Project 2

Alex Xu

I. DH Parameters

The model of the robot arm and gripper is a SCARA Mitsubishi Arm - Model RH-3FRH5515 and a Yamaha YRG-4220W. Fig. 1 and Fig. 2 show the measurements of the robot as well as the frames used for DH parameters.

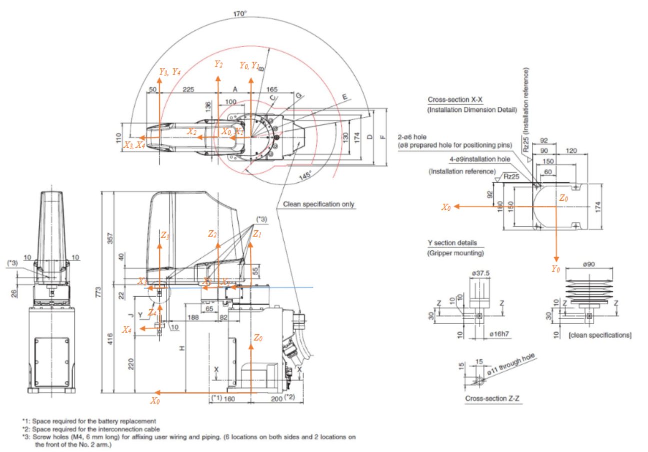

Fig. 1: Diagram of the robot arm with relevant frames [1]

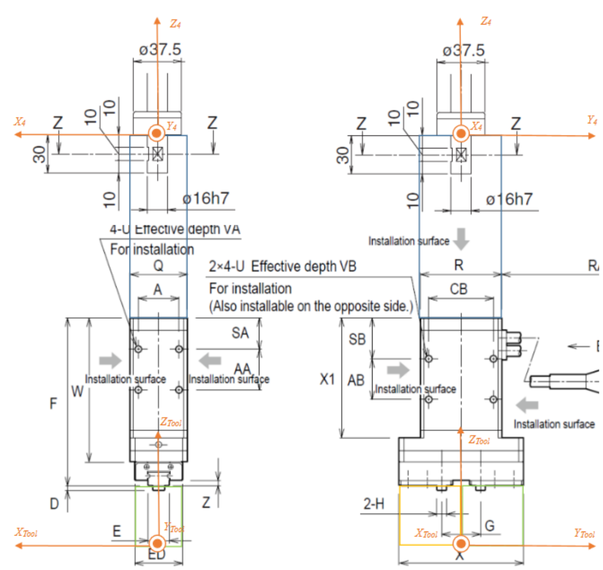

Fig. 2: Diagram of the gripper showing the 4th frame and tool frame [2]

The DH parameters used for the robot are described in Table I. The Matlab Robotics Toolbox used does not support a negative value for d so the 4th frame has a value of π for α. To compensate, the tool frame also has a value of π for α.

i-1	i	a (mm)	α	d (mm)	θ
0	1	0	0	400	θ_1
1	2	325	0	0	θ_2
2	3	225	0	0	θ_3
3	4	0	0	d	0
4	tool	0	0	30	0

TABLE I: DH parameters

DOF	Range	Max Velocity	Acceleration
θ_1	$340(\pm 170)°$	$420\frac{deg}{s}$	$50\frac{deg}{s^2}$
θ_2	$290(\pm 145)°$	$720\frac{deg}{s}$	$50\frac{deg}{s^2}$
θ_3	$720(\pm 360)°$	$3000\frac{deg}{s}$	$50\frac{deg}{s^2}$
d	$0-150mm$	$1100\frac{mm}{s}$	$50\frac{mm}{s^2}$

TABLE II: DH parameters

The joint ranges, velocities, and accelerations are shown in Table II

II. Forward and Inverse Kinematics

The robot is an RRRP-type. Within its workspace it can access a point $(p_x, p_y, p_z$ in 3D space and apply a yaw orientation. The frame of the end effector looks like:

$$T_{Tool}^{0} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since the only joint that offers any control in the z-axis is J4, $d = p_z$ provides that the p_z is within the range of d.

Now we can restrict this to a 2D problem as shown in Fig. 3.

