ME453 - Homework 2

Herman Kolstad Jakobsen
20196493

October 13, 2019

ME453 Homework 2 Herman K. Jakobsen

20196493

Problem 1-1)

The euclidean norm is defined as

$$\|x\| = \sqrt{x^T x}$$

Hence

$$\|Rx - Ry\| = \sqrt{(Rx - Ry)^T (Rx - Ry)}$$

$$= \sqrt{(x^T R^T - y^T R^T)(Rx - Ry)}$$

$$= \sqrt{x^T \underbrace{R^T R}_{=I} x - x^T R^T R y - y^T R^T R x + y^T R^T R y}$$

$$= \sqrt{x^T x - x^T y - y^T x + y^T y}$$

$$= \sqrt{(x-y)^T (x-y)}$$

$$= \|x - y\|$$

So

$$\|Rx - Ry\| = \|x - y\|$$

The standard Euclidean inner product is defined as

$$\langle x, y \rangle = x^T y$$

Hence

$$\langle Rx - Rz , Ry - Rz \rangle = (Rx - Rz)^T (Ry - Rz)$$

$$= (x^T R^T - z^T R^T)(Ry - Rz)$$

$$= x^T R^T R y - x^T R^T R z - z^T R^T R y + z^T R^T R z$$

$$= x^T y - x^T z - z^T y + z^T z$$

$$= (x - z)^T (y - z)$$

$$= \langle x - z , y - z \rangle$$

Scanned by CamScanner

Problem 1-2)

Need to show that

$$AB = \left\{ AB \in \mathbb{R}^{3\times3} \,\middle|\, (AB)^T (AB) = I, \ \det(AB) = 1 \right\}$$

Since both $A \in \mathbb{R}^{3\times3}$ and $B \in \mathbb{R}^{3\times3}$, we have that

$$\underline{AB \in \mathbb{R}^{3\times3}}$$

Further, the determinant is given as

$$\det(AB) = \det(A)\det(B) = 1 \cdot 1 = \underline{1}$$

Lastly,

$$(AB)^T (AB) = B^T \underbrace{A^T A}_{=I} B = B^T B = \underline{I}$$

Hence,

$$\underline{\underline{AB \in SO(3)}}$$

Scanned by CamScanner

Problem 1-3)

Defining

$$r^a = R_b^a \, r^b \qquad (I)$$

$$r^b = R_c^b \, r^c \qquad (II)$$

$$r^a = R_c^a \, r^c \qquad (III)$$

Inserting (II) into (I) yields

$$r^a = R_b^a R_c^b \, r^c \qquad (IV)$$

By comparing (III) and (IV), we can conclude that

$$\underline{\underline{R_c^a = R_b^a R_c^b}}$$

Problem 1-4)

Know that post-multiplication of a rotation matrix yields rotation around current frame. Hence

$$R_b^s \, R$$

will yield a rotation, given by R, around the current frame {b}.

Further

$$R_b^s \, P$$

will result in a translation, P, with respect to the current frame {b}.

Ultimately, the new transformation H, will transform with respect to the current frame {b}.

Pre-multiplication of a rotation matrix yields a rotation around a fixed frame.

Scanned by CamScanner

So

$$RR^s_b$$

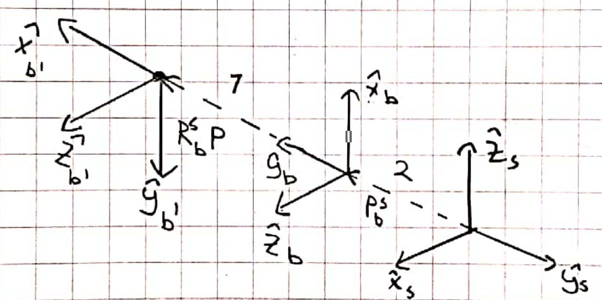
will give a rotation, R, around the fixed frame $\{s\}$,

Further

$$P$$

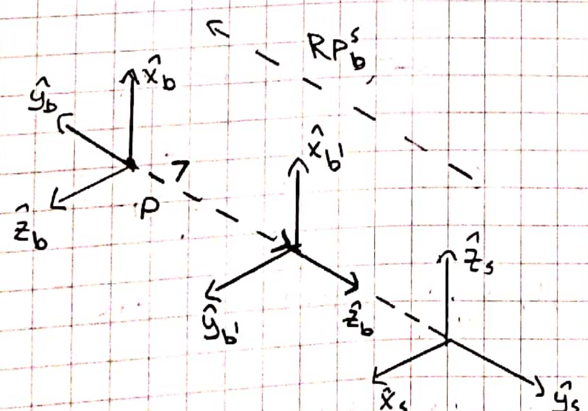
will be a translation, P, with respect to the current frame $\{s\}$.
RP^s_b is the translation P^s_b with respect to the current frame $\{s\}$.

For $H^s_b H$:



Translate 7 along $\hat{y}_b$ and rotate 90° around $\hat{z}_b$.

