

# Koncept povezivanja virtualnog modela robota s robotom u sklopu stvarne okoline

---

**Kukurin, Luka**

**Master's thesis / Diplomski rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:496462>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-14**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Luka Kukurin**

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

doc. dr. sc. Tomislav Stipančić, dipl. ing.

Student:

Luka Kukurin

Zagreb, 2022.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem mentoru doc. dr. sc. Tomislavu Stipančiću, dipl. ing., na prenesenom znanju, pomoći i susretljivosti tijekom cijelog studija te asistentu Leonu Korenu, mag. ing. mech. na pomoći i razumijevanju tijekom izrade diplomskog rada.

Luka Kukurin



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602-04/22-6/1	
Ur. broj: 15-1703-22-	

## DIPLOMSKI ZADATAK

Student: **LUKA KUKURIN** Mat. br.: 0035204686

Naslov rada na hrvatskom jeziku: **Koncept povezivanja virtualnog modela robota s robotom u sklopu stvarne okoline**

Naslov rada na engleskom jeziku: **The concept of connecting a virtual model of the robot with a robot within a real environment**

Opis zadatka:

Model robota ili manipulatora kao reprezentaciju koja je realizirana u sklopu virtualne okoline moguće je povezati s fizičkim robotom u sklopu stvarne okoline. Na taj je način moguće ostvariti sinkrono (engl. online) ili asinkrono (engl. offline) programiranje iz virtualne okoline koje se ostvaruje u realnom svijetu. Navedena rješenja mogu ići do razine digitalnog blizanca (engl. digital twin) koji predstavlja cjelovit virtualni model koji je kreiran da precizno prostorno i vremenski predstavlja objekt iz fizičke okoline. U tome slučaju promjene koje se događaju u sklopu virtualne okoline mogu se vjerno odražavati u sklopu fizičke okoline.

U radu je potrebno:

- načiniti 3D-model robota kartezijske strukture
- prenijeti ga u RoboDK virtualnu okolinu
- definirati pripadajuću kinematiku za svaku od osi robota
- definirati odgovarajući upravljački program koji omogućuje komunikaciju modela robota iz virtualne okoline s robotom u sklopu stvarne okoline te tako omogućiti sinkrono (engl. online) programiranje.

Razvijeno rješenje potrebno je eksperimentalno verificirati na opremi dostupnoj u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava. U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:  
20. siječnja 2022.

Zadatak zadao:  
doc. dr. sc. Tomislav Stipančić

Rok predaje rada:  
24. ožujka 2022.

Predvideni datum obrane:  
28. ožujka do 1. travnja 2022.

Predsjednica Povjerenstva:  
prof. dr. sc. Biserka Runje

**SADRŽAJ**

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA.....	V
POPIS KRATICA .....	VI
POPIS OZNAKA .....	VII
SAŽETAK.....	VIII
SUMMARY .....	IX
1. UVOD.....	1
2. POVIJEST ROBOTA.....	2
3. KOORDINATNI SUSTAVI .....	6
3.1. Kartezijev koordinatni sustav.....	6
3.2. Cilindrični koordinatni sustav .....	6
3.3. Sferni koordinatni sustav .....	7
3.4. Koordinatni sustavi robota .....	8
3.4.1. Koordinatni sustav svijeta.....	8
3.4.2. Bazni koordinatni sustav .....	8
3.4.3. Koordinatni sustav prirubnice.....	9
3.4.4. Alatni koordinatni sustav .....	9
4. VRSTE INDUSTRIJSKIH ROBOTA.....	10
4.1. Kartezijevi roboti .....	10
4.2. SCARA roboti.....	10
4.3. DELTA roboti .....	11
4.4. Polarni roboti.....	12
4.5. Zglobni roboti .....	12
4.6. Cilindrični roboti.....	13
5. POGON ROBOTSКИH ČLANAKA .....	14
6. IZVRŠNI ELEMENTI ROBOTA .....	15
6.1. Vakuumske hvataljke .....	15

6.2. Pneumatske hvataljke.....	16
6.3. Električne hvataljke.....	16
6.4. Hidrauličke hvataljke .....	17
7. SENZORI U ROBOTICI.....	18
7.1. Svjetlosni senzori .....	18
7.2. Senzori zvuka .....	19
7.3. Senzori blizine.....	19
7.4. Taktilni senzori .....	19
8. ROBOT IZ ZADATKA.....	20
9. MODELIRANJE ROBOTA.....	22
10. RoboDK.....	25
11. IMPLEMENTACIJA ROBOTA U RoboDK.....	27
11.1. Vrste putanja gibanja .....	30
12. UPRAVLJAČKI PROGRAM .....	31
13. SIMULACIJA .....	33
14. ZAKLJUČAK.....	36
LITERATURA.....	37
PRILOZI.....	39