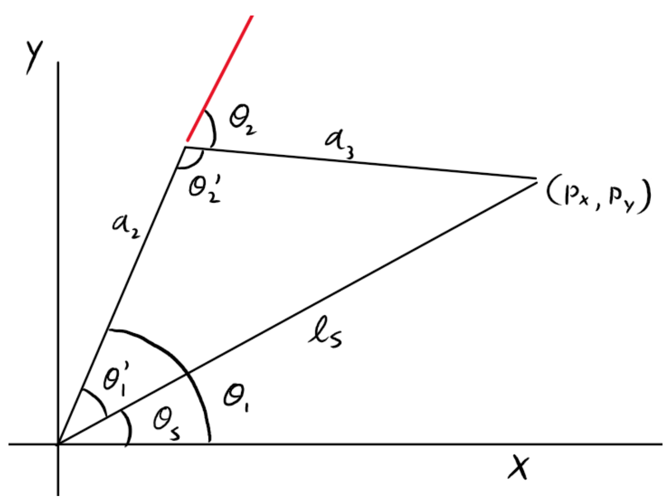

Fig. 3: 2D Geometric setup for IK.

The angle and distance of a point relative to the origin is:

$$\theta_S = atan2(p_y, p_x) \tag{2}$$

$$l_S = \sqrt{p_x^2 + p_y^2} \tag{3}$$

Using the law of cosines, the angle of J2 can be found as:

$$c = \frac{(a_2^2 + a_3^2 - l_s^2)}{2 * a_2 * a_3} \tag{4}$$

$$\theta_2 = atan2(\pm\sqrt{1-(c)^2}, c) + \pi \tag{5}$$

Similarly, the angle of J1 can be found as:

$$s = \frac{a_3 * \sqrt{1-(c)^2}}{l_s} \tag{6}$$

$$\theta_2 = atan2(\pm s, \sqrt{1-s^2}) + \theta_S \tag{7}$$

Finally, the angle of J3 is:

$$\theta_3 = atan2(r_{21}, r_{11}) - \theta_2 - \theta_1 \tag{8}$$

III. Locations of the Feeder and PCB

The robot is centered on the world frame S at:

$$T_0^S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The feeder station is located $0.5m$ along the y-axis from the robot at a height of $0.22m$ above the ground. This can be represented as:

$$T_{Feeder}^S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The PCB station is located $0.35m$ along the y-axis from the robot at a height of $0.22m$ above the ground. This can be represented as:

$$T_{PCB}^S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.35 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

The four target locations of the PCB are the corners. The frame of each corner with reference to the world is:

$$T_S^1 = \begin{bmatrix} 1 & 0 & 0 & 0.045 \\ 0 & 1 & 0 & 0.395 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$T_S^2 = \begin{bmatrix} 0 & -1 & 0 & 0.045 \\ 1 & 0 & 0 & 0.305 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$T_S^3 = \begin{bmatrix} -1 & 0 & 0 & -0.045 \\ 0 & -1 & 0 & 0.305 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$T_S^4 = \begin{bmatrix} 0 & 1 & 0 & -0.045 \\ -1 & 0 & 0 & 0.395 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

The positions of the feeder, PCB and chip are shown in Fig. 4.