For $H H^s_b$:



Translate 7 along $\hat{y}_s$ and rotate 90° around $\hat{z}_s$.

Pre-multiplication $\Rightarrow$ Transformation with respect to fixed frame

Post-multiplication $\Rightarrow$ Transformation with respect to current frame

Scanned by CamScanner

Problem 2a)

For relative position and orientation between end-effector and object, we have

$$H_o^e = H_b^e H_c^b H_o^c = \underline{\underline{(H_e^b)^{-1} H_c^b H_o^c}}$$

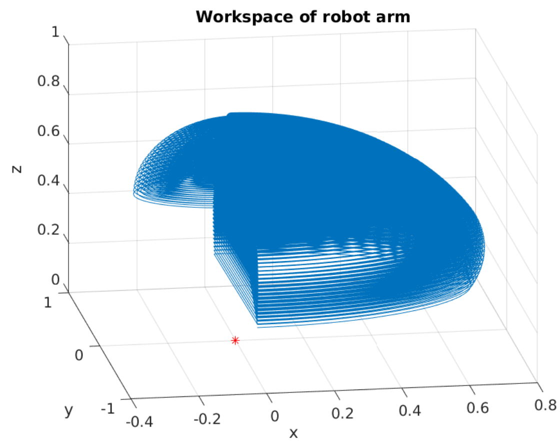
For the global location and orientation of the object, we have

$$H_o^w = \underline{\underline{H_b^w H_c^b H_o^c}}$$

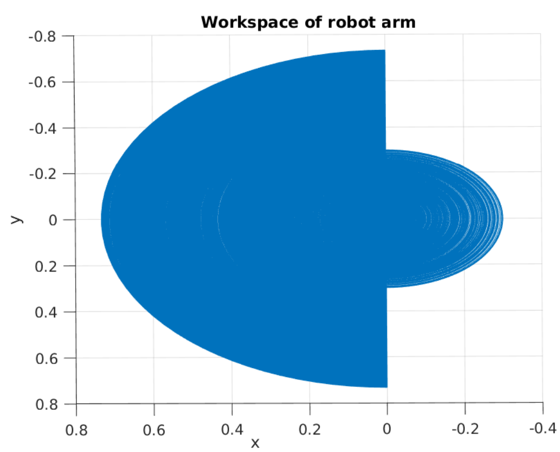
Scanned by CamScanner

Problem 2b

The workspace of the robot arm is shown in fig. 1.



(a) 3D representation of workspace of the robot arm



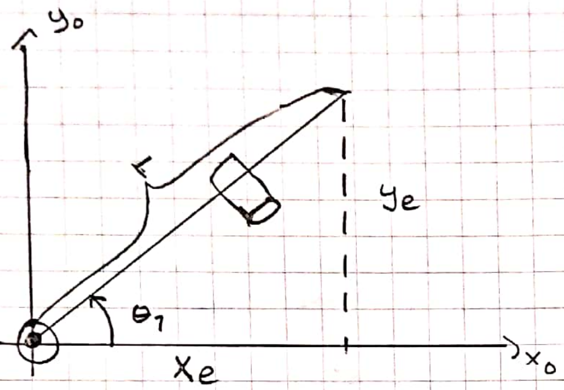
(b) Workspace of robot arm projected onto x-y plane

Figure 1: Workspace of robot arm

The code used to obtain and plot the workspace is implemented in lst 1.

2c)

