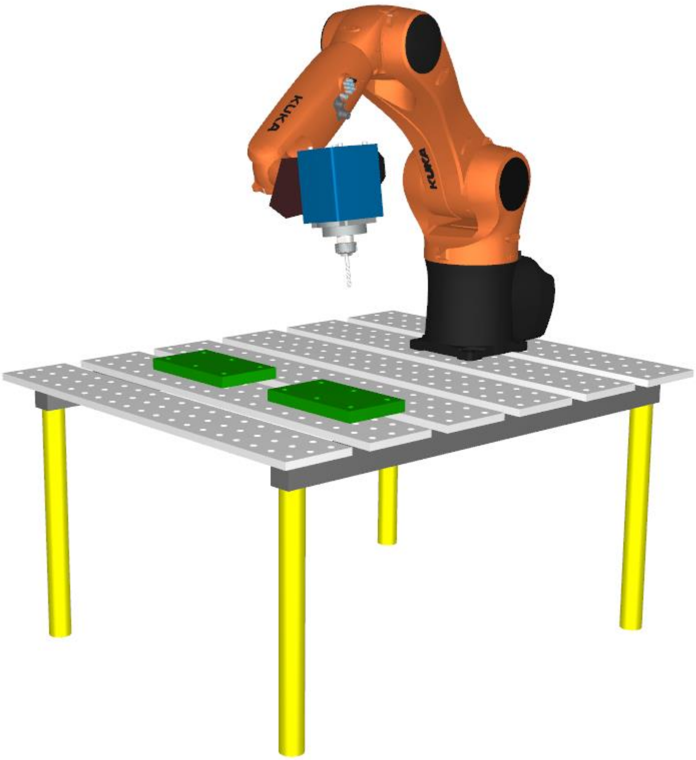
2022



Anvendt Robotteknikk 2022 Høst

EKSAMEN - FASIT

HARIS JASAREVIC

Oppgave 1

1. Rotasjonsmatriser er ikke kommutativ fordi 2 forskjellige rotasjoner etter hverandre vil ikke nødvendigvis produsere samme resultat dersom de er utført i motsatt rekkefølge.

2. Fordelen med homogene transformasjonsmatriser er at de samler både rotasjon og translasjon inn i en og samme matrise. Dette reduserer kompleksiteten og sørger for at translasjon og rotasjon er ivaretatt i en og samme matrise.

3. Fordelen med DH-konvensjonen er at den representerer forholdet mellom 2 ledd ved hjelp av kun 4 parametere i motsettning uten DH hvor det blir totalt 6. Dette forenkler matematikken og prosesseringen av roboter med mange ledd.

Oppgave 2

1. «Axis-Angle» representasjonen går ut på å beskrive en eller en sekvens av rotasjoner ved kun en vektor og vinkel.

2. Vektoren og vinkelen kan man hente ut fra en rotasjonsmatrise ved hjelp av egenverdier og egenvektorer. En ekte ortonormal matrise vil alltid ha en egenverdi lik 1, hvor dens tilsvarende vektor vil være uendret av transformasjonen. Det er denne vektoren hvor rotasjonen foregår rundt.

3. I motsetning til «Axis-Angle» beskrevet i punkt 1 brukes Euler konvensjonen til å beskrive rotasjonen til et legeme eller koordinatsystem ved hjelp av 3 etterfølgende rotasjoner, hvor rotasjoner rundt samme akse ikke er tatt rett etter hverandre.