**POPIS SLIKA**

Slika 1.	Mehanička ptica [1].....	2
Slika 2.	Patka koja probavlja [2] .....	3
Slika 3.	Reklamni plakat predstave RUR [3] .....	3
Slika 4.	Robot Unimate i njegovi izumitelji [4] .....	4
Slika 5.	Broj aktivnih industrijskih robota po godinama (u tisućama) [5] .....	5
Slika 6.	Raspodjela broja aktivnih robota prema industrijama [6] .....	5
Slika 7.	Kartezijev koordinatni sustav .....	6
Slika 8.	Cilindrični koordinatni sustav .....	7
Slika 9.	Sferni koordinatni sustav .....	8
Slika 10.	Bazni koordinatni sustav robota [7] .....	8
Slika 11.	Koordinatni sustav prirubnice [8].....	9
Slika 12.	Alatni koordinatni sustav [9].....	9
Slika 13.	Kartezijev robot [10] .....	10
Slika 14.	SCARA robot [11].....	11
Slika 15.	DELTA robot [12].....	11
Slika 16.	Polarni robot [13] .....	12
Slika 17.	Zglobni robot sa 6 stupnjeva slobode gibanja [14] .....	13
Slika 18.	Cilindrični robot [15].....	13
Slika 19.	Vakuumska hvataljka [16].....	15
Slika 20.	Način rada pneumatske hvataljke [17] .....	16
Slika 21.	Električna hvataljka [18] .....	16
Slika 22.	Hidraulička hvataljka [19].....	17
Slika 23.	Kartezijev robot u Laboratoriju za projektiranje izradbenih i montažnih sustava	21
Slika 24.	Aluminijski profil 90x90 (lijevo) i aluminijski profil 90x45 (desno) .....	22
Slika 25.	Postolje robota s radnim stolom .....	22
Slika 26.	Os robota .....	23
Slika 27.	Nosač y osi (lijevo) i nosač klizača z osi (desno).....	23



Slika 28.	Nosač servomotora (lijevo) i pločica (desno).....	24
Slika 29.	3D sklop robota .....	24
Slika 30.	Isječak knjižnice robota u softveru RoboDK .....	25
Slika 31.	Model robota u programu RoboDK nakon naredbe <i>split</i> .....	27
Slika 32.	Koordinatni sustav prihvatnice.....	28
Slika 33.	Prozor za stvaranje mehanizma .....	29
Slika 34.	Robot s kreiranim mehanizmom .....	30
Slika 35.	Dio koda upravljačkog programa za stvaranje rječnika s koordinatama.....	31
Slika 36.	Dio koda upravljačkog programa za pozivanje funkcija pomaka na strani robota	32
Slika 37.	Izbornik za povezivanje s robotom .....	33
Slika 38.	Stanje robota u virtualnoj okolini na početku simulacije (lijevo) i na kraju simulacije (desno).....	34
Slika 39.	Stanje robota u stvarnoj okolini na početku simulacije (lijevo) i na kraju simulacije (desno).....	34

## **POPIS TABLICA**

Tablica 1. Dimenzije postolja robota i pomaci osi .....	20
Tablica 2. Snage servomotora za svaku os .....	20

## POPIS KRATICA

CAD	<i>Computer Aided Design</i> – računalom potpomognuto oblikovanje
TCP	<i>Tool center point</i> – Središnja točka alata
SCARA	<i>Selective Compliance Assembly Robot Arm</i>
PLC	<i>Programmable Logic Controller</i> – programirljivi logički kontroler
IP	<i>Internet protocol</i> – Internet protokol

**POPIS OZNAKA**

<b>Oznaka</b>	<b>Jedinica</b>	<b>Opis</b>
$x$	m	Udaljenost točke P od ishodišta O u smjeru x osi
$\rho$	m	Euklidska udaljenost između točke P i osi z
$\varphi$	Rad	Kut između osi x i vektora usmjerenog od ishodišta prema projekciji točke P na referentnu ravninu
$y$	m	Udaljenost točke P od ishodišta O u smjeru y osi
$z$	m	Udaljenost točke P od ishodišta O u smjeru z osi
$\theta$	Rad	Kut između osi z i vektora usmjerenog od ishodišta prema točki P

