ME453 - Homework 3

Herman Kolstad Jakobsen
20196493

November 12, 2019

ME 453 Homework 3 Herman K. Jakobsen

20196493

Problem 7a)

Know that JJ^T is symmetric since

$$B = JJ^T$$

$$B^T = (JJ^T)^T = (J^T)^T J^T = JJ^T = B$$

Further, let λ be an arbitrary eigenvalue of JJ^T and let v be an eigenvector corresponding to the eigenvalue λ.

Then

$$(JJ^T)v = \lambda v \qquad (*)$$

The Jacobian J is obtained from physical quantities such as positions and coordinates. Hence, it is fair to assume that the Jacobian, and then also JJ^T, is real.

Taking the complex conjugate of $(*)$ therefore yields

$$(JJ^T)\bar{v} = \bar{\lambda}\bar{v}$$

Pre-multiplying $(*)$ with $\bar{v}^T$ gives

$$\bar{v}^T (JJ^T) v = \bar{v}^T \lambda v$$

$$\Rightarrow \lambda \bar{v}^T v = ((JJ^T)^T \bar{v})^T v$$

$$= ((JJ^T)\bar{v})^T v$$

$$= (\bar{\lambda}\bar{v})^T v$$

$$= \bar{\lambda}\bar{v}^T v$$

So

$$\lambda \bar{v}^T v = \bar{\lambda}\bar{v}^T v \quad \Rightarrow \quad (\lambda - \bar{\lambda})\bar{v}^T v = 0$$

Since v is an eigenvector and hence is non-zero, we have

$$\lambda = \bar{\lambda}$$

which means that λ is real. ■

Have that

$$x^T J J^T x = (J^T x)^T (J^T x) = \| J^T x \| \geq 0$$

So the matrix JJ^T is positive semi-definite

$$JJ^T \geq 0$$

This implies

$$\lambda \geq 0 \quad \blacksquare$$

7b)

Let $\lambda_i \neq \lambda_j$, and

$$(JJ^T)v_i = \lambda_i \qquad (*)$$

$$(JJ^T)v_j = \lambda_j \qquad (**)$$

Pre-multiplying $(*)$ with y^T yields

$$\lambda_i v_j^T v_i = v_j^T (JJ^T) v_i = ((JJ^T)v_j)^T v_i = (\lambda_j v_j)^T v_i = \lambda_j v_j^T v_i$$

Thus

$$\lambda_i v_j^T v_i - \lambda_j v_j^T v_i = 0$$

$$(\lambda_i - \lambda_j) v_j^T v_i = 0$$

Since $\lambda_i \neq \lambda_j$, we must have

$$v_j^T v_i = 0 \quad , \quad i \neq j$$

Hence, all eigenvectors are orthogonal to each eigenvector. ∎

Scanned by CamScanner

Problem 2a)

We are given the rotation matrix

$$R = R_{z,\alpha} R_{y,\beta} R_{z,\gamma}$$

Using the relationship

$$S(\omega) = \dot{R} R^T$$

we obtain

$$\dot{R} = S(\omega) R$$

Using this, and the product rule, we get

$$\dot{R} = \dot{R}_{z,\alpha} R_{y,\beta} R_{z,\gamma} + R_{z,\alpha} \dot{R}_{y,\beta} R_{z,\gamma} + R_{z,\alpha} R_{y,\beta} \dot{R}_{z,\gamma}$$

$$= S(\dot{\alpha} k) R_{z,\alpha} R_{y,\beta} R_{z,\gamma} + R_{z,\gamma} S(\dot{\beta} j) R_{y,\beta} R_{z,\gamma} + R_{z,\alpha} R_{y,\beta} S(\dot{\gamma} k) R_{z,\gamma}$$

Expressing the angular velocities in the correct frame (base frame) yields

$$\dot{R} = S(\dot{\alpha} k) R_{z,\alpha} R_{y,\beta} R_{z,\gamma} + S(R_{z,\alpha} \dot{\beta} j) R_{z,\alpha} R_{y,\beta} R_{z,\gamma} + S(R_{z,\alpha} R_{y,\beta} \dot{\gamma} k) R_{z,\alpha} R_{y,\beta} R_{z,\gamma}$$

$$= \left[S(\dot{\alpha} k) + S(R_{z,\alpha} \dot{\beta} j) + S(R_{z,\alpha} R_{y,\beta} \dot{\gamma} k) \right] R$$

