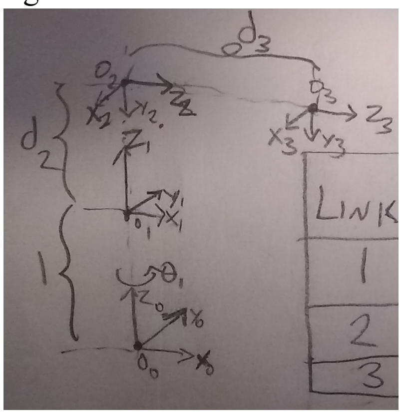
16-384 Capstone

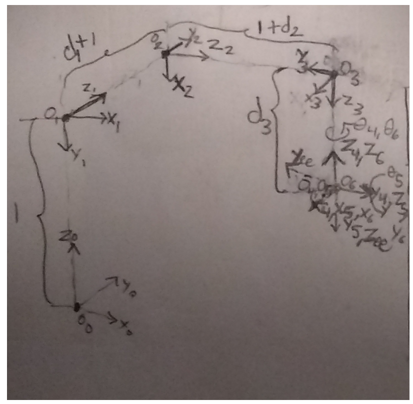
2.2 Report

2.2.1. Written Questions

2.2.1.1.) **Denavit-Hartenberg Transformations**

Frame assignments:

RobotA: 

RobotB: 

Robot A				
LINK i	a_i	α_i	d_i	θ_i
1	0	0	1	θ_1^*
2	0	$-\pi/2$	d_2^*	$-\pi/2$
3	0	0	d_3^*	0

Computed using Robot.fk(q) in Report_2211.m

$$q = [45° \quad 2 \quad 3]^T$$

$$T_3^0 = H_1^0 H_2^1 H_3^2 = A_1 * A_2 * A_3$$

$$\rightarrow T_3^0 = \begin{bmatrix} \sqrt{2}/2 & 0 & \sqrt{2}/2 & 2.1213 \\ -\sqrt{2}/2 & 0 & \sqrt{2}/2 & 2.1213 \\ 0 & -1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Robot B				
LINK i	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	1	0
2	0	$\pi/2$	$d_1^* + 1$	$-\pi/2$
3	0	$-\pi/2$	$d_2^* + 1$	$-\pi/2$
4	0	π	d_3^*	0
5	0	$-\pi/2$	0	θ_4^*
6	0	$\pi/2$	0	θ_5^*
7 = ee	0	π	0	θ_6^*

Computed using Robot.fkf(3,q) in Report_2211.m

$$q = [2 \quad 3 \quad 1 \quad 20° \quad 45° \quad 90°]^T$$

$$T_{ee}^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4 H_6^5 H_{ee}^6 = A_1 A_2 A_3 A_4 A_5 A_6 A_{ee}$$

$$\rightarrow T_{ee}^0 = \begin{bmatrix} -0.9397 & -0.2418 & 0.2418 & -4 \\ 0.3420 & -0.6645 & 0.6645 & 3 \\ 0 & \sqrt{2}/2 & \sqrt{2}/2 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Robot C (Capstone Robot)				
LINK i	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{d_i}$	$\boldsymbol{\theta_i}$
1	0	$\pi/2$	116.23	θ_1^*
2	327.76	π	0	θ_2^*
3	0	$-\pi/2$	2.5	$\theta_3^* - \pi/2$
4	254.10	$\pi/2$	94.05	$\theta_4^* - \pi/2$
5	266.70	$\pi/2$	54.23	θ_5^*

All units in [mm] and [rad].

- NOTE: Piazza Post @292 said to analyze the new diagram of the capstone robot and not the RobotC in the assignment.
-
- NOTE: Upon further measurement of the CAD, it was determined that d_3=2.5mm. This is only true for physical RobotB (not RobotA), which used dissimilar right angle brackets on joints 2 and 3, as is also the case in the provided CAD.

Computed using Robot.fkf(3,q) in Report_2211.