## **SAŽETAK**

U radu je ostvarena realizacija robota u sklopu virtualne okoline koja je povezana s fizičkim robotom u sklopu stvarne okoline, te je ostvarena mogućnost online programiranja robota. To znači da se promjene u virtualnoj okolini vjerno odražavaju u stvarnoj okolini. Uvodni dio rada opisuje povijest robota te navodi značajne trenutke u razvoju robotike. Opisani su najčešći tipovi robota, vrste aktuatora i često korišteni senzori. Robot je modeliran u virtualnoj okolini i implementiran u softver RoboDK. Izrađen je upravljački program koji omogućuje komunikaciju i upravljanje stvarnim robotom.

Ključne riječi: Kartezijev robot, 3D modeliranje, RoboDK, upravljački program, robotika

## **SUMMARY**

The paper examines the realization of a robot within a virtual environment that is connected to a physical robot within a real environment, and the possibility of online robot programming is accomplished. This means that changes in the virtual environment are faithfully reflected in the real environment. The introductory part of the paper describes the history of robotics and lists significant moments in the development of robotics. The most common types of robots, types of actuators and frequently used sensors are described. The robot is modeled in a virtual environment and implemented in RoboDK software. A driver that enables communication and control of the real robot has been created.

Key words: Cartesian robot, 3D modeling, RoboDK, driver, robotics

## 1. UVOD

Robotika je interdisciplinarna znanost koja se bavi konceptom, konstrukcijom, upravljanjem i proizvodnjom robota. U njoj se u najvećoj mjeri susreću znanosti elektrotehnike, strojarstva i informatike. Pisac znanstvene fantastike Issac Asimov je 1942. u kratkoj priči predložio tri zakona robotike:

1. Robot ne smije naškoditi čovjeku ili svojom pasivnošću dopustiti da se čovjeku naškodi.
2. Robot mora slušati ljudske naredbe, osim kad su one u suprotnosti s prvim zakonom.
3. Robot treba štiti svoj integritet, osim kad je to u suprotnosti s prvim ili drugim zakonom.

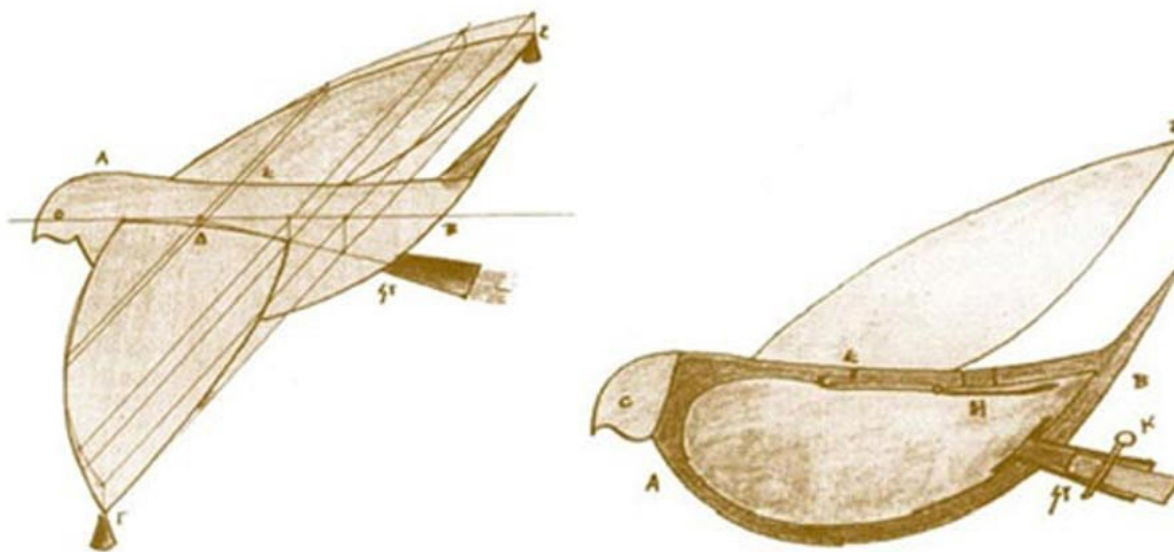
Kasnije je nadodan nulti zakon koji ima prednost nad ostalima.

