

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH
TEHNOLOGIJA OSIJEK

Diplomski studij – Robotika i umjetna inteligencija

Osnove Robotike

Laboratorijska vježba 1

Direktna kinematika robotskog manipulatora

Ivan Gudelj, DRB

Osijek, 2022.

I. Cilj vježbe

Upoznati se s metodologijom opisa prostornih odnosa između trodimenzionalnih objekata primjenom matrica homogenih transformacija te steći vještinu u izgradnji kinematičkog modela robotskog manipulatora primjenom Denavit-Hartenbergove metode.

II. Opis vježbe

Problem direktne kinematike koji se razmatra u ovoj vježbi je odrediti položaj alata, odnosno koordinatnog sustava L6, u odnosu na bazni koordinatni sustav L0, koji miruje u odnosu na okolinu robota, za zadane vrijednosti varijabli zglobova. Položaj nekog koordinatnog sustava u odnosu na neki drugi koordinatni sustav može se opisati matricom homogene transformacije. Matrica T_6^0 koja opisuje položaj koordinatnog sustava L6 u odnosu na L0 može se odrediti na temelju kinematičkog modela manipulatora. Zadani robotski manipulator sa 6 rotacijskih zglobova može se modelirati kao kinematički lanac članaka povezanih zglobovima. Primjenom Denavit-Hartenbergove metode potrebno je svakom itom članku pridružiti koordinatni sustav L_i koji miruje u odnosu na taj članak te, uz postavljanje koordinatnog sustava L0 koji miruje u odnosu na taj članak, odrediti kinematičke parametre koji opisuju međusobni položaj koordinatnih sustava L_i i L_{i-1} . Pomoću tih parametara mogu se odrediti sljedeće matrice homogene transformacije.

$$\mathbf{T}_{i-1}^i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad i = 1, 2, 3, 4, 5, 6,$$

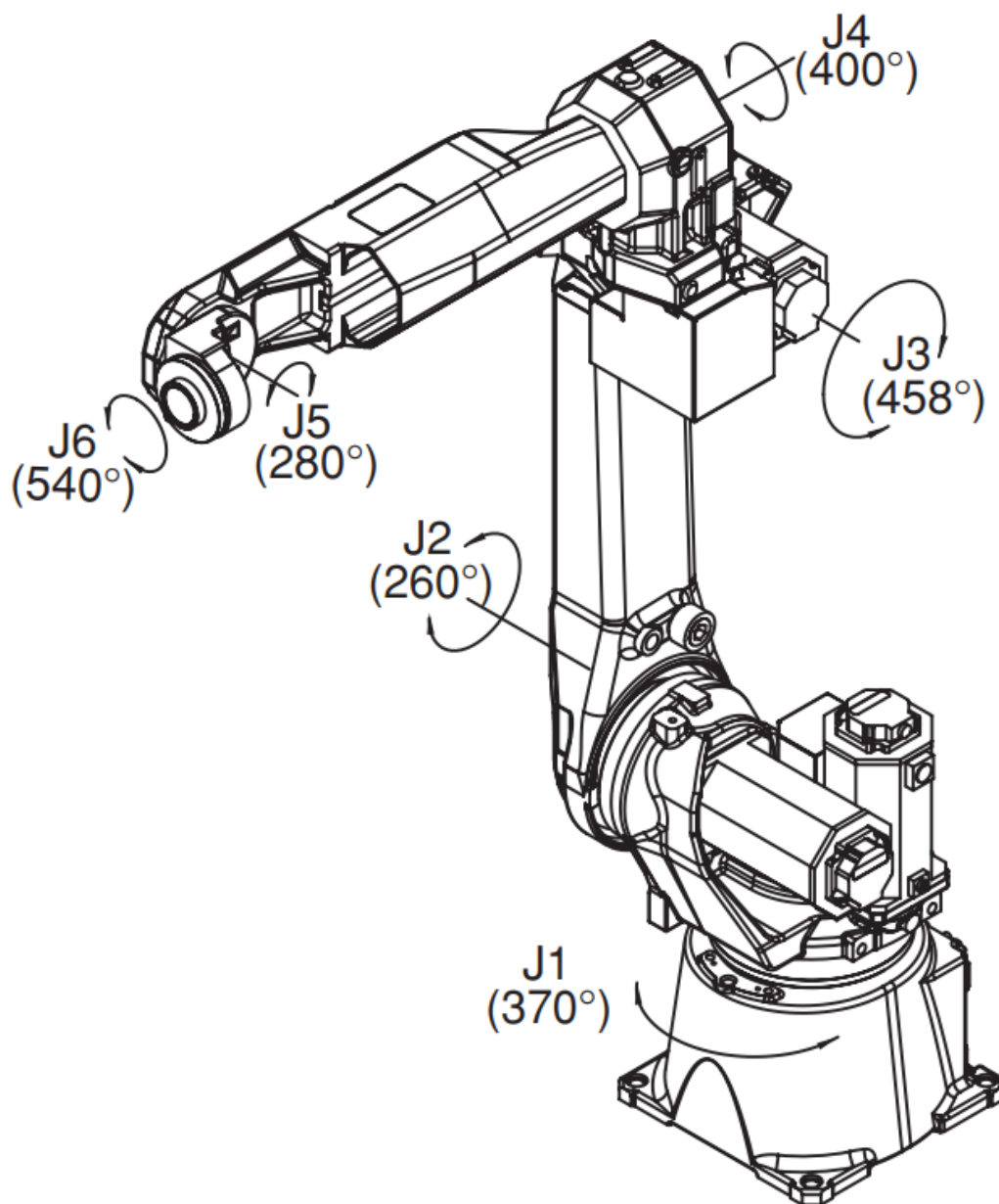
$$c\theta_i = \cos\theta_i, \quad s\theta_i = \sin\theta_i, \quad c\alpha_i = \cos\alpha_i, \quad s\alpha_i = \sin\alpha_i.$$