Drawing the arm onto $x_0 - y_0$ plane yields the following



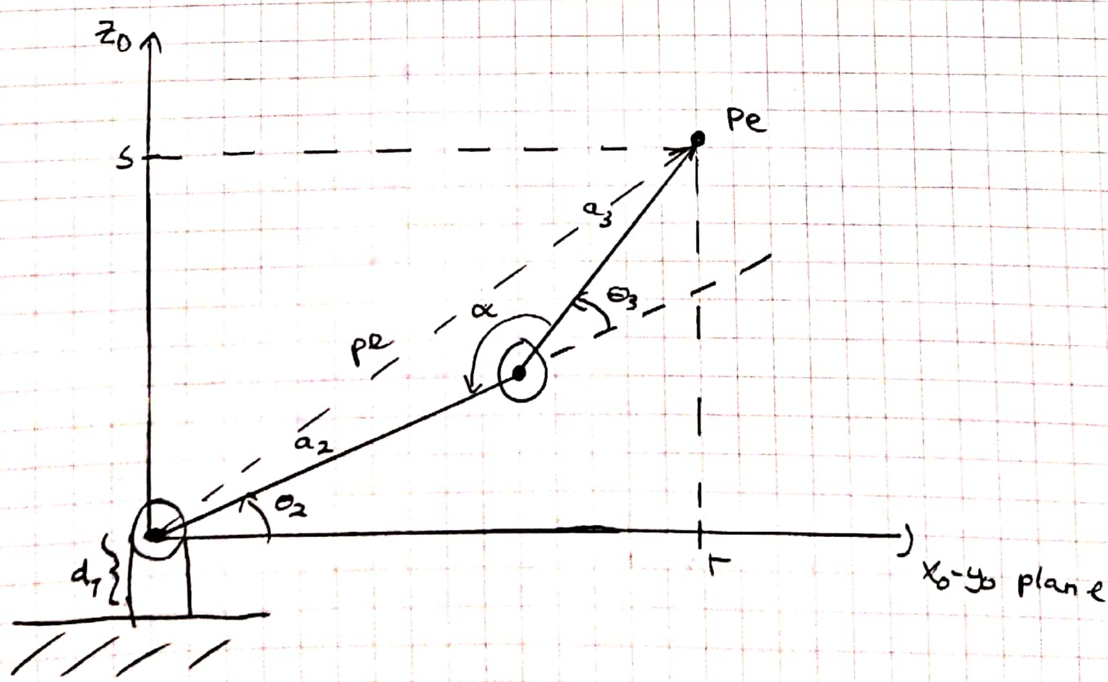
From the projection, we get

$$\tan(\theta_1) = \frac{y_e}{x_e}$$

This gives two valid solutions

$$\theta_1 = \operatorname{atan2}(y_e, x_e) \quad \vee \quad \theta_1 = \pi + \operatorname{atan2}(y_e, x_e)$$

Projecting onto the plane formed by link 2 and 3 gives the following drawing



This mechanism can now be considered as a planar two-link system.

Scanned by CamScanner

Using Pythagoras' theorem we obtain

$$p_e^2 = r^2 + s^2$$

Applying the cosine rule $\left(A^2 = B^2 + C^2 - 2BC\cos(a)\right)$ yields

$$p_e^2 = a_2^2 + a_3^2 - 2a_2a_3\cos\alpha$$

$$\Rightarrow \cos\alpha = \frac{a_2^2 + a_3^2 - p_e^2}{2a_2a_3}$$

$$= \frac{a_2^2 + a_3^2 - (r^2 + s^2)}{2a_2a_3} \quad , \quad r = \sqrt{x_e^2 + y_e^2}$$

$$= \frac{a_2^2 + a_3^2 - (x_e^2 + y_e^2 + s^2)}{2a_2a_3} \quad , \quad s = z_e - d_1$$

$$= \frac{a_2^2 + a_3^2 - \left(x_e^2 + y_e^2 + (z_e - d_1)^2\right)}{2a_2a_3}$$

Further

$$\theta_3 = \pi - \alpha$$

So

$$\cos\alpha = \cos(\pi - \theta_3) = -\cos(\theta_3)$$

Hence

$$\cos\theta_3 = \frac{x_e^2 + y_e^2 + (z_e - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3}$$

A known identity is

$$\cos^2\theta_3 + \sin^2\theta_3 = 1 \quad \Rightarrow \quad \sin\theta_3 = \pm\sqrt{1 - \cos^2\theta_3}$$

Scanned by CamScanner

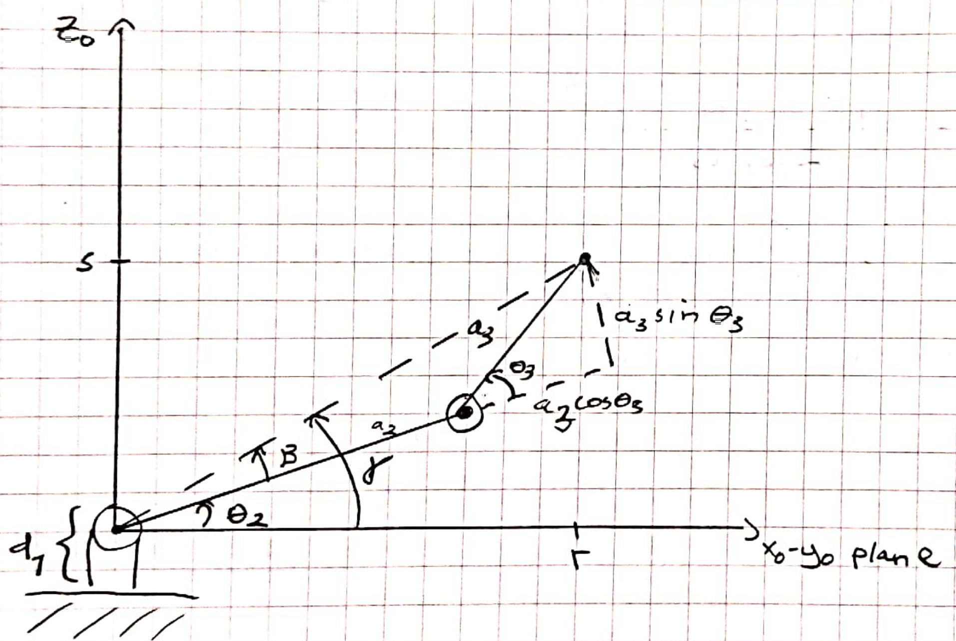
This means that

$$\tan\theta_3 = \frac{\sin\theta_3}{\cos\theta_3}$$

$$\Rightarrow \quad \theta_3 = \text{atan}2(\sin\theta_3 \;,\; \cos\theta_3)$$

$$\underline{\theta_3 = \text{atan}2\left(\pm\sqrt{1-\cos^2\theta_3}\;,\; \cos\theta_3\right)}$$