0. Robot ne smije naškoditi čovječanstvu, ili svojom pasivnošću dozvoliti da se čovječanstvu naškodi.

Iako nije teško smisliti situacije u kojima se jednoznačnost ovih zakona dovodi u pitanje, oni još uvijek nose svoju težinu. Postoji nešto utješno u spoznaji da su roboti programirani tako da neće nauditi ljudima niti čovječanstvu, jer je strah od robota kroz prizmu straha od nepoznatog duboko ukorijenjen u ljude. Međutim, cilj robota je da pomognu ljudima. To je posebno izraženo kod industrijskih robota. Iako će kritičari reći da roboti oduzimaju poslove ljudima, zdraviji pogled na situaciju je da industrijski roboti oslobađaju ljude zamornih, repetitivnih i fizički teških poslova te im omogućuju da se bave nečim kreativnim. Jer to je ono u čemu ljudi, barem za sada, imaju prednost.

## 2. POVIJEST ROBOTA

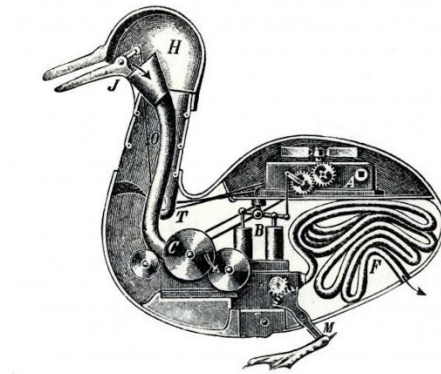
Koliko nam je danas poznato, prve ideje o stvaranju nečeg nalik današnjem poimanju riječi robot sežu još u četvrto stoljeće prije Krista, kada je Grčki matematičar Archytas dobio ideju za izradu mehaničke ptice [Slika 1]. Ovo se dostignuće često spominje kao prvi robot u povijesti čovječanstva, a možemo slobodno reći da se radi i o prvom dronu. Archytasova ptica je koristila mehanizam mahanja krilima kojeg je pogonila para.



Slika 1. Mehanička ptica [1]

Postoje zapisi o mnogim drugim pokušajima i idejama koje su često bile inspirirane prirodom i težile k njenoj imitaciji, pa možemo nastaviti s primjerom patke koja probavlja [Slika 2]. Ta se kreacija Jacquesa de Vaucansona ponašala kao prava patka – kvakala, gegala se, lamatala krilima i čak probavljala hranu. Iako se ispostavilo da je ovo posljednje zapravo prevara i da patka ima poseban spremnik s probavljenom hranom, to ne umanjuje značaj ove mehaničke životinje koja je na svakom krilu imala preko 400 dijelova i za to vrijeme bila mehanički vrlo napredna kreacija.





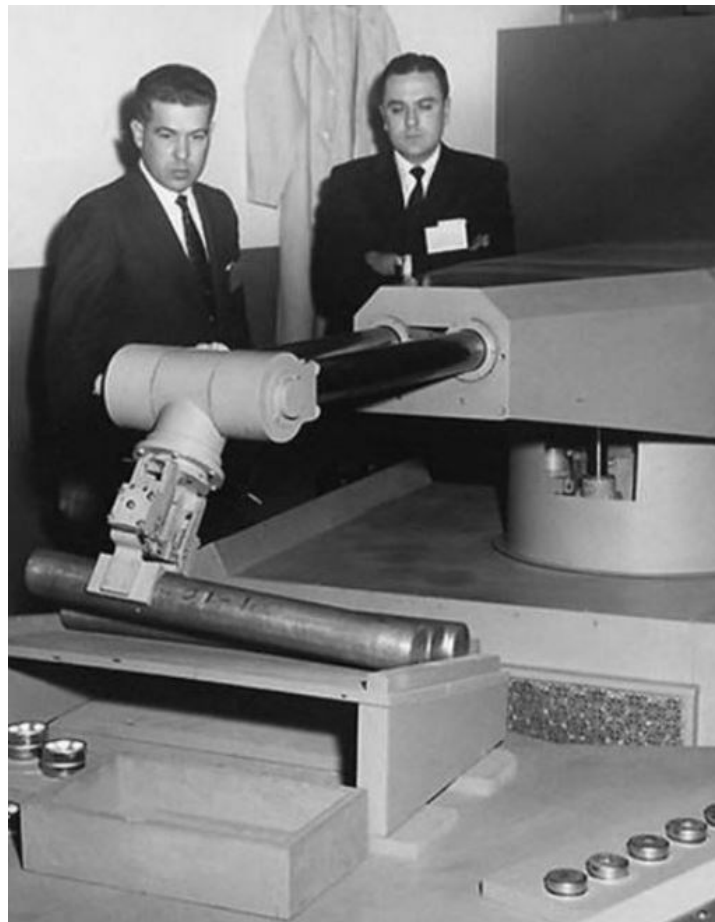
Slika 2. Patka koja probavlja [2]