Oppgave 3

Svar:

5, 2, 1, 3, 4

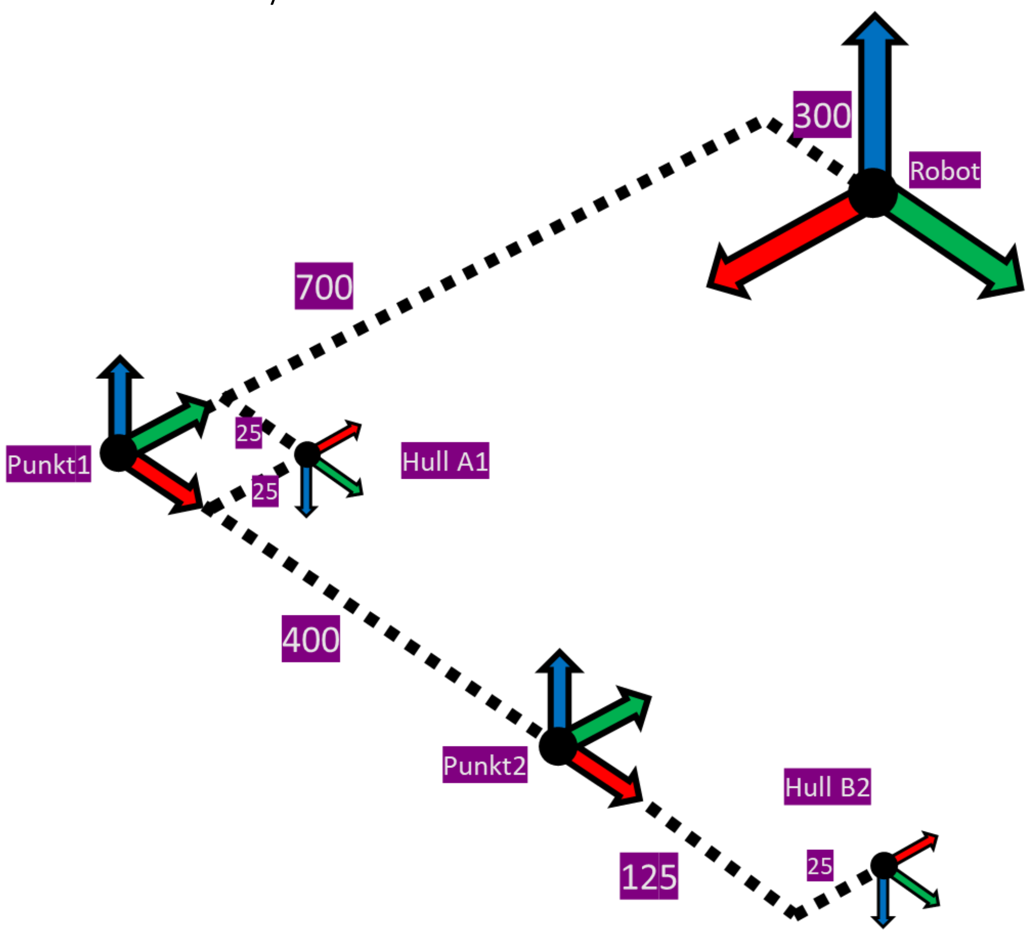
Oppgave 4

1. End-Effector.
2. TCP står for Tool-Center-Point og er punktet på robot-verktøyet som brukes til å definere banene til robot beveger seg i. Punktet brukes på selve kinematikken til roboten.
3.

$$(H_{bord}^{base})^T H_{TCP}^{base} = H_{base}^{bord} H_{TCP}^{base} = H_{TCP}^{bord}$$

Oppgave 5

1. Tegner inn alle koordinatsystemene:



2. Matrisene blir:

$$R_{Punkt1}^{Rob} = \begin{bmatrix} 0 & -1 & 0 & 700 \\ 1 & 0 & 0 & -300 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{Punkt2}^{Punkt1} = \begin{bmatrix} 1 & 0 & 0 & 400 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

