Mohammadreza Mousaei
Dongwei Bai
Andrew-ID: mmousaei, dongweib

Homework 3

16-711
March 19, 2021

Problem 1

(a)

$$\vec{q}_{1,2,3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{\omega}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{\omega}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\hat{\xi}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \hat{\xi}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \quad \hat{\xi}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

$$e^{\hat{\xi}_1\theta_1} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad e^{\hat{\xi}_2\theta_2} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$e^{\hat{\xi}_3\theta_3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_3) & -\sin(\theta_3) & 0 \\ 0 & \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$g_{st}(0) = \begin{bmatrix} 1 & 0 & 0 & l_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} g_{st}(\theta) &= e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} e^{\hat{\xi}_3\theta_3} g_{st}(0) \\ &= \begin{bmatrix} c_1c_2 & -s_1c_3 + c_1s_2s_3 & c_1s_2c_3 + s_1s_3 & l_0c_1c_2 \\ s_1c_2 & c_1c_3 + s_1s_2s_3 & s_1s_2s_3 - c_1s_3 & l_0s_1c_2 \\ -s_2 & c_2s_3 & c_2c_3 & -l_0s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Mohammadreza Mousaei
Dongwei Bai
Andrew-ID: mmousaei, dongweib **Homework 3**

(b)

$$g_{sb}(\theta) = e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} e^{\hat{\xi}_3\theta_3} g_{st}(0)$$
$$\longrightarrow g_{sb}(\theta) g_{st}^{-1}(0) = e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} e^{\hat{\xi}_3\theta_3}$$

assuming end effector has reached the destination: $g_{sb}(\theta) = g_d$, knowing that p_d is on third axis of rotation, we have $\vec{p}_d = e^{\hat{\xi}_3\theta_3}\vec{p}_d$

$$g_{sb}(\theta) g_{st}^{-1}(0)\vec{p} = e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2}\vec{p}$$

so, θ_1 and θ_2 could be solved using Paden-Kahan subproblem $e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2}\vec{p} = \vec{q}$. Then we have

$$e^{\hat{\xi}_3\theta_3} = e^{-\hat{\xi}_1\theta_1} e^{-\hat{\xi}_2\theta_2} g_d g_{st}^{-1}(0)$$

using arbitrary point $\vec{p}_{ar}$(which should not be along the axis 3)

$$e^{\hat{\xi}_3\theta_3}\vec{p}_{ar} = e^{-\hat{\xi}_1\theta_1} e^{-\hat{\xi}_2\theta_2} g_d g_{st}^{-1}(0)\vec{p}_{ar}$$

so, θ_3 could be solved using Paden-Kahan subproblem $e^{\hat{\xi}\theta}\vec{p} = \vec{q}$

(c)

For spatial Jacobin we have

$$J_{st}^s = \begin{bmatrix} \vec{\xi_1} & \vec{\xi_2'} & \vec{\xi_3'} \end{bmatrix}$$

where

$$\vec{\xi_i'} = Ad_{(e^{\hat{\xi}_1\theta_1}\ldots e^{\hat{\xi}_{i-1}\theta_{i-1}})}\hat{\xi}_i$$

$$Ad_{(e^{\hat{\xi}_1\theta_1})} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 & 0 & 0 \\ s_1 & c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1 & -s_1 & 0 \\ 0 & 0 & 0 & s_1 & c_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad Ad_{(e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2})} = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & 0 & 0 & 0 \\ s_1c_2 & c_1 & s_1c_2 & 0 & 0 & 0 \\ -s_2 & 0 & c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1c_2 & -s_1 & c_1s_2 \\ 0 & 0 & 0 & s_1c_2 & c_1 & s_1c_2 \\ 0 & 0 & 0 & -s_2 & 0 & c_2 \end{bmatrix}$$

$$J_{st}^s = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -s_1 & c_1c_2 \\ 0 & c_1 & s_1c_2 \\ 1 & 0 & -s_2 \end{bmatrix}$$