Ako se fokusiramo na bližu povijest neizbježno je spomenuti jednu predstavu. Radi se o predstavi RUR [Slika 3] Češkog pisca Karel Čapeka napisanoj 1920. godine, gdje se roboti koriste poput robova, kao jeftina radna snaga bez radničkih prava. I sama riječ robot potječe iz češkog *robota*, odnosno prisilan rad. Roboti se ipak na kraju pobune protiv svojih izrabljivača te ih unište zajedno s cjelokupnim stanovništvom. Koliko je ova drama proročanska vjerojatno nećemo saznati u skorije vrijeme, s obzirom na to da se civilizacijski još uvijek nalazimo u prvom činu spomenute drame.



Slika 3. Reklamni plakat predstave RUR [3]

Van scenskih dasaka značajan trenutak u razvoju (industrijske) robotike je robot zvan Unimate [Slika 4], koji se može nazvati prvim industrijskim robotom. Razvijen od strane Amerikanaca Devola i Engelberga, ova robotska ruka služila je prijenosu tlačno lijevanih komada s pokretne trake i zavarivanju istih na automobile. Posao koji je opasan po čovjeka (zbog otrovnih dimova koji se ispuštaju u procesu) sada je pao na robota, što na neobičan način zadovoljava prvi Asimov zakon koji kaže “Robot ne smije ozlijediti ljudsko biće ili, nečinjenjem, dopustiti ljudskom biću da se ozlijedi.”.



**Slika 4. Robot Unimate i njegovi izumitelji [4]**

Unimateovo prvo zaposlenje dogodilo se 1959. godine na traci General Motorsa u New Jerseyu. Do 1961. proizvedeno je oko 450 primjeraka, a taj je uspjeh potaknuo izumitelje da pokrenu prvu kompaniju u grani robotike.

Do sljedećeg značajnog perioda u razvoju industrijske robotike došlo je u osamdesetim godinama prošlog stoljeća zahvaljujući automobilskoj industriji koja je prepoznala važnost i mogućnosti industrijskih robota te investirala velike svote novaca u njihov razvoj. U robote se

integriraju razni senzori i jednostavni vizijski sustavi, pa možemo govoriti o začecima inteligencije u robota.

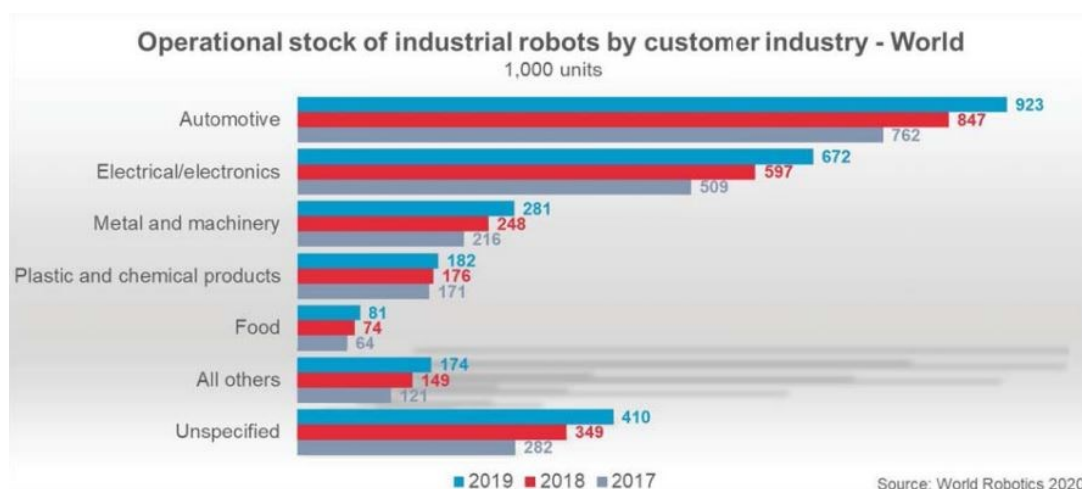
U dvadeset prvom stoljeću nositelji razvoja robotike su inovacije u svijetu strojnog učenja i umjetne inteligencije. Roboti su sve samostaljniji u donošenju odluka, a svi se mjerljivi podatci automatski spremaju u oblak te se mogu koristiti za razne spoznaje o efikasnosti robota.