Angular velocities expressed in the same frame can be added together, so

$$\dot{R} = S(\dot{\alpha} k + R_{z,\alpha} \dot{\beta} j + R_{z,\alpha} R_{y,\beta} \dot{\gamma} k) R$$

$$= S(\omega) R$$

$$= \hat{\omega} R$$

Thus

$$\hat{\omega} = S(\dot{\alpha} k + R_{z,\alpha} \dot{\beta} j + R_{z,\alpha} R_{y,\beta} \dot{\gamma} k)$$

where

$$\dot{\alpha} k = \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix}$$

$$R_{z,\alpha} \dot{\beta} j = \dot{\beta} \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -s\alpha \\ c\alpha \\ 0 \end{bmatrix} \dot{\beta}$$

$$R_{z,\alpha} R_{y,\beta} \dot{\gamma} k = \dot{\gamma} \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \dot{\gamma} \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s\beta \\ 0 \\ c\beta \end{bmatrix}$$

$$= \dot{\gamma} \begin{bmatrix} c\alpha\, s\beta \\ s\alpha\, s\beta \\ c\beta \end{bmatrix}$$

Then

$$\omega = \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} -\dot{\beta} s_\alpha \\ \dot{\beta} c_\alpha \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\gamma} c_\alpha s_\beta \\ \dot{\gamma} s_\gamma s_\beta \\ \dot{\gamma} c_\beta \end{bmatrix}$$

$$= \begin{bmatrix} \dot{\gamma} c_\alpha s_\beta - \dot{\beta} s_\alpha \\ \dot{\gamma} s_\gamma s_\beta + \dot{\beta} c_\alpha \\ \dot{\gamma} c_\beta + \dot{\alpha} \end{bmatrix}$$

$$= \begin{bmatrix} c_\alpha s_\beta & -s_\alpha & 0 \\ s_\gamma s_\beta & c_\alpha & 0 \\ c_\beta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix}$$

where

$$J_w = \begin{bmatrix} c_\alpha s_\beta & -s_\alpha & 0 \\ s_\alpha s_\beta & c_\alpha & 0 \\ c_\beta & 0 & 1 \end{bmatrix}$$

2b)

The Jacobian loses its rank if its determinant is zero.

Thus

$$\det(J_w) = c_\beta \begin{vmatrix} -s_\alpha & 0 \\ c_\alpha & 0 \end{vmatrix} + \begin{vmatrix} c_\alpha s_\beta & -s_\alpha \\ s_\alpha s_\beta & c_\alpha \end{vmatrix}$$

$$= c_\alpha^2 s_\beta + s_\alpha^2 s_\beta$$

$$= s_\beta$$

So the Jacobian loses its rank if

$$\sin(\beta) = 0$$

which implies

$$\beta = \pm n\pi \quad , \; n = 0, 1, 2, \ldots$$

Problem 3

The MATLAB code for this assignment is listed in lst 1.

3a)

By using forward kinematics, the homogeneous transformation matrices H_2^0 and H_e^0 were found. Since all the the joints are revolute, the translation velocity part of the Jacobian matrix of the end-effector is given by

$$J_v = \begin{bmatrix} z_0^0 \times (P_e^0 - P_0^0) & | & z_1^0 \times (P_e^0 - P_1^0) & | & z_2^0 \times (P_e^0 - P_2^0) \end{bmatrix} \tag{1}$$

The position vectors and axis vectors can be easily obtained by looking at the third and fourth column of the corresponding homogeneous transformation matrix. Performing the cross-products then yields

$$J_v = \begin{bmatrix} -L_3c_1c_{23} - L_2c_1c_2 & L_3s_1s_{23} + L_2s_1s_2 & L_3s_1s_{23} \\ -L_3s_1c_{23} - L_2s_1c_2 & -L_3c_1s_{23} - L_2c_1s_2 & -L_3c_1s_{23} \\ 0 & L_3c_{23} + L_2c_2 & L_3c_{23} \end{bmatrix} \tag{2}$$

3b)

Using MATLAB, the determinant was calculated to be

$$det(J_v) = -L_2L_3(L_2c_2s_3 - L_3s_3 + L_3c_3^2s_2 + L_3c_2c_3s_3) \tag{3}$$

Applying algebra and trigonometric identities, the determinant can be rewritten as

$$det(J_v) = -L_2L_3s_3(L_2c_2 + L_3c_{23}) \tag{4}$$

We can see from eq. (4) that the manipulator has singular position whenever

$$s_3 = 0 \quad \Longrightarrow \quad \theta_3 = 0 \ \vee \ \theta_3 = \pi \tag{5}$$

and whenever

$$L_2c_2 + L_3c_{23} = 0 \tag{6}$$

3c)

When J_v is rank deficient, we have that $N(J_v) \neq \phi$. As found in the previous task, the Jacobian is rank deficient in three different cases.