For body Jacobian we have

$$J_{st}^b = Ad_{g_{st}(\theta)}^{-1} J_{st}^s$$

$$g_{st}(\theta) = \begin{bmatrix} c_1c_2 & -s_1c_3 + c_1s_2s_3 & c_1s_2c_3 + s_1s_3 & l_0c_1c_2 \\ s_1c_2 & c_1c_3 + s_1s_2s_3 & s_1s_2s_3 - c_1s_3 & l_0s_1c_2 \\ -s_2 & c_2s_3 & c_2c_3 & -l_0s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Ad_{g_{st}(\theta)}^{-1} = \begin{bmatrix} c_1c_3 - s_1s_2c_3 & s_1c_3 + c_1s_2c_3 & -c_2s_3 & (c_1s_3 + c_1s_2c_3)l_0 & (s_1s_3 - c_1s_2c_3)l_0 & c_2c_3l_0 \\ s_1c_2 & -c_1c_2 & -s_2 & 0 & 0 & 0 \\ -c_1s_3 - s_1s_2c_3 & -s_1s_3 + c_1s_2c_3 & -c_2s_3 & (c_1c_3 + s_1s_2s_3)l_0 & (s_1 + c_3 - c_1s_2s_3)l_0 & -c_2s_3l_0 \\ 0 & 0 & 0 & c_1c_3 - s_1s_2s_3 & s_1c_3 + c_1s_2c_3 & -c_2s_3 \\ 0 & 0 & 0 & s_1c_2 & -c_1c_2 & -s_2 \\ 0 & 0 & 0 & -c_1s_3 - s_1s_2c_3 & -s_1s_3 + c_1s_2c_3 & -c_2c_3 \end{bmatrix}$$

$$\begin{aligned} J_{st}^b &= Ad_{g_{st}(\theta)}^{-1} J_{st}^s \\ &= \begin{bmatrix} 0 & 0 & 0 \\ l_0c_2c_3 & -l_0s_3 & 0 \\ -l_0c_2s_3 & -l_0c_3 & 0 \\ -s_2 & 0 & 1 \\ c_2s_3 & c_3 & 0 \\ c_2c_3 & -s_3 & 0 \end{bmatrix} \end{aligned}$$

Problem 2

In singular configuration, the Jacobian matrix drops rank, so does its transpose. Which means the nullspace of the Jacobian matrix transpose is nonzero. Using the equation

$$\vec{\tau} = J^T\vec{F}$$

Since the nullspace of J^T is non-zero, it means that there exists a non-zero vector $\vec{F}$ which satisfies

$$J^T\vec{F} = \vec{0}$$

which means there exists a force vector, for which no torque is needed to balance that force.

Problem 3

The code for this problem is provided in $Q3.m$ (first section of the code is leftover from last assignment).

(a)

As in assignment 2, ω and q used in this assignment are

$$\omega = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad q = 0.001 \begin{bmatrix} 610 & 720 & (1000+346) \\ 610 & 720 & (1000+346) \\ 610 & 720 & (1000+346) \\ 610 & 720+45 & (1000+346+550) \\ 610 & 720 & (1000+346+850) \\ 610 & 720 & (1000+346+850) \\ 610 & 720 & (1000+346+850) \end{bmatrix}$$

except q is converted to meters. From the code, part (a), jacobian matrices for the test joint angles are computed as:

$$J_s = \begin{bmatrix} 0.7200 & 0.1344 & 0.4396 & 0.7708 & -0.3219 & 1.8178 & -1.1043 \\ -0.6100 & -1.3393 & -0.6245 & -1.7636 & -0.8168 & -1.5531 & -1.4406 \\ 0 & 0.6555 & 0.1349 & 0.5781 & 0.4599 & 0.2569 & 0.9476 \\ 0 & -0.9950 & -0.0198 & -0.9216 & -0.1692 & -0.6414 & -0.5622 \\ 0 & -0.0998 & 0.1977 & -0.3836 & 0.5334 & -0.6981 & 0.7100 \\ 1.0000 & 0 & 0.9801 & 0.0587 & 0.8288 & 0.3183 & 0.4242 \end{bmatrix}$$