2019. godine je u svijetu postojalo oko 2.7 milijuna industrijskih robota u tvornicama diljem svijeta, što je gotovo 3 puta više u odnosu na 2009. godinu.



Slika 5. Broj aktivnih industrijskih robota po godinama (u tisućama) [5]

Kao što je već napomenuto, značajan interes i investicije u razvoj robotike pokazala je autoindustrija, u kojoj se nalazi svaki treći industrijski robot na svijetu. Slijede ju industrija elektronike sa 672 000 robota i metaloprerađivačka industrija s 281 000 robota.



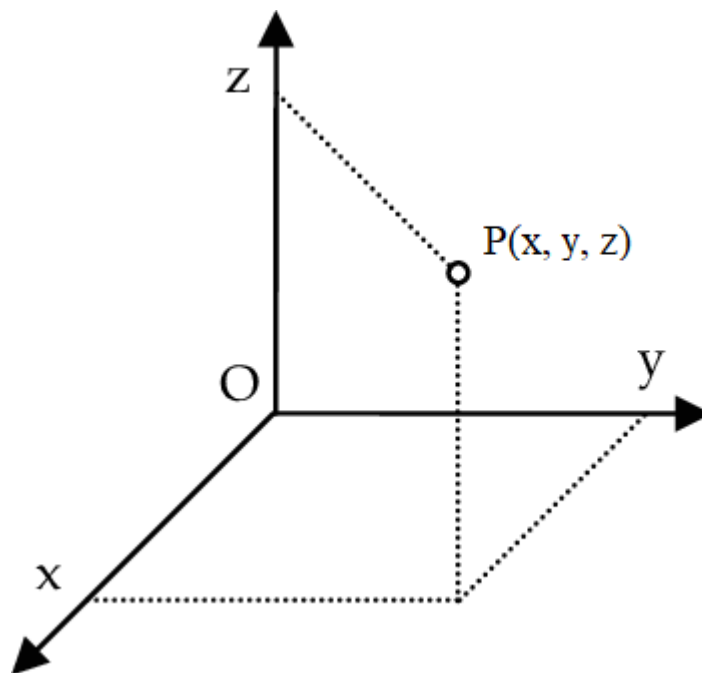
Slika 6. Raspodjela broja aktivnih robota prema industrijama [6]

### 3. KOORDINATNI SUSTAVI

Koordinatni sustavi su osnova poimanja 3D prostora i neophodni su za pravilno i precizno funkcioniranje robota. U trodimenzionalnom prostoru razlikujemo kartezijev, cilindrični i sferni koordinatni sustav. S obzirom na to da svi sustavi daju istu informaciju ali s različitim prikazom, postoje jednačbe za transformacije iz jednog koordinatnog sustava u drugi.

#### 3.1. Kartezijev koordinatni sustav

Kartezijev koordinatni sustav [Slika 7] se sastoji od ishodišta u kojem se sijeku koordinatne osi, te od 3 međusobno okomite koordinatne osi. U kartezijevom sustavu se neka točka u prostoru može opisati pomoću 3 numeričke koordinate koje predstavljaju udaljenost točke od ishodišta u smjeru određene koordinate osi. Ovaj se koordinatni sustav najčešće primjenjuje u robotici.



Slika 7. Kartezijev koordinatni sustav

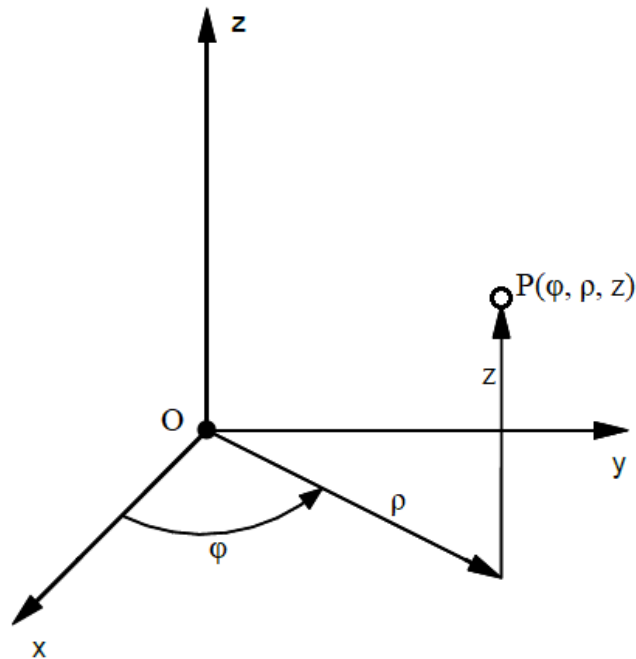
#### 3.2. Cilindrični koordinatni sustav