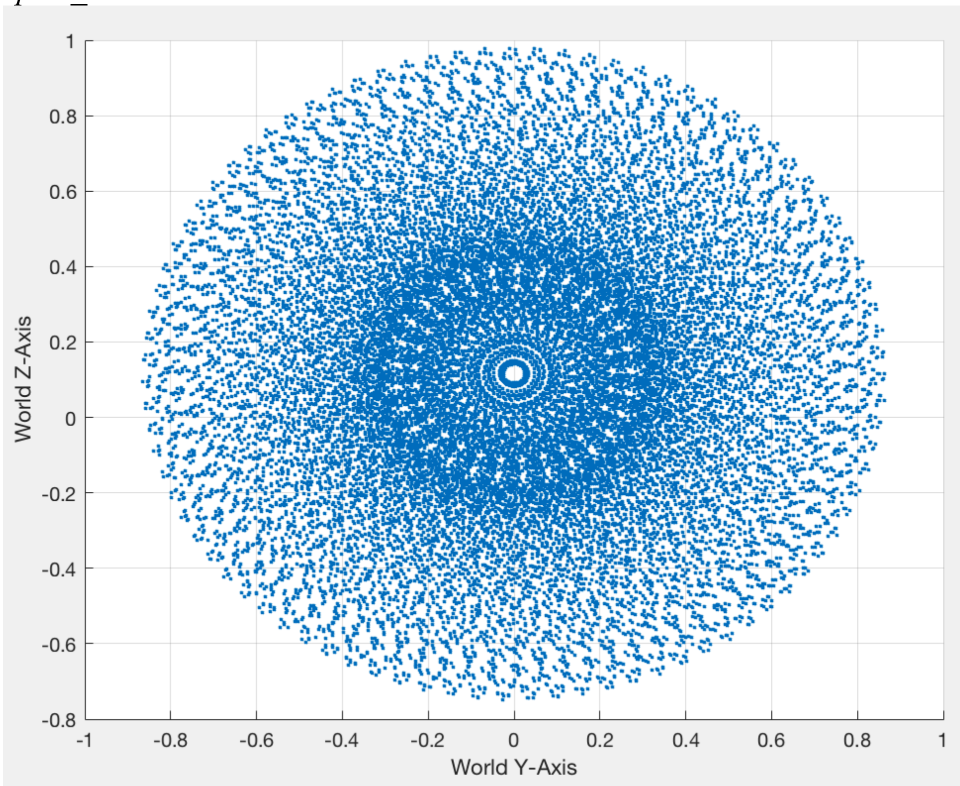
$$q = [20° \quad 45° \quad 45° \quad -45° \quad 90°]^T$$

$$T_{ee}^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4 = A_1 A_2 A_3 A_4 A_5$$

$$\rightarrow T_{ee}^0 = \begin{bmatrix} 0.9397 & 0.2418 & -0.2418 & 0.5076 \\ 0.3420 & -0.6645 & 0.6645 & 0.3378 \\ 0 & -\sqrt{2}/2 & -\sqrt{2}/2 & 0.1300 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2.1.2.) **Workspace**