$$J_b = \begin{bmatrix} 1.1989 & -0.8225 & 1.4414 & -1.4776 & 1.5426 & -1.4147 & 0.7200 \\ -1.7450 & -0.8785 & -1.6899 & -1.0252 & -1.3023 & -1.6796 & -0.6100 \\ 0.4079 & 1.1637 & 0.2066 & 1.0150 & -0.0015 & 0.9437 & 0 \\ -0.7896 & -0.2128 & -0.6595 & -0.4163 & -0.3638 & -0.7648 & 0 \\ -0.4434 & 0.8462 & -0.4932 & 0.8680 & -0.4319 & 0.6442 & -0.0000 \\ 0.4242 & 0.4885 & 0.5672 & 0.2707 & 0.8253 & -0.0000 & 1.0000 \end{bmatrix}$$

(b)

Position and the orientation errors are computed as

$$e_p = p_d - p_s = [0.0178 \quad 0.0408 \quad -0.0059]^T$$
$$e_o = Q_d \cdot (Q_s)^{-1} = [0.0841 \quad -0.0776 \quad -0.1591]^T$$

corresponding velocities for these errors are:

$$v_p = \begin{bmatrix} e_p \\ 0 \end{bmatrix} = \begin{bmatrix} 0.0178 \\ 0.0408 \\ -0.0059 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$v_o = \begin{bmatrix} -e_o \times x_s \\ e_o \end{bmatrix} = \begin{bmatrix} -0.0059 \\ 0.2580 \\ -0.1290 \\ 0.0841 \\ -0.0776 \\ -0.1591 \end{bmatrix}$$

then the velocity would be

$$v = v_p + v_o = \begin{bmatrix} 0.0118 \\ 0.2989 \\ -0.1349 \\ 0.0841 \\ -0.0776 \\ -0.1591 \end{bmatrix}$$

(c)

Implementation could be found in part (c) of the code. Chosen parameters are $K_p = 0.1$ since higher p gains makes the p controller to overshoot, and lower p gains doesn't add to stability while makes the optimization slower. Stopping condition is total error squared (L2 norm of $[e_p \ \ e_o]$) less than $threshold = 1e - 5$. Joint angle trajectories are provided in $psudo\_1.txt$ and $psudo\_2.txt$. For x_{d1}, it took 149 iterations to converge and for x_{d2}, it took 178 iterations to converge. Final joint angles are

$$\theta_1 = \begin{bmatrix} 0.1543 \\ 0.2014 \\ 0.1962 \\ 0.4998 \\ 0.3135 \\ 0.3944 \\ 0.5403 \end{bmatrix}$$
$$\theta_2 = \begin{bmatrix} 0.6402 \\ 0.2245 \\ -0.9296 \\ 0.2332 \\ -0.0001 \\ -0.1505 \\ 0.6971 \end{bmatrix}$$

(d)

With tuned value of $\lambda = 0.0001$, DSL results in general are slower than psudo inverse (lower convergance rate), but the system has more stability, this can be shown in higher p gains, as an example Fig. 1 shows that with choice of $K_p = 0.5$, psudo inverse becomes unstable (for x_{d2}), but DLS remains stable. This shows even though DLS is slower with same p gain, we could have higher p gains for DSL while keeping it stable, thus achieve higher speed than psudoinverse with lower p gain.

Problem 4

The code for this problem is provided in $Q4.m$.

(a)

Inverse kinematics problem could be solved analytically. Since $l_{ds} = r_{ds}$

$$L = \sqrt{z_0^2 + r_{ds}^2} = 0.8246, \quad L = \sqrt{L_1^2 + L_2^2 - 2L_1L_2\cos(\pi - \theta_{lk,rk})}$$

