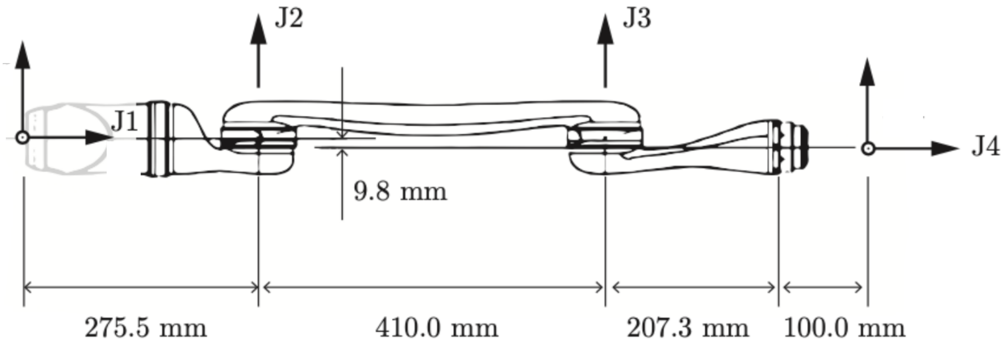
Exercises – Forward kinematics

ELVE3610 Introduction to Robotics

Exercise 1: KINOVA Robot arm



a) Assign reference frames to each link according to Denavit-Hartneberg convention
b) Determine the homogeneous transform ${}^{0}T_4$ 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 $(\theta_1, \theta_2, \theta_3, \theta_4)$.
e) 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 T04
# Returns 4x4 homogeneous transformation matrix Tsb
```

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

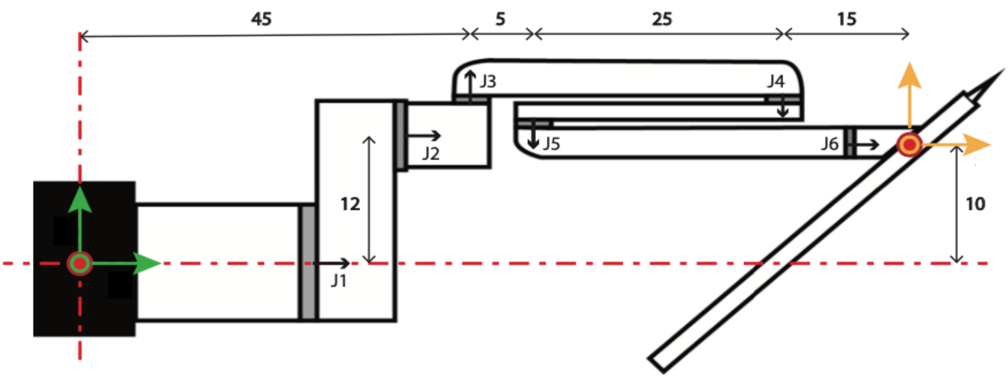


Figure 4.6: Top view of one da Vinci Xi surgical robot arm. Note that the grey regions represent R joints, green indicates the {s} frame, and yellow represents the end-effector frame {b} in this exercise. Dimensions are in cm.

a) Assign reference frames to each link according to Denavit-Hartneberg convention
b) Determine the homogeneous transform ${}^{0}T_{6}$ 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

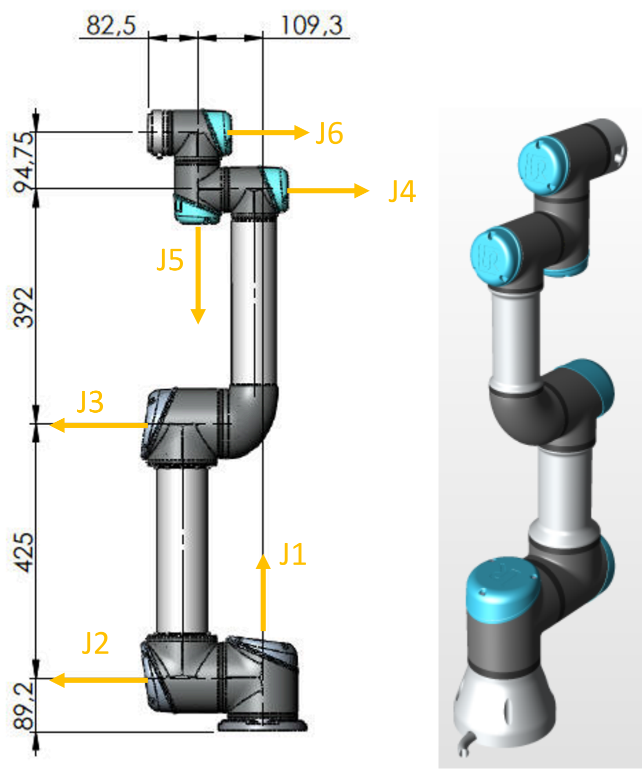
```
def forward_kinematics_davinci(q):
```

```
# input q = [q1,…,q6] is a numpy array of dimension 6
Return Tsb
# Returns 4x4 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 $(\theta_1, \dots, \theta_6) = (\frac{\pi}{2}, \frac{\pi}{5}, \frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{3\pi}{4})$

Exercise 3: Universal robots UR3



a) Determine link referenece frames of each link based on Denavit Hartneberg convention

b) Determine the homogeneous transform $^0T_6$ 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 $(\theta_1, \dots, \theta_6)$.

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 Toe
```

```
# Returns 4x4 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 $(\theta_1, \dots, \theta_6) = (\frac{\pi}{2}, \frac{\pi}{5}, \frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{3\pi}{4})$