Making a new drawing to find θ_2



So

$$\tan\beta = \frac{a_3 \sin\theta_3}{a_2 + a_3\cos\theta_3} \quad \Rightarrow \quad \beta = \text{atan}2(a_3\sin\theta_3 \;,\; a_2 + a_3\cos\theta_3)$$

and

$$\tan\gamma = \frac{s}{r} \quad \Rightarrow \quad \gamma = \text{atan}2(s, r)$$

Ultimately

$$\underline{\theta_2 = \gamma - \beta = \text{atan}2(s, r) - \text{atan}2(a_3\sin\theta_3 \;,\; a_2 + a_3\cos\theta_3)}$$

Scanned by CamScanner

To summarize :

$p_e^b = [x_e \quad y_e \quad z_e]^T$ is given

The joint angles are then :

$$\theta_1 = \text{atan2}(y_e, x_e) \quad \vee \quad \theta_1 = \pi + \text{atan2}(y_e, x_e)$$

$$\theta_2 = \text{atan2}\left(z_e - d_1, \sqrt{x_e^2 + y_e^2}\right) - \text{atan2}(a_3 \sin\theta_3, a_2 + a_3\cos\theta_3)$$

$$\theta_3 = \text{atan2}\left(\pm\sqrt{1-\cos^2\theta_3}, \cos\theta_3\right)$$

where

$$\cos\theta_3 = \frac{x_e^2 + y_e^2 + (z_e - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3}$$

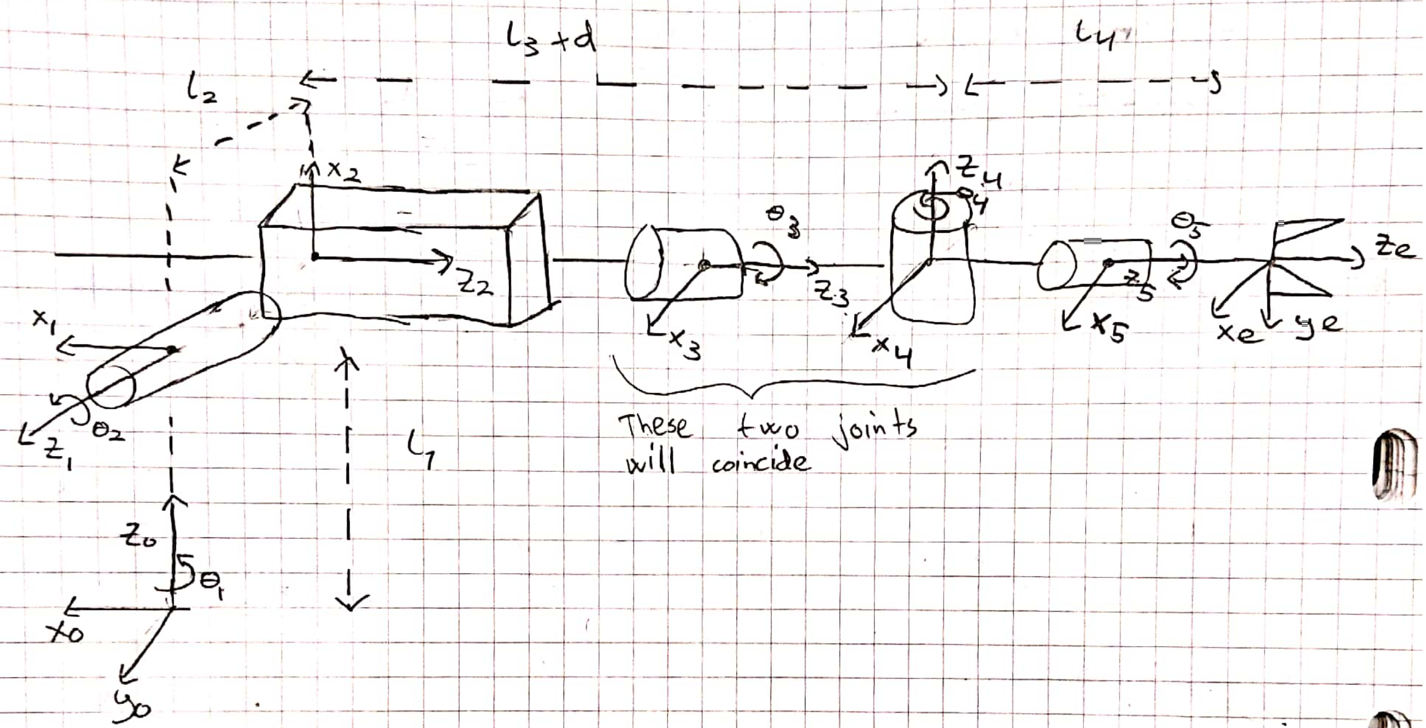
Based on this, we have $\underline{4}$ different solutions.