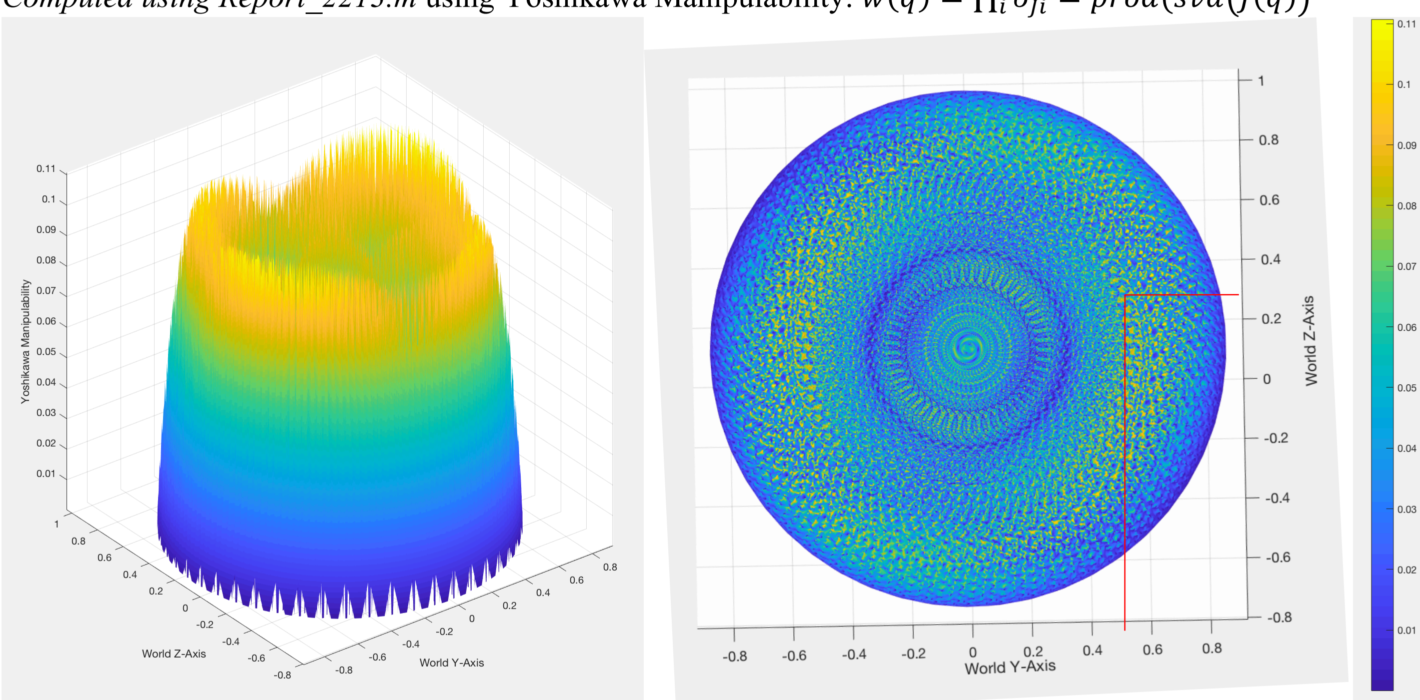
Computed using Report_2212.m



** units in [m].*

2.2.1.3.) **Manipulability**

Computed using Report_2213.m using Yoshikawa Manipulability: $w(\vec{q}) = \prod_i \sigma_{Ji} = prod\big(svd\big(J(\vec{q})\big)\big)$



** units in [m].*

*NOTE: All positions below z=0 are **inaccessible** due to the table.*

2.2.1.4.) **Determine Initial Home Position**

An ideal home position would be one with high manipulability and access to a large area of nearby points with similarly high manipulability. As such, some point near the center of the large yellow band of high manipulability which is also near to the starting position of the paths (eg. $[x_0, y_0, z_0] = [0.2, 0.6623, 0.27]m$ for the straight path). would be ideal.

Since, the plane of (y, z) is at $x = -0.05673m$, one such point which shares the same starting z value as the straight path and is closer to the origin is $[\boldsymbol{x_h}, \boldsymbol{y_h}, \boldsymbol{z_h}] = [\text{-}\mathbf{0.05673}, \mathbf{0.525}, \mathbf{0.27}]$**.** Using IK from the Robot3D class (in *Report_2214.m*), this gives an ideal home joint configuration of:
$\vec{\boldsymbol{q}}_{\boldsymbol{h}} = [\mathbf{1.1399}, -\mathbf{0.8125}, -\mathbf{0.4382}, \mathbf{1.1352}, -\mathbf{0.0938}]$**.**

2.2.1.5.) **3D Jacobians**

Since each ith column of the analytical jacobian for a 5-DOF serial revolute kinematic chain is:

$$J_{ai} = \begin{bmatrix} \frac{\partial x}{\partial \theta_i} \\ \frac{\partial y}{\partial \theta_i} \\ \frac{\partial z}{\partial \theta_i} \\ \frac{\partial \phi}{\partial \theta_i} \\ \frac{\partial \theta}{\partial \theta_i} \\ \frac{\partial \psi}{\partial \theta_i} \end{bmatrix} \rightarrow J_a = \begin{bmatrix} I & 0 \\ 0 & B^{-1}(\alpha) \end{bmatrix} J_g \text{ where } B = \begin{bmatrix} c_\psi s_\theta & -s_\psi & 0 \\ s_\psi s_\theta & c_\psi & 0 \\ c_\theta & 0 & 1 \end{bmatrix}$$

ϕ, θ, ψ are roll, pitch, and yaw respectively of the Z-Y-Z Euler angles and J_g is the geometric jacobian, where each column is

$$J_i = \begin{bmatrix} \hat{z}_{i-1}^0 \times (o_n - o_i) \\ \hat{z}_{i-1}^0 \end{bmatrix}$$

In turn, each of these can be computed as follows:

$$\hat{z}_{i-1}^0 = H_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad o_i = H_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

Each of these homogeneous transforms can be computed using serial multiplication of local Denavit-Hartenberg transforms.

$$H_i^0 = \prod_{k=1}^i A_k = \prod_{k=1}^i \begin{bmatrix} c_{\theta_k'} & -s_{\theta_k'} c_{\alpha_k'} & s_{\theta_k'} s_{\alpha_k'} & a_k c_{\theta_k'} \\ s_{\theta_k'} & c_{\theta_k'} c_{\alpha_k'} & -c_{\theta_k'} s_{\alpha_k'} & a_k s_{\theta_k'} \\ 0 & s_{\alpha_k'} & c_{\alpha_k'} & d_k' \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ where d', θ_k', α_k' refer to Denavit-Hartenberg parameters, not Euler angles.