Pomoću tih matrica može se izračunati matrica T_0^6 na sljedeći način:

$$\mathbf{T}_0^6 = \mathbf{T}_0^1 \cdot \mathbf{T}_1^2 \cdot \mathbf{T}_2^3 \cdot \mathbf{T}_3^4 \cdot \mathbf{T}_4^5 \cdot \mathbf{T}_5^6.$$

III. Rješenje

Slika prikazuje zglobove:



Parametre Θ_i , d_i , a_i , α_i smo odredili na sljedeći način:

$\Theta_i \rightarrow \text{rot}(z_{i-1}); x_{i-1} \rightarrow x_i$

$d_i \rightarrow |L_{i-1} \rightarrow L_i|$ po z_{i-1}

$a_i \rightarrow |L_{i-1} \rightarrow L_i|$ po x_i

$\alpha_i \rightarrow \text{rot}(x_i); z_{i-1} \rightarrow z_i$

Nakon određivanja parametara smo implementirali zadani model u python skriptu čiji se kod nalazi unutar .zip datoteke.

Model članaka je implementiran koristeći se vtk bibliotekom i njenim metodama kao što su *cube* i *cylinder* koje kreiraju kvadar odnosno valjak zadanih dimenzija.

```
# Link 4.
self.link4 = vis.cube(0.15, 0.790, 0.15)
s.add_actor(self.link4)

# Link 5.
self.link5 = vis.cylinder(0.075, 0.150)
s.add_actor(self.link5)
```

Primjer jednog članka u obliku kvadra i jednog u obliku valjka

Nakon stvaranja svih dijelova robota bilo je potrebno definirati matrice koje nam govore odnose između pojedinih djelova kao i njihov odnos sa scenom.

```
# Link 1.
T10 = dh(q[0], d[0], a[0], al[0])
T1S = T0S @ T10
#Rotacijska 1
TL11 = np.identity(4)
TL11[1,3] = 0.08125
TL1S = T1S @ TL11
vis.set_pose(self.link1, TL1S)
```

Primjer implementacije jednog članka (Link 1)

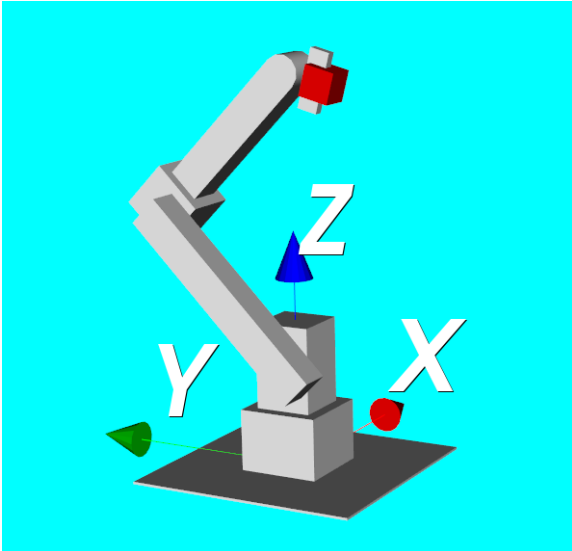
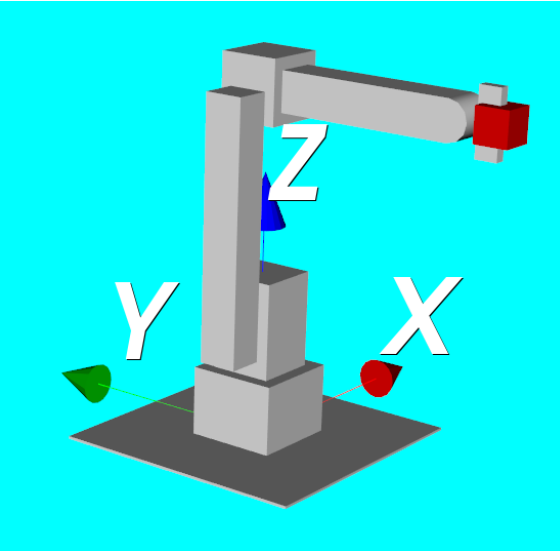
Dakle matrica T10 se (pomoću predefinirane funkcije *dh*) isčitava iz naše prethodno definirane tablice Denavit -Hartenbergove metode.