Scanned by CamScanner

Problem 3

The MATLAB code for this assignment is listed in lst 2.

Problem 3a)



D-H parameters

Link	a_i	α_i	d_i	θ_i
0	0	$-\frac{\pi}{2}$	l_1	θ_1
1	0	$\frac{\pi}{2}$	$-l_2$	$\theta_2 - \frac{\pi}{2}$
2	0	0	l_3+d	$\frac{\pi}{2}$
3	0	$\frac{\pi}{2}$	0	θ_3
4	0	$-\frac{\pi}{2}$	0	θ_4
5	0	0	l_4	θ_5

Note that the drawing is a zero-configuration. Hence, the D-H parameters have been obtained while assuming $\theta_i = 0$.

The H_e^0 matrix is implemented in the MATLAB code.

Given by $H_e^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4 H_e^5$

Scanned by CamScanner

Problem 3a

The $\boldsymbol{H_e^0}$ matrix was quite extensive, so it will not be stated. The transformation matrix can be displayed by running the MATLAB code in lst 2.

Problem 3b

The position and orientation was found to be the following

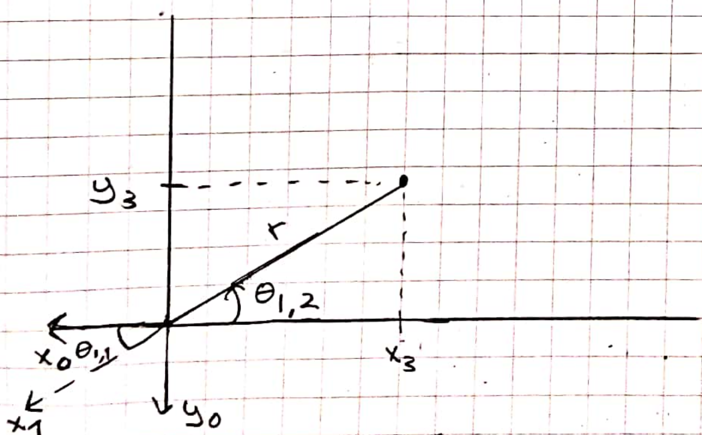
$$\boldsymbol{r_{obj}^0} = \begin{bmatrix} -2.1651 \\ -0.1472 \\ 1.5165 \end{bmatrix} \quad \text{and} \quad \boldsymbol{R_{obj}^0} = \begin{bmatrix} -0.0217 & -0.3201 & -0.9527 \\ 0.3257 & -0.9028 & 0.2880 \\ -0.9429 & -0.3026 & 0.1337 \end{bmatrix} \tag{1}$$

3c)

Assuming the end-effector is reaching the object, so

$$r_3^0 = r_{obj}^0 - c_4 R_{obj}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = r_{obj}^0 - \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} \overset{\text{MATLAB}}{=} \begin{bmatrix} -1,2123 \\ -0,4352 \\ 1,3828 \end{bmatrix}$$

Drawing the projection of the first link onto the x-y plane yields



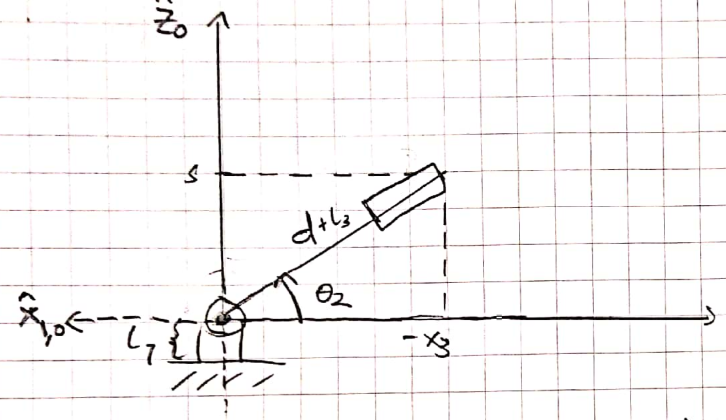
There are two valid solutions

$\theta_{1,1} = \text{atan2}(y_3, x_3)$ $\vee$ $\theta_{1,2} = \text{atan2}(-y_3, -x_3) = \pi + \text{atan2}(y_3, x_3)$

Will choose the second one, since it is closer to the orignal configuration.

Scanned by CamScanner

Drawing the projections of link 2 and link 3 gives



Note l_2 is zero. So no offset into the plane, which makes things easier.

