Exercises – Forward kinematics

ELVE3610 Introduction to Robotics

Exercise 1: KINOVA Robot arm

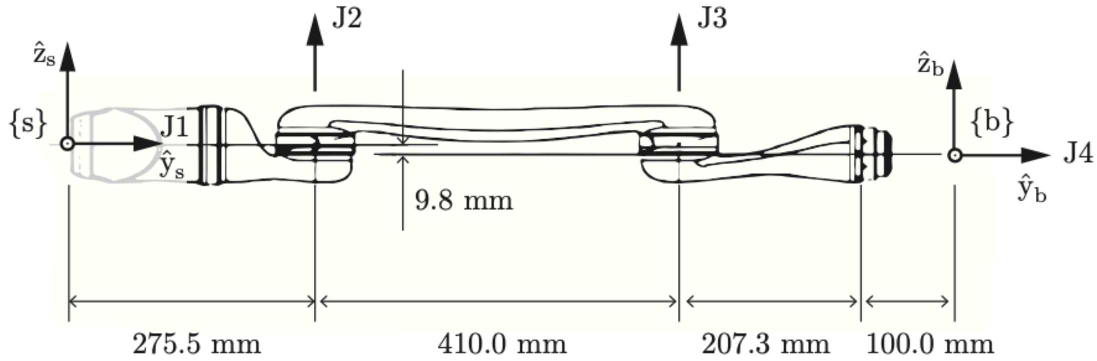


Figure 4.2: The KINOVA ultra lightweight 4-dof robot arm at its home configuration.

a) Determine the homogeneous transform ${}^{s}T_{b}$ in the home configuration. Result should be numeric (no symbols or math)
b) Determine Denavit Hartneberg parameters
c) Determine forward kinematics for generic joint angles $(\theta_1, \theta_2, \theta_3, \theta_4)$.
d) Implement forward kinematics as a python function that returns end effector pose

```
def forward_kinematics_kinova(q)

# input q = [q1,q2,q3,q4] is a numpy array of dimension 4

Return Tsb

# Returns homogeneous transformation matrix Tsb
```

e) Compute end effector pose for $(\theta_1, \ldots, \theta_4) = (\frac{\pi}{2}, 0, \frac{\pi}{2}, 0)$
f) Compute end effector pose for $(\theta_1, \ldots, \theta_4) = (\frac{\pi}{2}, \frac{3\pi}{4}, \frac{\pi}{4}, \pi)$

Exercise 2: Da Vinci surgical robot

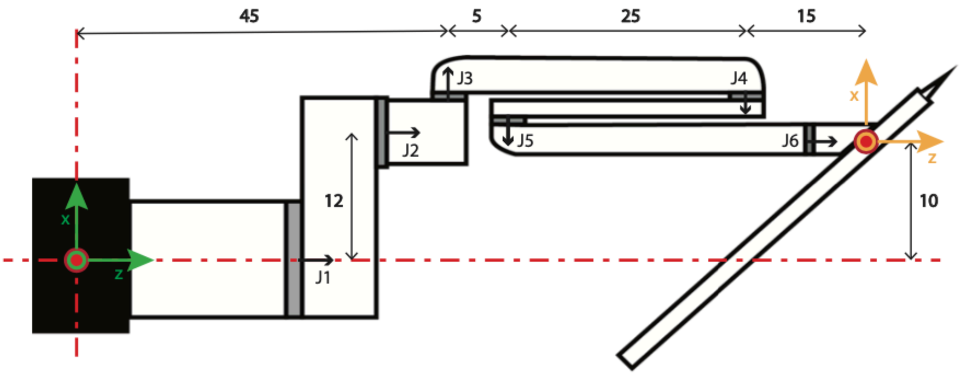


Figure 4.6: Top view of one da Vinci Xi surgical robot arm. Note that the grey regions represent R joints, green indicates the {s} frame, and yellow represents the end-effector frame {b} in this exercise. Dimensions are in cm.

a) Determine the homogeneous transform ${}^{s}\mathrm{T}_{\mathrm{b}}$ in the home configuration. Result should be numeric (no symbols or math)

b) Determine Denavit Hartneberg parameters

c) Determine forward kinematics for generic joint angles $(\theta_1, \ldots, \theta_6)$.

d) Implement forward kinematics as a python function that returns end effector pose

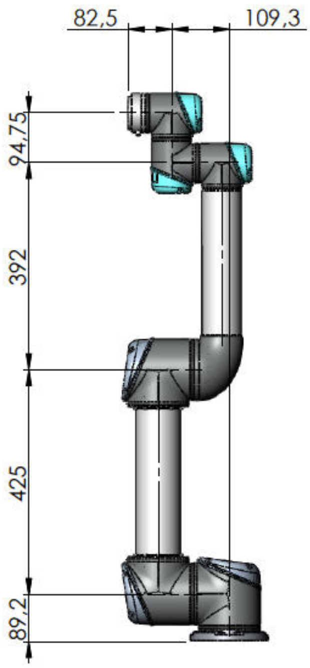
```
def forward_kinematics_davinci(q)
```

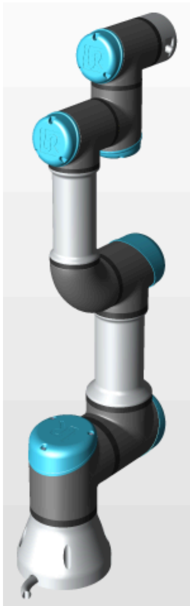
```
# input q = [q1,…,q6] is a numpy array of dimension 6

Return Tsb

# Returns homogeneous transformation matrix Tsb
```

e) Compute end effector pose for $(\theta_1, \ldots, \theta_6) = (\frac{\pi}{2}, 0, 0, \frac{\pi}{4}, \frac{\pi}{4}, 0)$

f) Compute end effector pose for $(\theta_1, \ldots, \theta_6) = (\frac{\pi}{2}, \frac{\pi}{5}, \frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{3\pi}{4})$

Exercise 3: Universal robots UR3





a) Determine link referenece frames based on Denavit Hartneberg convention
b) Determine the homogeneous transform $^{s}T_b$ in the home configuration. Result should be numeric (no symbols or math)
c) Determine Denavit Hartneberg parameters
d) Determine forward kinematics for generic joint angles $(\theta_1, \ldots, \theta_6)$.
e) Implement forward kinematics as a python function that returns end effector pose

```
def forward_kinematics_UR3(q)

# input q = [q1,…,q6] is a numpy array of dimension 6

Return Tsb

# Returns homogeneous transformation matrix Tsb
```

f) Compute end effector pose for $(\theta_1, \dots, \theta_6) = (\frac{\pi}{2}, 0{,}0, \frac{\pi}{4}, \frac{\pi}{4}, 0\ )$

g) Compute end effector pose for $(\theta_1, \dots, \theta_6) = (\frac{\pi}{2}, \frac{\pi}{5}, \frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{3\pi}{4})$