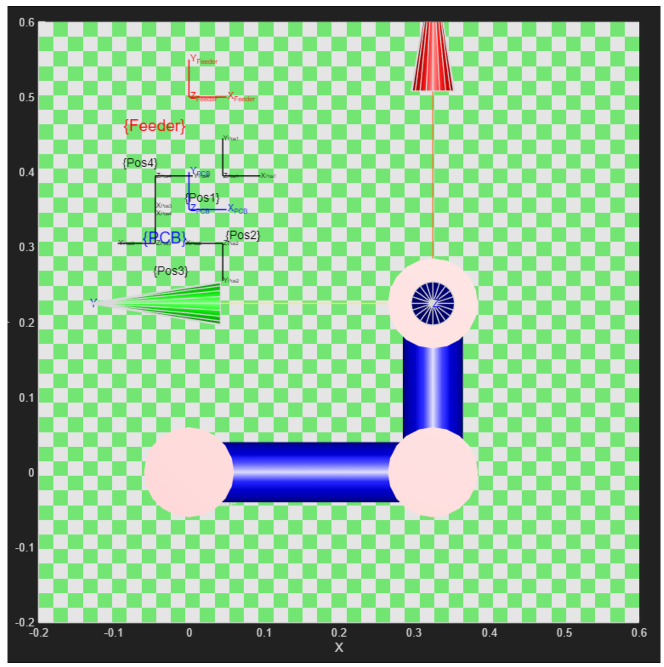

Fig. 4: Locations of the feeder, PCB, and chip positions relative to the robot. The robot is in its initial position.

IV. Locations of the via points

In order to avoid collisions with the PCB or feeder, the robot brings the chip component up before moving to another spot. These locations are represented as follows.

$$T_S^{1,via_1} = \begin{bmatrix} 1 & 0 & 0 & 0.045 \\ 0 & 1 & 0 & 0.395 \\ 0 & 0 & 1 & 0.37 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$T_S^{2,via_1} = \begin{bmatrix} 0 & -1 & 0 & -0.045 \\ 1 & 0 & 0 & 0.305 \\ 0 & 0 & 1 & 0.37 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$T_S^{Feeder,via_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0.37 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

The robot also avoids moving over the PCB when going from the feeder to position 2. These via points are:

$$T_S^{2,via_2} = \begin{bmatrix} 0 & -1 & 0 & 0.065 \\ 1 & 0 & 0 & 0.415 \\ 0 & 0 & 1 & 0.37 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

$$T_S^{2,via_3} = \begin{bmatrix} 0 & -1 & 0 & 0.055 \\ 1 & 0 & 0 & 0.315 \\ 0 & 0 & 1 & 0.37 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

V. Joint Space Trajectory

The joint space trajectory is generated using a linear-parabolic blending for single time intervals. Multiple-time-interval blending may make the process faster, however tests done with my code were hard to tune, and I could not get the last velocity to be constrained to 0. Single-time intervals blending means that the robot will reach every via-point and stop completely.

The maximum acceleration for each joint is fixed. Since we want the robot to work as fast as possible, we can use the max acceleration for the parabolic blends.

The blending time is calculated as:

$$t_b = \frac{T}{2} - \frac{\sqrt{(a_{max}^2 T^2 - 4\|a_{max}\| \|q_f - q_0\|)}}{\|(2 * a_{max})\|} \tag{21}$$

In order for t_b to be valid, $a_{max}^2 T^2 >= 4\{a_{max}\| \|q_f - q_0\|)$. To find a value of t_b, we increase T from 0 until the above condition is satisfied.

Figs. 5 to 7 show the position, velocity, and acceleration of the 4 joints.