From the figure, the chosen D-H parameters and the defined axes, we get

$$d+l_3 = \sqrt{x_3^2 + s^2}$$

$$d = \sqrt{x_3^2 + (z_3 - l_7)^2} - l_3$$

and

$$\theta_2 = \text{atan2}(s, -x_3) = \text{atan2}(z_3 - l_7, -x_3)$$

To summarize:

$$\theta_1 = \text{atan2}(y_3, x_3) + \pi$$

$$\theta_2 = \text{atan2}(z_3 - l_7, -x_3)$$

$$d = \sqrt{x_3^2 + y_3^2 + (z_3 - l_7)^2} - l_3$$

Scanned by CamScanner

3d)

From 3c), θ_1, θ_2 and d are now known. Hence

$$R_3^0 = R_1^0 R_2^1 R_3^2$$

is also known. Further

$$R_{obj}^0 = R_3^0 R_{obj}^3$$

$$\Rightarrow R_{obj}^3 = (R_3^0)^T R_{obj}^0$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Implementing the forward kinematics in MATLAB, we obtain

$$R_e^3 = R_{obj}^3 = R_4^3 R_5^4 R_e^5 = \begin{bmatrix} c_4 c_5 c_3 - s_5 s_3 & -c_5 s_3 - c_4 s_5 c_3 & -s_4 c_3 \\ s_5 c_3 + c_4 c_5 s_3 & c_5 c_3 - c_4 s_5 s_3 & -s_4 s_3 \\ c_5 s_4 & -s_4 s_5 & c_4 \end{bmatrix}$$

Comparing the two matrices yields

$$c_4 = r_{33}$$

and

$$r_{13}^2 + r_{23}^2 = s_4^2 c_3^2 + s_4^2 s_3^2 = s_4^2$$

$$\Rightarrow s_4 = \pm\sqrt{r_{13}^2 + r_{23}^2}$$

Hence

$$\underline{\theta_4 = \text{atan2}\left(\pm\sqrt{r_{13}^2 + r_{23}^2},\ r_{33}\right)}$$

Scanned by CamScanner

Will then have two cases. Further comparison yields

i)

$$s_4 = \sqrt{r_{13}^2 + r_{23}^2}$$

$$\underline{\theta_4 = \text{atan2}\left(\sqrt{r_{13}^2 + r_{23}^2},\ r_{23}\right)}$$

$$\tan\theta_5 = \frac{-(-s_4 s_5)}{s_4 c_5} = \frac{s_5}{c_5} = \frac{-r_{32}}{r_{31}}$$

$$\Rightarrow \underline{\theta_5 = \text{atan2}(-r_{32},\ r_{31})}$$

$$\underline{\theta_3 = \text{atan2}(r_{23},\ r_{13})}$$

ii)

$$s_4 = -\sqrt{r_{13}^2 + r_{23}^2}$$

$$\underline{\theta_4 = \text{atan2}\left(-\sqrt{r_{13}^2 + r_{23}^2},\ r_{23}\right)}$$

$$\underline{\theta_5 = \text{atan2}(r_{32},\ -r_{31})}$$

$$\underline{\theta_3 = \text{atan2}(-r_{23},\ -r_{13})}$$

Scanned by CamScanner

Problem 3e

The inverse kinematics was not done correctly. Using the parameters from case i) in problem 3c), together with the parameters from problem 3b, the inverse kinematics resulted in a position with huge deviations from $\boldsymbol{r^0_{obj}}$. Moreover, the obtained rotation matrix had no similarities with $\boldsymbol{R^0_{obj}}$. The obtained position and orientation was

$$\boldsymbol{r^0_{obj}}(\boldsymbol{q_{des}}) = \begin{bmatrix} 2.0922 \\ 0.1277 \\ 1.2569 \end{bmatrix} \neq \boldsymbol{r^0_{obj}} = \begin{bmatrix} -2.1651 \\ -0.1472 \\ 1.5165 \end{bmatrix}$$

and

$$\boldsymbol{R^0_{obj}}(\boldsymbol{q_{des}}) = \begin{bmatrix} -0.2846 & -0.1198 & 0.9511 \\ -0.9587 & 0.0385 & -0.2820 \\ -0.0028 & -0.9920 & -0.1258 \end{bmatrix} \neq \boldsymbol{R^0_{obj}} = \begin{bmatrix} -0.0217 & -0.3201 & -0.9527 \\ 0.3257 & -0.9028 & 0.2880 \\ -0.9429 & -0.3026 & 0.1337 \end{bmatrix}$$