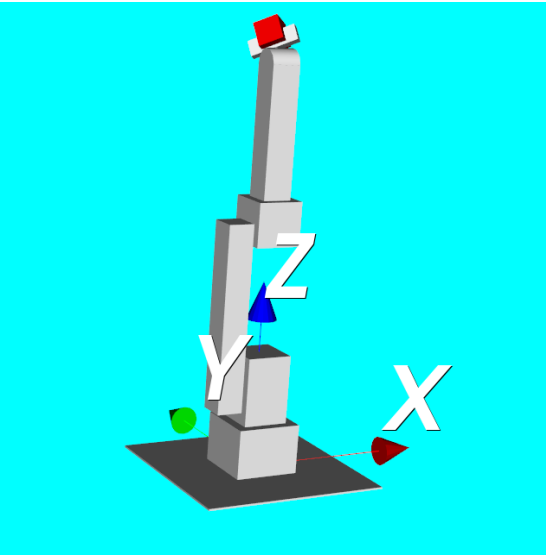
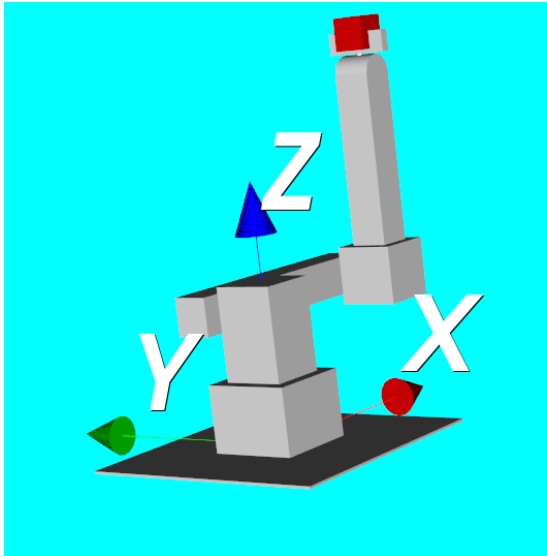
Matrica T1S predstavlja odnos između i-tog članka i scene dok matrica TL11 predstavlja Matricu homogene transformacije tog članka (rotacija i translacija) u odnosu na scenu (TL1S).

Nakon kreiranja 5 članaka bilo je potrebno dodati i alat (ručicu) koja zapravo robotu daje smisao. Same dimenzije alata su na slici ispod.