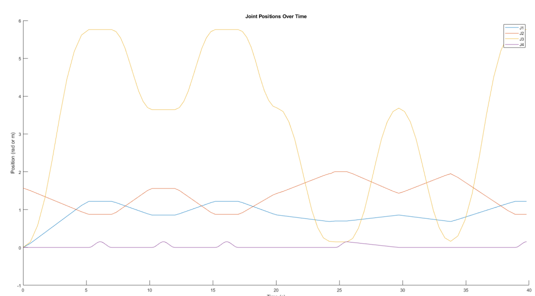

Fig. 5: Joint positions during all trajectories

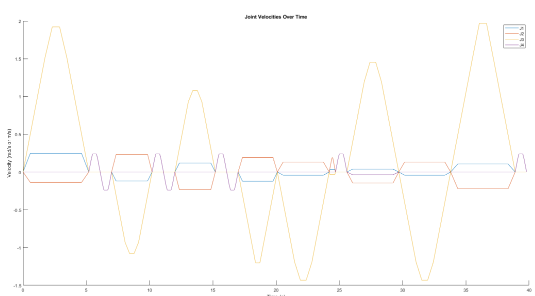

Fig. 6: Joint velocities during all trajectories

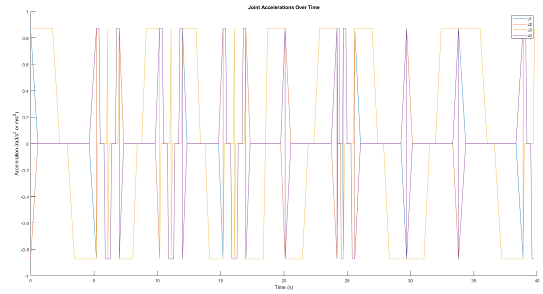

Fig. 7: Joint accelerations during all trajectories

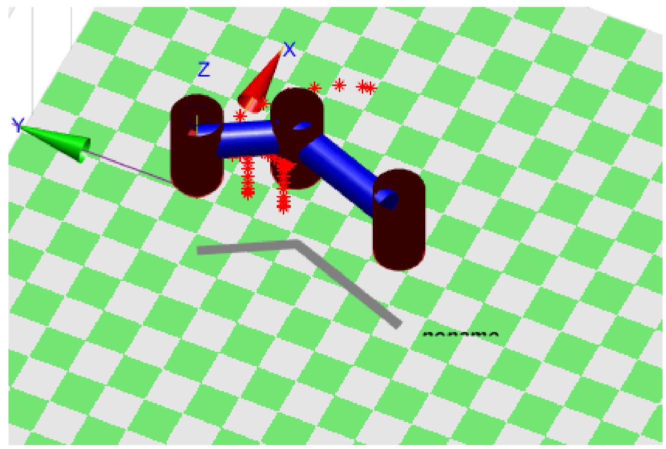

Fig. 8: Full Joint Space Trajectory

The entire process took about 40 seconds. Fig. 8 shows the full trajectory. Due to the nature of joint interpolations, the robot tends to move in arcs.

VI. Task Space Trajectory

In order to interpolate in the task space, the components of the tool matrix need to be extracted, as the rotation matrix cannot be linearly interpolated. Each transformation matrix is represented by 4 numbers (p_x, p_y, p_z, θ). $theta$ is the z-rotation in the rotation matrix, and there is only one rotation happening.