Code

Listing 1: Code for obtaining and plotting workspace of robot arm

```
clc; clear; close all;
%% Define parameters and constants

% Link 1
a1 = 0;
alpha1 = pi/2;
d1 = 0.1;

% Link 2
a2 = 0.5;
alpha2 = 0;
d2 = 0;

% Link 3
a3 = 0.3;
alpha3 = 0;
d3 = 0;

% Joint angles
theta1 = linspace(-pi/2,pi/2,50); % theta1 range from -pi
    /2 to pi/2 and divide into 50 units
theta2 = linspace(pi/6, pi/2, 50);
theta3 = linspace(-pi/3, pi/2, 50);

%% Forward kinematics

% Define rotation matrices and translation matrices
syms alpha;
syms theta;
syms a;
syms d;

Rx = [1 0 0 0; 0 cos(alpha) -sin(alpha) 0; 0 sin(alpha)
    cos(alpha) 0; 0 0 0 1];
Rz = [cos(theta) -sin(theta) 0 0; sin(theta) cos(theta) 0
     0; 0 0 1 0; 0 0 0 1];

Tx = [1 0 0 a; 0 1 0 0; 0 0 1 0; 0 0 0 1];
Tz = [1 0 0 0; 0 1 0 0; 0 0 1 d; 0 0 0 1];

% Homogeneous transformation
H = Tz*Rz*Tx*Rx;
```

```
syms theta01;
syms theta12;
syms theta23;

H01 = subs(H, [alpha a theta d], [alpha1 a1 theta01 d1]);
H12 = subs(H, [alpha a theta d], [alpha2 a2 theta12 d2]);
H23 = subs(H, [alpha a theta d], [alpha3 a3 theta23 d3]);

H03= H01*H12*H23; % Position and orientation of frame 3
    with respect to frame 0
R03 = H03(1:3, 1:3);
p03 = H03(1:3, 4);

% Coordinates of end-effector with respect to frame 0
x = p03(1);
% x = (cos(theta01)*cos(theta12))/2 - (3*cos(theta01)*sin
    (theta12)*sin(theta23))/10 + (3*cos(theta01)*cos(
    theta12)*cos(theta23))/10
y = p03(2);
% y = (cos(theta12)*sin(theta01))/2 - (3*sin(theta01)*sin
    (theta12)*sin(theta23))/10 + (3*cos(theta12)*cos(
    theta23)*sin(theta01))/10
z = p03(3);
% z = sin(theta12)/2 + (3*cos(theta12)*sin(theta23))/10 +
     (3*cos(theta23)*sin(theta12))/10 + 1/10


%% Plot workspace

% Obtain all the different x-y-z coordinates
[THETA1,THETA2,THETA3]=ndgrid(theta1,theta2,theta3);
xM = (cos(THETA1).*cos(THETA2))/2-(3*cos(THETA1).*sin(
    THETA2).*sin(THETA3))/10 + (3*cos(THETA1).*cos(THETA2)
    .*cos(THETA3))/10;
yM = (cos(THETA2).*sin(THETA1))/2 - (3*sin(THETA1).*sin(
    THETA2).*sin(THETA3))/10 + (3*cos(THETA2).*cos(THETA3)
    .*sin(THETA1))/10;
zM = sin(THETA2)/2 + (3*cos(THETA2).*sin(THETA3))/10 +
    (3*cos(THETA3).*sin(THETA2))/10 + 1/10;

plot3(xM(:),yM(:),zM(:));
hold on;
xlabel('x');
ylabel('y');
zlabel('z');
```

```
title('Workspace of robot arm');
plot3(0,0,0, '*', 'color','r');
grid on;


%% Test 2c - Inverse kinematics

% Define an arbitrary point inside the arm's workspace
xe = 0.2;
ye = 0.1;
ze = 0.5;

% Calculate the joint angles
c3 = (xe^2+ye^2+(ze-d1)^2-a2^2-a3^2)/(2*a2*a3);
t1 = atan2(ye, xe);
t3 = atan2(sqrt(1-c3^2), c3);
t2 = atan2(ze-d1, sqrt(xe^2+ye^2)) - atan2(a3*sin(t3), a2+a3*cos(t3));

% Calculate the coordinates using forward kinematics with the newly obtained angles
xcalc = (cos(t1).*cos(t2))/2-(3*cos(t1).*sin(t2).*sin(t3))/10 + (3*cos(t1).*cos(t2).*cos(t3))/10;
ycalc = (cos(t2).*sin(t1))/2 - (3*sin(t1).*sin(t2).*sin(t3))/10 + (3*cos(t2).*cos(t3).*sin(t1))/10;
zcalc = sin(t2)/2 + (3*cos(t2).*sin(t3))/10 + (3*cos(t3).*sin(t2))/10 + 1/10;

% Check if calculated coordinates equal defined coordinates
testCase = matlab.unittest.TestCase.forInteractiveUse;
assertEqual(testCase,xe,round(xcalc, 10)); % Using round() because of precision loss in calculations
assertEqual(testCase,ye,round(ycalc, 10));
assertEqual(testCase,ze,round(zcalc, 10));
```