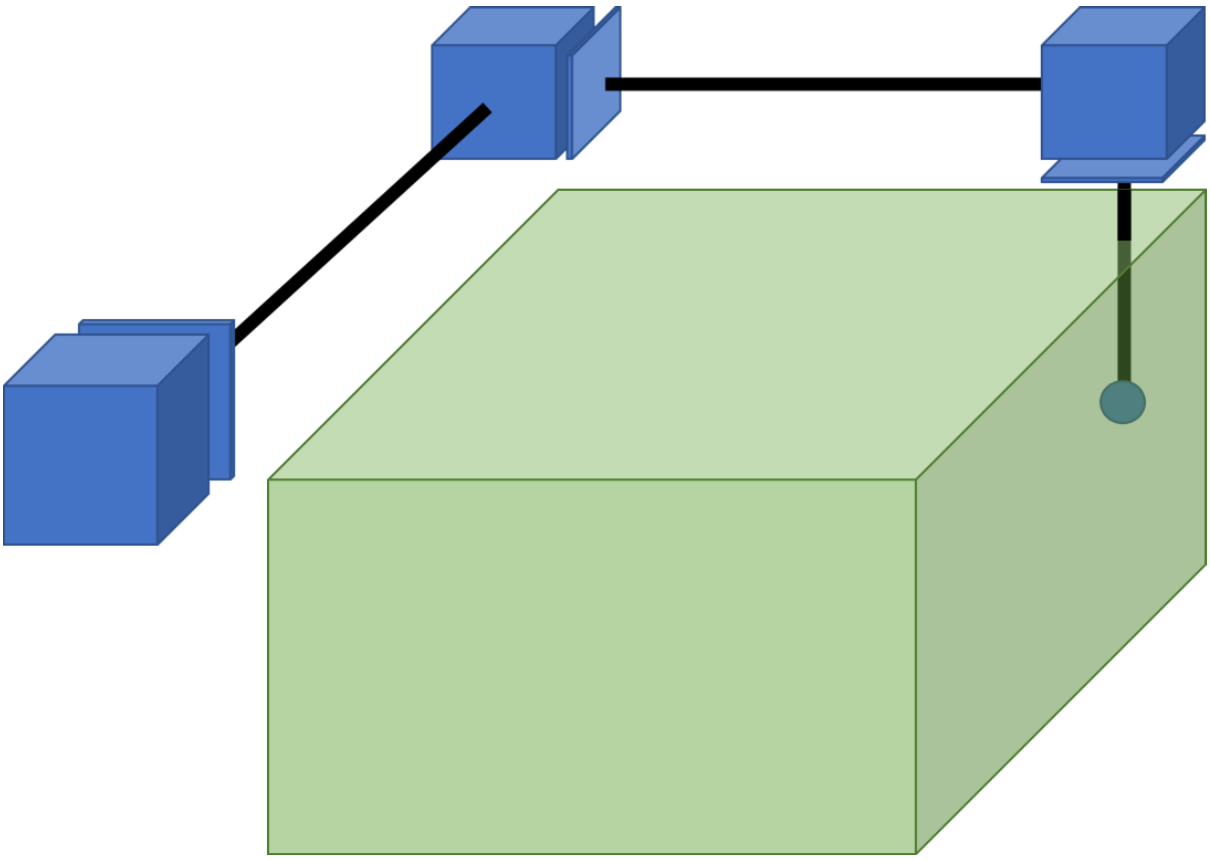
$$R_{Hull\_B1}^{Rob} = \begin{bmatrix} -1 & 0 & 0 & 675 \\ 0 & 1 & 0 & -175 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{Hull\_D2}^{Rob} = \begin{bmatrix} -1 & 0 & 0 & 575 \\ 0 & 1 & 0 & 125 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Fullfører koden:

```python
from robolink import *    # RoboDK API
from robodk import *      # Robot toolbox
sim = Robolink()
robot = sim.Item('KUKA KR 10 R900 sixx')  # The robot to use

RobotRamme = sim.Item("Base")
Punkt1Ramme = sim.Item("Plate 1")
Punkt2Ramme = sim.Item("Plate 2")

#--------Robot Ramme Aktiv----------
robot.setPoseFrame(RobotRamme) #Roboten går ut ifra sin egen base
home = sim.Item('Home')
robot.MoveJ(home)
#--------Plate 1 Ramme Aktiv-------
robot.setPoseFrame(Punkt1Ramme)
A1Upp = sim.Item("AppDrilling 1") #Over A1 punktet før drilling
A1Down = A1Upp.Pose()*transl(0,0,20) #A1 punktet etter drilling
robot.MoveL(A1Upp)
robot.MoveL(A1Down)
robot.MoveL(A1Upp)

for i in range(2) :
    if i == 0:
        robot.setPoseFrame(Punkt1Ramme)
    elif i == 1:
        robot.setPoseFrame(Punkt2Ramme)
    for j in range(2) :
        for k in range(3) :
            XXUpp = A1Upp.Pose()*transl(100*j,100*k,0)
            XXDown = XXUpp*transl(0,0,20)
            robot.MoveL(XXUpp)
            robot.MoveL(XXDown)
            robot.MoveL(XXUpp)

robot.setPoseFrame(RobotRamme)
robot.MoveJ(home)
```