Cilindrični koordinatni sustav [Slika 8] se sastoji od ishodišta i tri koordinatne osi, a poziciju točke u prostoru opisuje pomoću 3 vrijednosti. Dvije vrijednosti su udaljenosti točke od ishodišta duž dviju koordinatnih osi (analogno kartezijevom koordinatnom sustavu u dvije dimenzije), dok treća vrijednost predstavlja kut zakreta oko vertikalne koordinatne osi. Jednačbe za pretvorbu koordinata u kartezijev koordinatni sustav su:

$$x = \rho \cdot \cos \varphi \quad (1)$$

$$y = \rho \cdot \sin \varphi \quad (2)$$

$$z = z \quad (3)$$



Slika 8. Cilindrični koordinatni sustav

### 3.3. Sferni koordinatni sustav

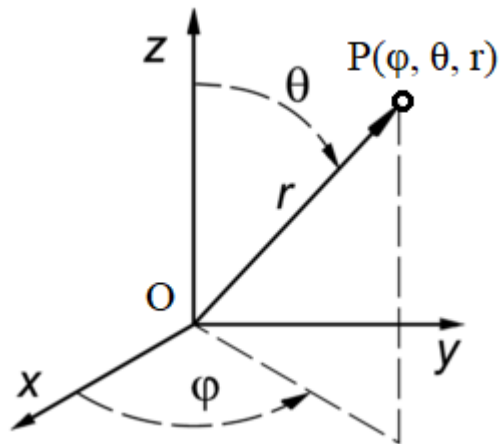
Točke u sfernom koordinatnom sustavu [Slika 9] su određene pomoću udaljenosti točke od ishodišta, te kutevima zakreta vektora usmjerenog iz ishodišta prema datoj točki oko dviju koordinatnih osi.

Prigodan je za korištenje u situacijama gdje je potrebno opisivati sferne oblike, pa se često koristi u geodeziji i kartografiji. Jednadžbe za pretvorbu sfernih koordinata u kartezijeve koordinate glase:

$$x = r \cdot \cos \varphi \cdot \sin \theta \quad (4)$$

$$y = r \cdot \sin \varphi \cdot \sin \theta \quad (5)$$

$$z = r \cdot \cos \theta \quad (6)$$



Slika 9. Sferni koordinatni sustav

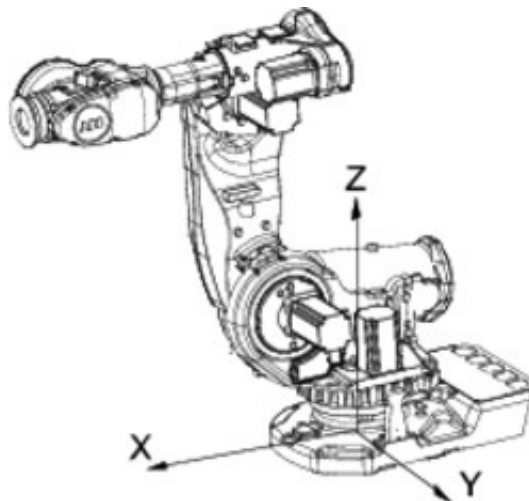
### 3.4. Koordinatni sustavi robota

#### 3.4.1. Koordinatni sustav svijeta

Koordinatni sustav svijeta je fiksiran u prostoru, a ovisno o kompleksnosti virtualne okoline može se nalaziti na baznom koordinatnom sustavu robota.