<u>Case 1</u>

This singular position is obtained when $\theta_3 = 0$. Using MATLAB, we obtain the following basis for the null space of the Jacobian

$$Null(J_v) = span\left\{ \begin{bmatrix} 0 \\ -2/5 \\ 1 \end{bmatrix} \right\} \tag{7}$$

In this configuration the manipulator has its third joint fully extended. It is therefore intuitive that no values of $\dot{\theta}_1$ lies in the null space of J_v. A rotation around the z_0-axis will impose a movement in the end-effector. The last two elements of the vector spanning the null space represents a linear relationship between $\dot{\theta}_2$ and $\dot{\theta}_3$. In order to keep the end-effector still, a decrease in θ_2 imposes an increase in θ_3. By looking at a drawing of the manipulator in this configuration, the linear relationship between the two angular velocities makes sense. The ratio between the two angular velocities is a consequence of non-equal link lengths.

Case 2

This singular position is obtained when $\theta_3 = \pi$. The basis for the null space of the Jacobian is in this case the following

$$Null(J_v) = span\left\{ \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} \right\} \tag{8}$$

In this configuration the manipulator has its third joint fully contracted. Equal to the previous case, a rotation around the z_0-axis will impose a movement in end-effector. Picturing the manipulator in this configuration, the linear relationship between $\dot{\theta}_2$ and $\dot{\theta}_3$ is sensible. If θ_2 increases, θ_3 must also increase in order to make end-effector still.

Case 3

This singular position is obtained when $L_2c_2 + L_3c_{23} = 0$. The expression can be rewritten into the following expression

$$\theta_3 = \cos^{-1}(-\frac{L_2}{L3}\cos\theta_2) - \theta_2 \tag{9}$$

The basis of the null space of Jacobian is in this case the following

$$Null(J_v) = span\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{10}$$

This singular configuration is shown in the textbook, where it is illustrated that the end-effector lies on the z_0-axis. Looking at the figure, it becomes clear that the only rotation that will not impose a movement in the end-effector is a rotation around the z_0-axis.

Code

Listing 1: Code for problem 3

```matlab
clc; clear; close all;

%% Parameters

syms theta1 theta2 theta3 L1 L2 L3 real;


%% Problem 3a

% Forward kinematics
H01 = DH_matrix(L1,theta1+sym(pi)/2,0,sym(pi)/2);
H12 = DH_matrix(0,theta2,L2,0);
H2e = DH_matrix(0,theta3,L3,0);

H02 = H01*H12;
H0e = H02*H2e;

% Extract position
P01 = H01(1:3,4);
P02 = H02(1:3,4);
P0e = H0e(1:3,4);

% Extract z-axis
z0 = [0 0 1]';
z1 = H01(1:3,3);
z2 = H02(1:3,3);
ze = H0e(1:3,3);

% Every joint is revolute
% Calculate translation velocity part of Jacobian matrix of end-effector
Jv = [cross(z0,P0e) cross(z1,(P0e-P01)) cross(z2,(P0e-P02))];


%% Problem 3b

% Calculate determinant
detJv = simplify(det(Jv));

% Applying further algebraic manipulation and trigonometric identities on the determinant gives
```

```
condition for
% singular configuration


%% Problem 3c

Jv = simplify(subs(Jv, [L1 L2 L3], [0.5 1.5 1]));

% Case 1, theta3 = 0
Jv1 = subs(Jv, theta3, 0);
nullsp1 = null(Jv1);

% Case 2, theta3 = pi
Jv2 = subs(Jv, theta3, pi);
nullsp2 = null(Jv2);

% Case 3, L2c2 + L3c23 = 0
% The expression can be rewritten as
% theta3 = acos(-L2/L3*cos(theta2))-theta2
Jv3 = subs(Jv, theta3, acos(-L2/L3*cos(theta2))-theta2);
Jv3 = simplify(subs(Jv3, [L1 L2 L3], [0.5 1.5 1]));
nullsp3 = null(Jv3);


%% Functions

function M = DH_matrix(d,theta,a,alpha) % Calculate Denavit-Hartenberg Matrix
    M = [cos(theta) -sin(theta)*cos(alpha) sin(theta)*sin(alpha) a*cos(theta); sin(theta) cos(theta)*cos(alpha) -cos(theta)*sin(alpha) a*sin(theta); 0 sin(alpha) cos(alpha) d; 0 0 0 1];
end
```