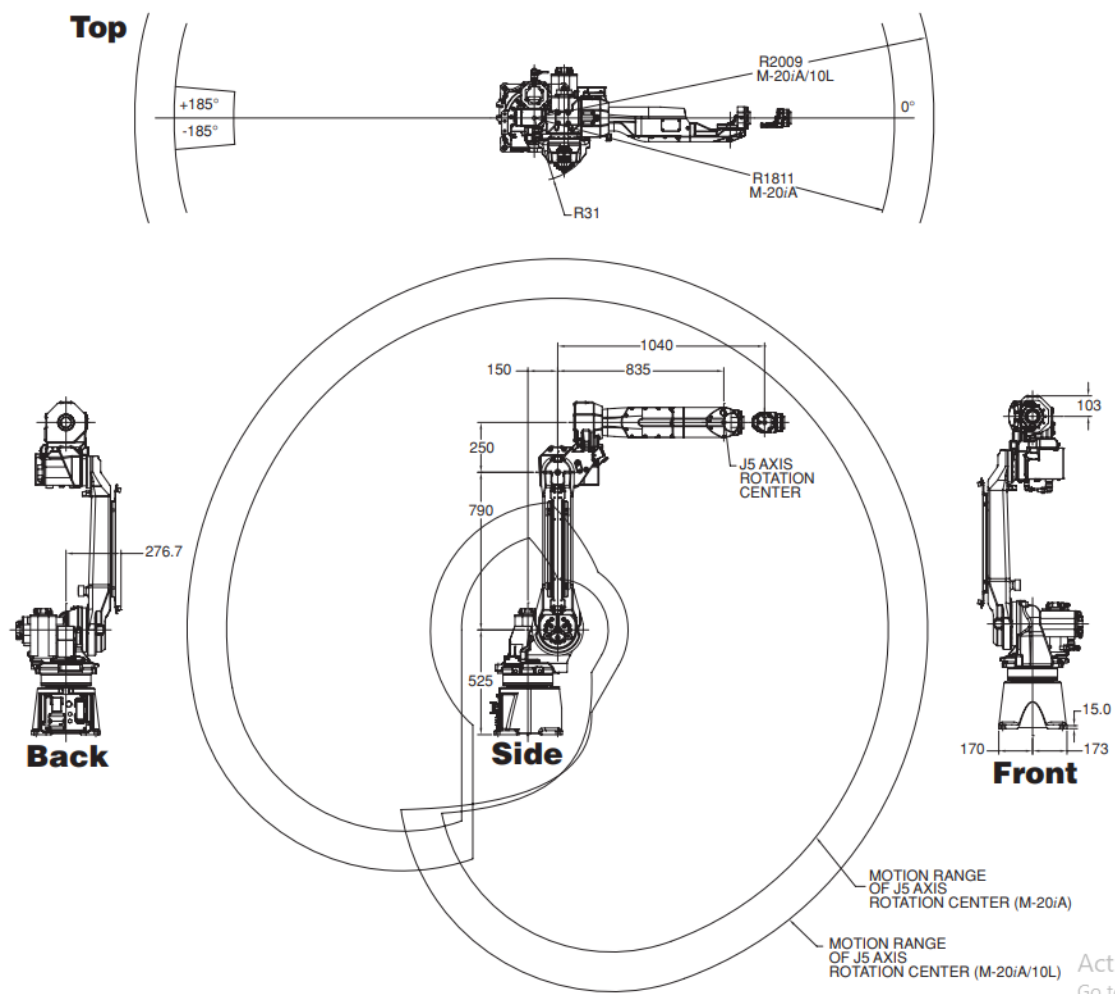
```
s = scene
self.finger1 = vis.cube(0.04, 0.06, 0.08)
s.add_actor(self.finger1)
self.finger2 = vis.cube(0.04, 0.06, 0.08)
s.add_actor(self.finger2)
self.palm = vis.cube(0.04, 0.25, 0.02)
s.add_actor(self.palm)
self.wrist = vis.cylinder(0.015, 0.04)
s.add_actor(self.wrist)
```

Nakon *slaganja* robota, bilo je potrebno smjestiti kvadrat u ručicu (alat) robota te sve to prikazati modelom. U nastavku slijede primjeri robota s različitim kutevima pojedinih članaka odnosno s različitim položajima. (kut 1. Članka, kut 2. Članka ...)

	
$\text{np.pi}/2, -\text{np.pi}/4, 0, 0, -\text{np.pi}/3, -\text{np.pi}/2$	$\text{np.pi}/2, -\text{np.pi}/2, 0, 0, 0, -\text{np.pi}/2$

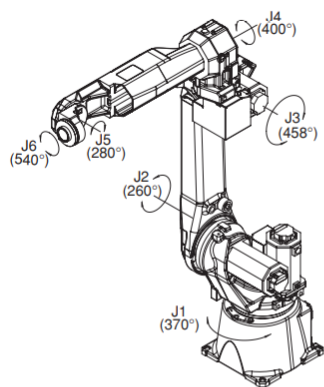
	
$\text{np.pi}/2, -\text{np.pi}/2, \text{np.pi}/2, 0, \text{np.pi}/5, -\text{np.pi}/4$	$0, 0, 0, 0, 0, 0$

Detalji o robotu:

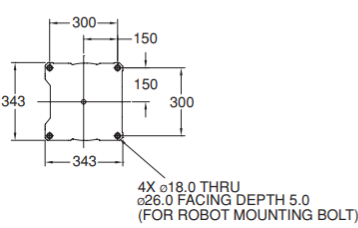


Activate Wind
Go to Settings to a

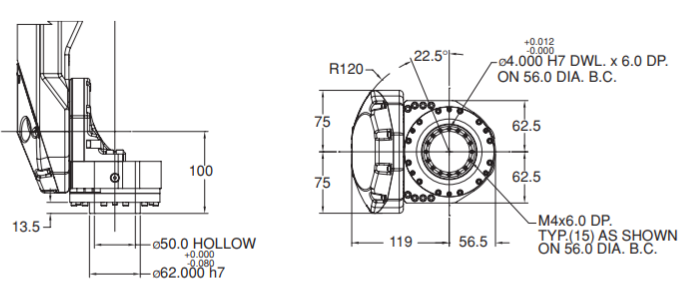
Isometric



Footprint



Wrist M-20iA



Wrist M-20iA/10L