Listing 2: Code for problem 3

```
clc; clear; close all;
%% Define parameters and constants

% Given vectors and matrices
rcobj = [0.10 0.05 0.30]';
Rcobj = [0.98 0.17 -0.09; -0.17 0.99 0.02; 0.08 0 1];
Hcobj = [Rcobj, rcobj; 0 0 0 1];

qnow = [30*pi/180 45*pi/180 0.5 30*pi/180 -60*pi/180 30*
    pi/180]';
qtest = [0 0 0 0 0 0];

% Constants
l1 = 0.5;
l2 = 0.0;
l3 = 1.0;
l4 = 1.0;


%% 3a - Forward Kinematics

% Define rotation matrices and translation matrices
syms alpha;
syms theta;
syms a;
syms zTrans;

Rx = [1 0 0 0; 0 cos(alpha) -sin(alpha) 0; 0 sin(alpha)
    cos(alpha) 0; 0 0 0 1];
Rz = [cos(theta) -sin(theta) 0 0; sin(theta) cos(theta) 0
    0; 0 0 1 0; 0 0 0 1];

Tx = [1 0 0 a; 0 1 0 0; 0 0 1 0; 0 0 0 1];
Tz = [1 0 0 0; 0 1 0 0; 0 0 1 zTrans; 0 0 0 1];

% Homogeneous transformation
H = Tz*Rz*Tx*Rx;

% DH-parameters
syms theta1;
syms theta2;
syms theta3;
syms theta4;
syms theta5;
```

```
syms d;

% Revolute joint, link {0}
alpha1 = -pi/2;
a1 = 0;
theta01 = theta1;
d1 = l1;

% Revolute joint, link {1}
alpha2 = pi/2;
a2 = 0;
theta12 = theta2 - pi/2;
d2 = -l2;

% Prismatic joint, link {2}
alpha3 = 0;
a3 = 0;
theta23 = pi/2;
d3 = l3+d;

% Revolute joint, link {3}
alpha4 = pi/2;
a4 = 0;
theta34 = theta3;
d4 = 0;

% Revolute joint, link {4}
alpha5 = -pi/2;
a5 = 0;
theta45 = theta4;
d5 = 0;

% Revolute joint, link {5}
alpha6 = 0;
a6 = 0;
theta56 = theta5;
d6 = l4;

H01 = subs(H, [alpha a theta zTrans], [alpha1 a1 theta01
    d1]);
H12 = subs(H, [alpha a theta zTrans], [alpha2 a2 theta12
    d2]);
H23 = subs(H, [alpha a theta zTrans], [alpha3 a3 theta23
    d3]);
H34 = subs(H, [alpha a theta zTrans], [alpha4 a4 theta34
    d4]);
```

```
H45 = subs(H, [alpha a theta zTrans], [alpha5 a5 theta45
    d5]);
H5e = subs(H, [alpha a theta zTrans], [alpha6 a6 theta56
    d6]);

H0e= H01*H12*H23*H34*H45*H5e;


%% 3b - Obtain r0obj and R0obj

% Given that frame {e} has the same position as frame {c
    }.
% Hence, orientation of frame {c} depends on frame {e}
    and will therefore
% assume {e}={c}.

% The transformation from frame {c} to frame {obj} is
    only valid when the
% parameters are equal to the parameters given in qnow.
% So must set the parameters in R0c and r0c equal to the
    parameters in
% qnow.

% Finding homogeneous transformation
H0c = H0e;
H0obj= H0c*Hcobj;

% Insert parameters
H0obj = subs(H0obj,[theta1 theta2 d theta3 theta4 theta5
    ], [qnow(1) qnow(2) qnow(3) qnow(4) qnow(5) qnow(6)]);

% Extracting rotation matrix and position vector
R0obj = H0obj(1:3, 1:3);
r0obj = H0obj(1:3, 4);


%% 3c - Inverse kinematics

% Obtain r03. Note that position of frame {e} is equal to
     position of frame
% {obj}

r03 = r0obj-l4*R0obj*[0 0 1]';

x03 = r03(1);
y03 = r03(2);
```

```
z03 = r03(3);

% Obtain parameters from inverse kinematics
t1 = atan2(y03, x03)+pi;
t2 = atan(z03-l1, -x03);
D = sqrt(x03^2+(z03-l1)^2)-l3;


%% 3d - Inverse kinematics
% Assuming that theta1, theta2 and d are given (found in
    previous
% task)
H03 = H01*H12*H23;
H03 = subs(H03, [theta1 theta2 d], [t1 t2 D]);

R03 = H03(1:3, 1:3);
R3obj = (R03')*R0obj;

r13 = R3obj(1,3);
r23 = R3obj(2,3);
r31 = R3obj(3,1);
r32 = R3obj(3,2);
r33 = R3obj(3,3);

% Case 1
t3_1 = atan2(r23, r13);
t4_1 = atan2(sqrt(r13^2+r23^2), r23);
t5_1 = atan2(-r32, r31);

% Case 2
t3_2 = atan2(-r23, -r13);
t4_2 = atan2(-sqrt(r13^2+r23^2), r23);
t5_2 = atan2(r32, -r31);


%% 3e - Comparison
Htest = subs(H0e, [theta1 theta2 d theta3 theta4 theta5],
     [t1 t2 D t3_1 t4_1 t5_1]);
Rtest = Htest(1:3, 1:3);
rtest = Htest(1:3, 4);

disp('Actual: ');
disp(double(r0obj));
disp(double(R0obj));

disp('From inverse kinematics: ');
```

```
disp(double(rtest));
disp(double(Rtest));
```