forward and the chair is mildly rotated.

2 Methods

2.1 Denavit-Hartenberg Parameters

index i	link length a_i	link twist α_i	link offset d_i	joint angle θ_i
0	0	0	0	0
1	0	0	0	θ_1
2	0	$-\pi/2$	0	θ_2
3	a_3	$\pi/4$	0	θ_3
4	a_4	$\pi/2$	0	θ_4
5	0	0	d_5	$\pi/2$

Table 1: Denavit-Hartenberg Parameters

The link lengths are the lengths of the arm in between the intermediary joints of the robocoaster. The estimated length of a_3 is 1.8m and the length of a_4 is 1.1m. The link offset d_5, the distance between the fourth joint and the end effector is estimated to be 0.5m.

Because all four joints analyzed are revolute joints, the joint variables are all θ and may be represented as

$$q = [\theta_{1,}\theta_{2,}\theta_{3,}\theta_4]^T$$

We can then derive the frames using the modified Denavit-Hartenberg parameters. The matrices of the different frames are calculated using the following

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i \; c\alpha_{i-1} & c\theta_i \; c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i \; s\alpha_{i-1} & c\theta_i \; s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrices we will calculate with the joint variables are ${}^0_1T(q_1)$, ${}^1_2T(q_2)$, ${}^2_3T(q_3)$, and ${}^3_4T(q_4)$. The transformation matrix ${}^4_5T()$ will also be used, but it does not depend on q, since there is no joint.

2.2 Forward Kinematics

Using forward kinematics, we can find the coordinates of the origin of the end effector in the base frame. We will use the default position as the set position with $(\theta_{1,}\theta_{2,}\theta_{3,}\theta_4) = (\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{2}, 0)$. The transformation is found by multiplying the transformations of each frame. This calculation is shown below

$$p_0 = {}_1^0T(q_1) * {}_2^1T(q_2) * {}_3^2T(q_3) * {}_4^3T(q_4) * {}_5^4T * p_5$$

The coordinates p_5 represents the coordinates of the origin of the end effector in its frame. Since it is in its own frame, the coordinates are (0, 0, 0).

I implemented this method using MATLAB. The script that calculates the forward kinematics is called FK.m. The result it returns for the end effector coordinates in world space is (0.354, 0.354, -2.9, 1), where the units are in meters.

2.3 Inverse Kinematics

We can use inverse kinematics to take a position in world space p_0 and calculate the angles. We will analyze the beginning ride position, defined as $p_0 = (0.4, 0.6, -2.5)$.

We first calculate the current position of the robocoaster using forward kinematics. Next, the difference between the final position and the current position is calculated. This will be done recursively until the error between the two values is minimized. We will need to use the Jacobian matrix below.

$$J = \begin{pmatrix} \partial x/\partial\theta_1 & \partial x/\partial\theta_2 & \partial x/\partial\theta_3 & \partial x/\partial\theta_4 \\ \partial y/\partial\theta_1 & \partial y/\partial\theta_2 & \partial y/\partial\theta_3 & \partial y/\partial\theta_4 \\ \partial z/\partial\theta_1 & \partial z/\partial\theta_2 & \partial z/\partial\theta_3 & \partial z/\partial\theta_4 \end{pmatrix}$$

Here, $(x\ y\ z)^T = {}_5^0T(q)p_5$. We will also need to use the pseudoinverse in finding the inverse kinematic. $As\ before, p_5$ is (0, 0, 0, 1) since this is the origin of the end effector in the base frame. (x, y, z) were discovered manually via matrix multiplication. The MATLAB function pinv (which finds the pseudoinverse) was also used. This calculation is documented in the IK.m script.

Results

To show how the recursive IK method gradually stepped and reduced the error between the position, a graph of the error on each iteration is shown below. I set the iteration to stop once the error reached 0.05, or the normalized difference between the current and final positions. The graph is shown below.

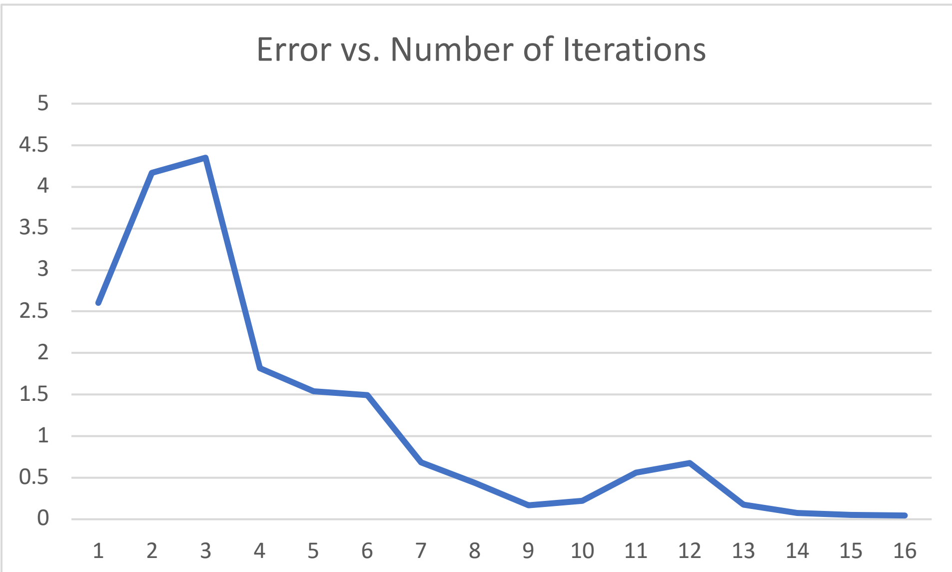

Figure 3: Graph of Error over Inverse Kinematic Iterations

I also recorded the position coordinates of the robocoaster after each iteration. The table below shows each movement the arm made to move toward the final position.

X	Y	Z
-2.33	1.63	0.48
0.06	1.43	1.76
-1.04	-0.50	-2.62
-0.83	-0.20	-2.03
-1.04	0.85	-2.18
-0.09	0.86	-2.10
0.70	0.29	-2.47
0.54	0.67	-2.45
0.61	0.66	-2.50
0.96	0.60	-2.47
0.62	0.77	-1.88
0.30	0.72	-2.43
0.48	0.62	-2.49
0.45	0.62	-2.50
0.44	0.62	-2.50

Figure 4: Position of the Robocoaster over Iterations

In summary, this lab took multiple hours. Finding the Jacobean by multiplying the matrices was the part that was least fun. I did like the idea of finding a real-world linkage and analyzing it. Drawing the linkage kinematic diagram was also challenging, but it really helped me to visualize what was happening in the problem.

The link to my github repo is: https://github.com/theDavidAmano/UCLA-EE-183DA/tree/master/Labs