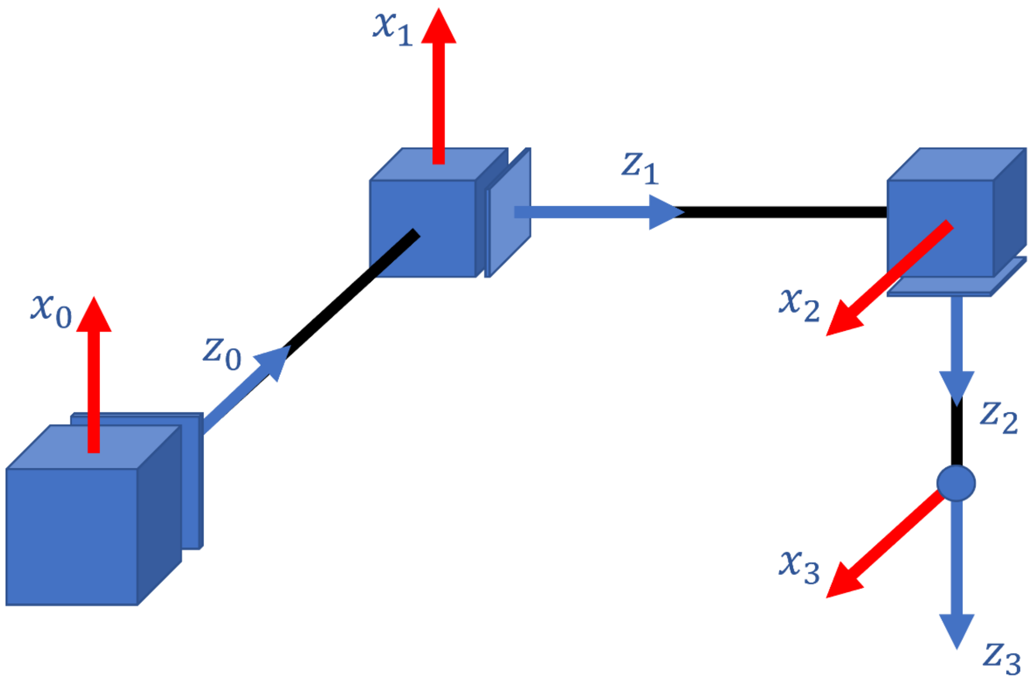
Oppgave 6

1. Roboten har 3 DOF.
2. Roboten består av 3 prismatiske ledd. En PPP robot.
3. Portalrobot.
4. Se bilder under:



Oppgave 7

1. Ny figur blir slik:



2. DH tabell er som følgende:

Ledd	d_i	a_i	α_i	θ_i
1	d_1^*	0	-90^o	0
2	d_2^*	0	-90^o	90^o
3	d_3^*	0	0	0

Oppgave 8

a. Kun 1.
b. Ingen flere enn 1. Roboten har kun 3 ledd hvor hvert ledd spenner i en av akseretningene til det kartesiske planet.
c. Med det nye leddet kan roboten innta høyst 3 konfigurasjoner på grunn av det siste roterende leddet som kan ha vinkel 0, 360 og -360.