Having $L_1 = L_2 = 0.5$, gives us

$$\theta_{lk,rk} = 68.8998$$

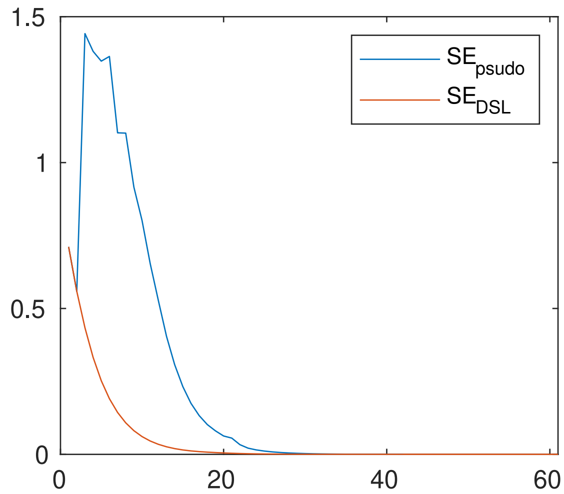

Figure 1: comparison between DLS and psudo-inverse methods with $K_p = 0.5$

Then we can write

$$\theta_{ra} = \arctan(\frac{x_0 - r_{ds}}{z_0}) - \arccos\left(\frac{L_1^2 + L^2 - L_2^2}{2LL_1}\right) = -20.4137$$
$$\theta_{la} = \arctan(\frac{-(l_{ds} + x_0)}{z_0}) - \arccos\left(\frac{L_1^2 + L^2 - L_2^2}{2LL_1}\right) = -48.4861$$

(b)

The code is provided in section (b) of $Q4.m$. Based on Fig. 2, the state will eventually converge to the desired state

(c)

The code is provided in section (c) of $Q4.m$. Based on Fig. 3 generated, the motors for knee joint will saturate and endanger the stabilization since both of them are over the value of 150Nm.

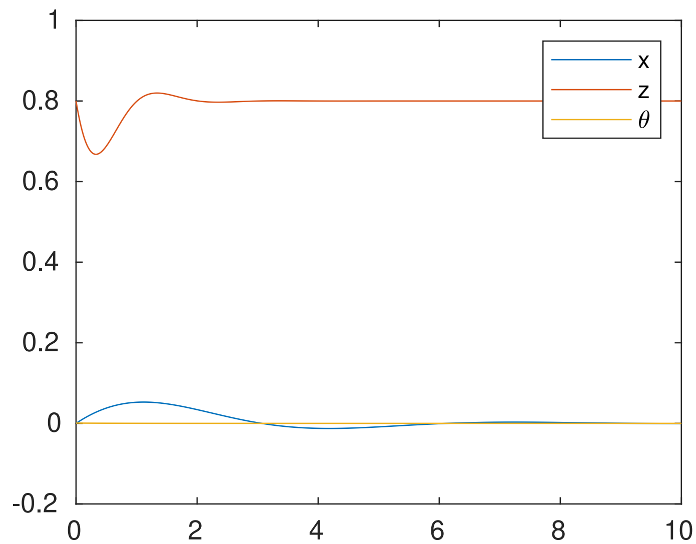

Figure 2: Convergence of states (x, z, θ)

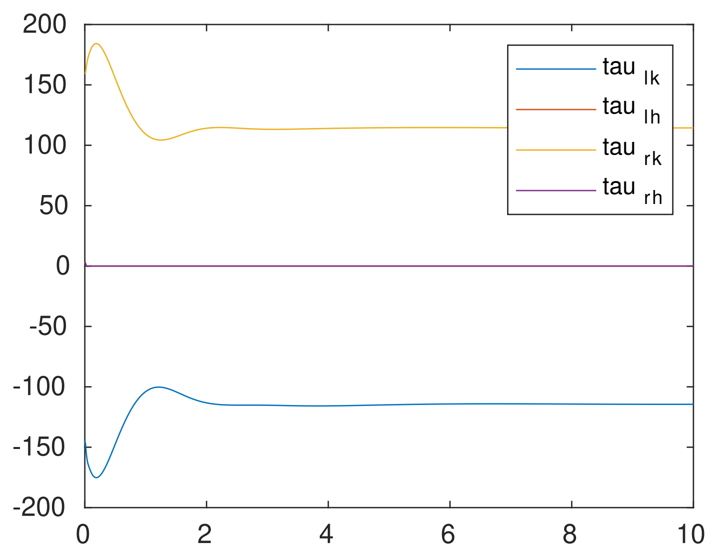

Figure 3: Joint torques