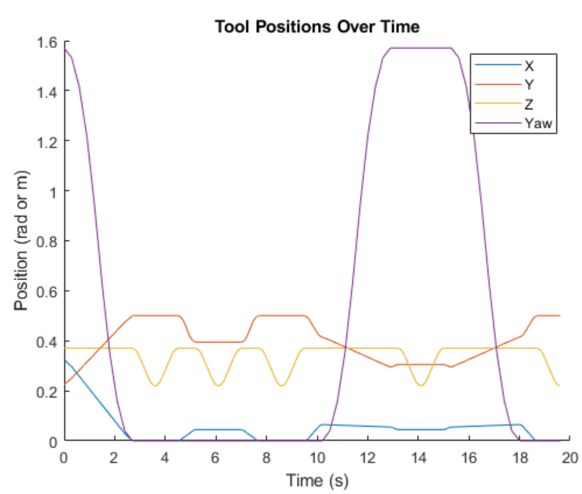

Fig. 9: Tool positions during all trajectories

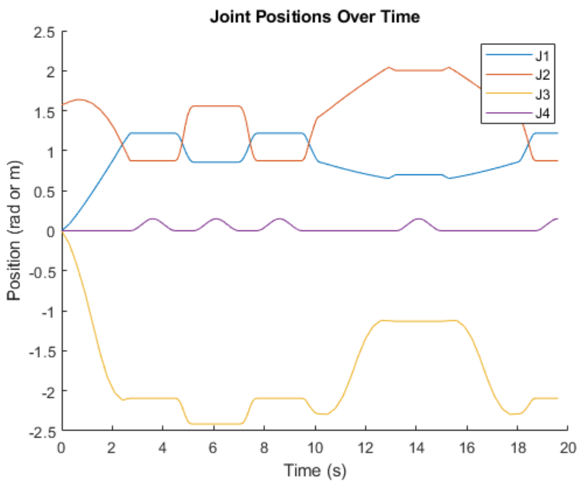

Fig. 10: Joint positions during all trajectories done in task space

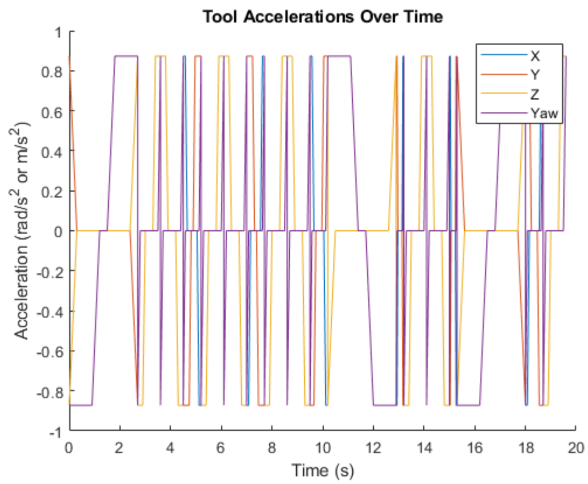

Fig. 12: Tool accelerations during all trajectories

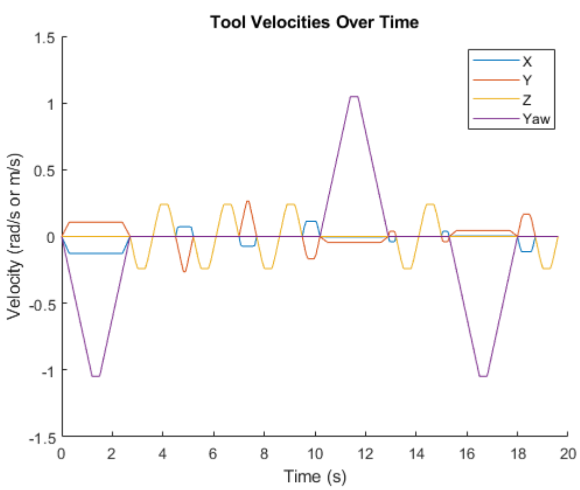

Fig. 11: Tool velocities during all trajectories

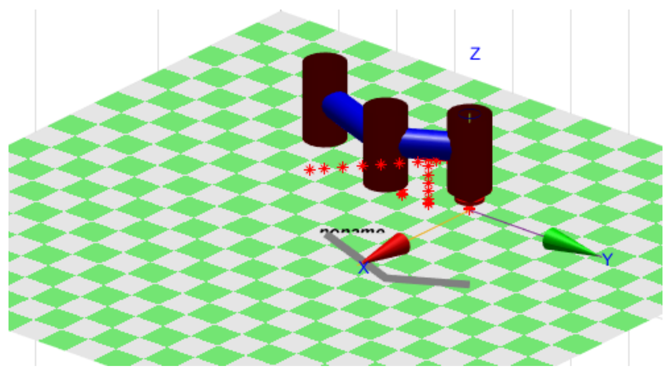

Fig. 13: Task space trajectory

The task space trajectory turned out much faster than the joint space trajectory. Unlike a joint interpolation from A to B which would cause the joint to swing in a circular arc, a task space interpolation from A to B would be a straight line which can be quicker. The downside to task space interpolation is that every single task interpolation needs to be run through the inverse kinematics calculations. Fig. 13 shows how the task space mainly moves in straight lines in comparison to Fig. 8.

VII. Trajectory Optimization

Putting the PCB to the side of the robot allows the robot to use J1 more, allowing for a wide swinging motion to reach its destination faster. Shifting the PCB 0.2 m to the left allows the robot to finish all the trajectories in 25s whereas the original route took 40s.

The new location of the PCB is:

$$T^S_{PCB} = \begin{bmatrix} 1 & 0 & 0 & -0.2 \\ 0 & 1 & 0 & 0.35 \\ 0 & 0 & 1 & 0.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

References

[1] M. Electric, "Mitsubishi electric knowledge base document," 2025, https://us.mitsubishielectric.com/fa/en/support/technical-support/knowledge-base/getdocument/?docid=3E26SJWH3ZZR-38-2664.

[2] L. Yamaha Motor Co., "Yamaha robotics general information," 2025, https://global.yamaha-motor.com/business/robot/lineup/yrg/w/.