Oppgave 9

For ledd1:

$$d_1 = z_c$$

For ledd2:

$$d_2 = y_c$$

For ledd3:

$$d_3 = -(x_c)$$

Oppgave 10

1. Vi har:

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad z_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad z_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

Med 3 prismatiske ledd blir vår Jacobian matrise:

$$J = \begin{bmatrix} z_0 & z_1 & z_2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

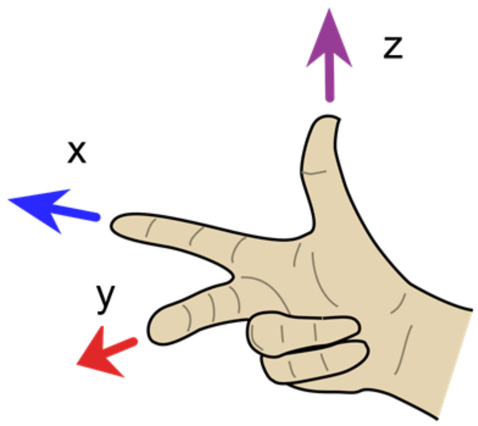
2. I teorien eksiterer det ikke noen singulariteter for denne roboten, men i praksis kan dette være tilknyttet begrensingene til robot leddene.

Appendiks

Her er hjelpestoff til eksamen listet opp

Appendiks 1

Høyrehåndsregelen er:



Positiv rotasjon av en akse er med klokken fra origo til enden av aksen.

Appendiks 2

Regler for klassisk DH-Konvensjon:

- a_i, K**oblings-Lengden (Link-Lenght)** for kobling i. Avstanden fra z_{i-1} til z_i målt langs x_i.
- α_i er **Koblings-Vridningen (Link-Twist)** for kobling i. Vinkelen mellom z_{i-1} til z_i, målt rundt x_i.
- d_i er **Koblings-Forskyvningen (Link-Offset)** for kobling i ←for prismatiske ledd. Avstanden mellom o_{i-1} til punktet der x_i aksen krysser z_{i-1}, målt langs z_{i-1}.
- θ_i er **Ledd-Vinkel (Joint-Angle)** for kolbing i. Er variabel for roterende ledd. Korteste vinkelen mellom x_{i-1} til x_i målt rundt z_{i-1}.

<u>Tilfelle1:</u>

z_i og z_{i-1} danner ikke samme plan. Det finnes bare en x_i, og det er den korteste veien mellom z_i og z_{i-1}. x_i må krysse og stå normalt på z_{i-1}.

<u>Tilfelle2:</u>

z_i og z_{i-1} er paralelle med hverandre. x_i og o_i kan bli dannet hvor som helst mellom z_i og z_{i-1}

<u>Tilfelle3:</u>

z_i og z_{i-1} krysser hverandre. x_i kan bli dannes på z_i, men må krysse og stå normalt på z_{i-1}.

Appendiks 3

Jacobian matrisen er definert som:

$$J = \left[\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Regler for Jacobian matrise for roboter i 3 dimensjoner.

Forholdet mellom en robots kartesiske fart med ledd-hastighet er:

$$\xi = J_n \dot{q}_n \leftrightarrow \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} J_{v_1} \dots J_{v_n} \\ J_{\omega_1} \dots J_{\omega_n} \end{bmatrix} \dot{q}_n$$

Den lineære hastigheten for hver kolonne av $J_v = \left[J_{v_1} \dots J_{v_n}\right]$, er definert som:

$$J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}), & for\ roterende \\ z_{i-1}, & for\ prismatiske \end{cases}$$

Den roterende hastigheten for hver kolonne av $J_\omega = \left[J_{\omega_1} \dots J_{\omega_n}\right]$ er definert som:

$$J_{\omega_i} = \begin{cases} z_{i-1}\ for\ roterende\ ledd \\ 0\ for\ prismatiske\ ledd \end{cases}$$

Kryss-produktet mellom 2 vektorer er definert som:

$$\vec{a} \times \vec{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

Determinanten til en 3x3 matrise:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$detA = a(ei - fh) - b(di - fg) + c(dh - eg)$$

Derivering av trigonometriske utrykk:

$$\sin(x)' = \cos(x)$$

$$\cos(x)' = -\sin(x)$$