Lastly, using these matrices, the Euler angles can be resolved as follows, using atan2(y,x) syntax:

$$let: T = H_n^0$$
$$\phi = atan2(T_{21}, T_{11})$$
$$\theta = atan2\left(-T_{31}, \sqrt{T_{32}^2 + T_{33}^2}\right)$$
$$\psi = atan2(T_{32}, T_{33})$$

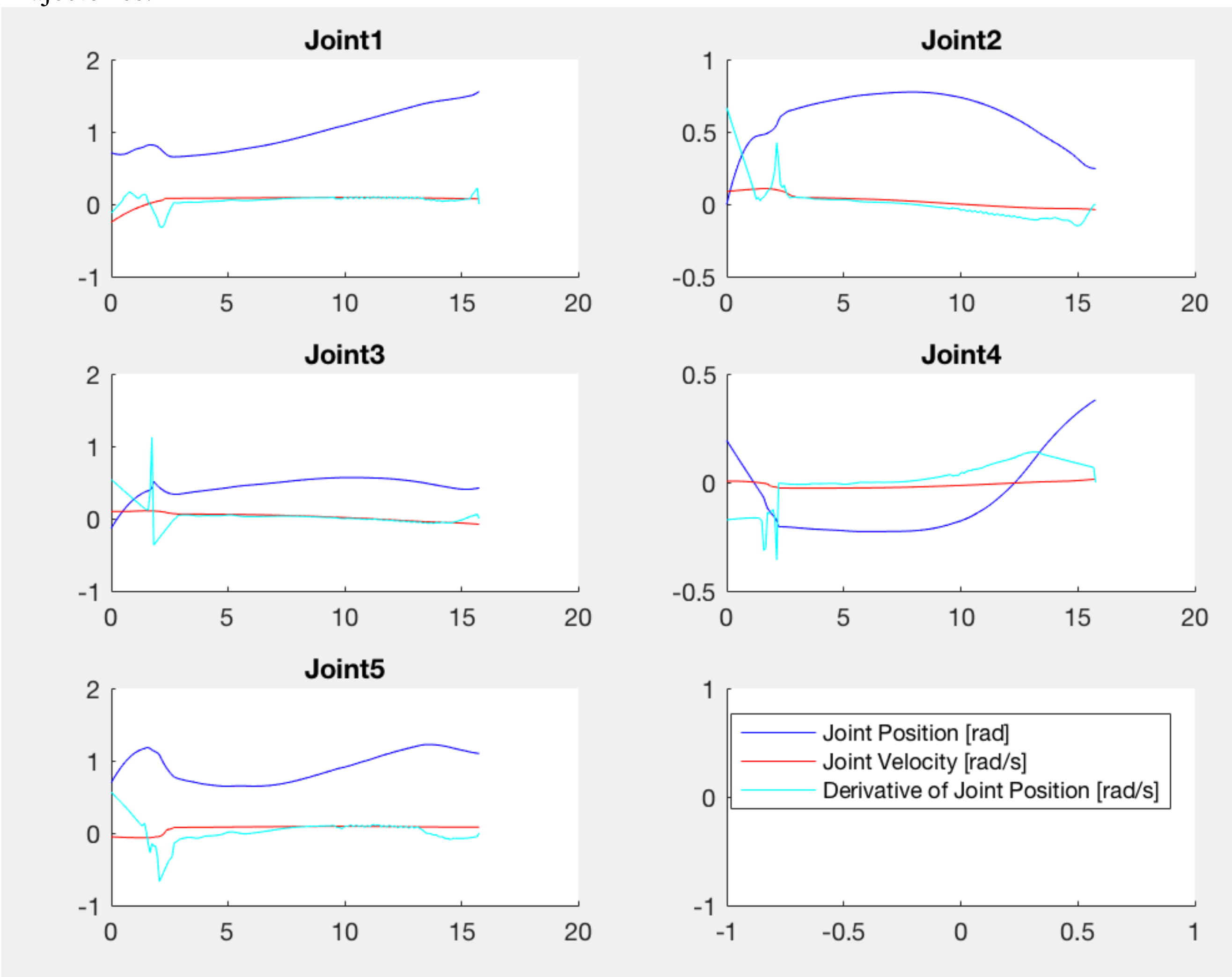
To solve for the analytical jacobian, all of these variables can be entered symbolically into MATLAB. However, my computer would crash trying to determine H_n^0.

2.2.2. Reflection

2.2.2.1.) **Description of Approach:**

The total path length of the interpolated trajectory between the given waypoints was determined. A time-parameterized constant jerk trajectory (CJT.m) was generated for this path. For each of these positions, IK was performed, ignoring the yaw euler angle to generate a matrix trajectory of joint angles and velocities. These were then smoothed out to remove any staircase effects from IK using matlab's smooth().

Trajectories:



See _*README*.m for map of files.

2.2.2.1.) **Plotting Data:**

Torque plot at 0.04m/s path velocity	Torque plot at 0.01m/s path velocity

When moving very slowly,

$\tau = \sum_i J_i^T F_i$ where i is each center of mass in the system and J_i^T is the jacobian to that center of mass.

When moving faster, dynamic loads, namely Coriolis forces, contribute substantial torques onto the joints. These could be computed by using the Lagrangian of the entire multi-rigidbody system to determine equations of motion. The Coriolis matrix $C(q, \dot{q})$, would allow one to determine the load on the system.