#### 3.4.2. Bazni koordinatni sustav

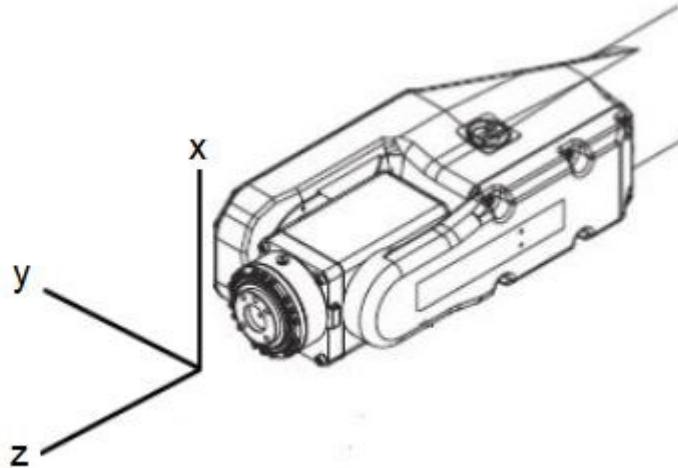
Bazni koordinatni sustav [Slika 10] je zadani koordinatni sustav robota te definira njegovu poziciju u odnosu na koordinatni sustav svijeta. Njegovo je ishodište fiksirano za bazu robota.



Slika 10. Bazni koordinatni sustav robota [7]

### 3.4.3. Koordinatni sustav prirubnice

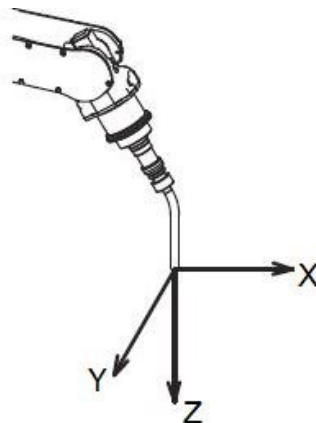
Koordinatni sustav prirubnice [Slika 11] je koordinatni sustav fiksiran na prirubnici robota tako da se njegovo ishodište nalazi u geometrijskoj sredini prirubnice. Ne može se mijenjati, a služi kao polazište za određivanje pozicije i orijentacije alatnog koordinatnog sustava.



Slika 11. Koordinatni sustav prirubnice [8]

### 3.4.4. Alatni koordinatni sustav

Alatni koordinatni sustav [Slika 12] može se slobodno pozicionirati prema potrebi korisnika. Najčešće se postavlja tako da „visi“ u zraku u središnjoj točki alata (eng. TCP – tool center point), ali ne mora biti tako, pogotovo kod kompleksnijih hvataljki i alata gdje nije jasno što je centar prihvata. Definiira se u odnosu na koordinatni sustav prirubnice.

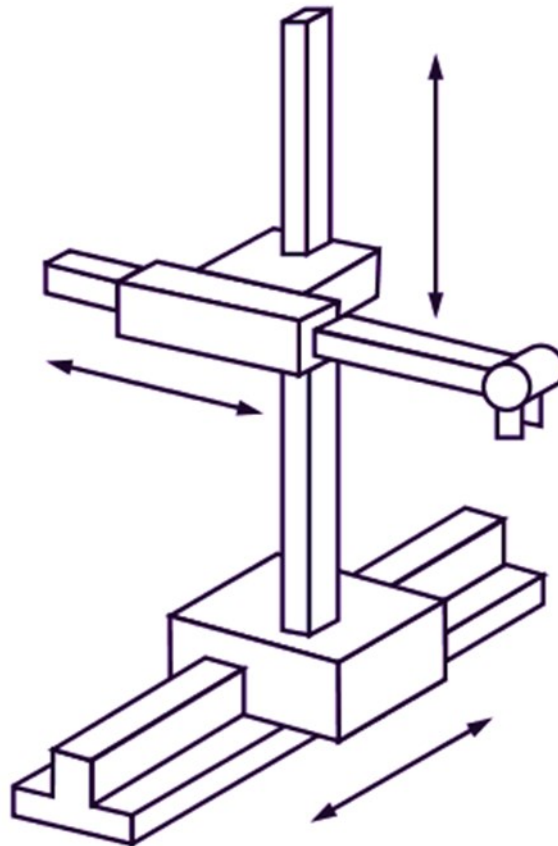


Slika 12. Alatni koordinatni sustav [9]

## 4. VRSTE INDUSTRIJSKIH ROBOTA

### 4.1. Kartezijevi roboti

Kartezijevi roboti [Slika 13] još su poznati pod nazivima linearni roboti i gantry roboti. Često su korišteni u industriji zbog njihove jednostavne kinematike, brzine i preciznosti izvođenja pokreta te niske cijene. Najčešće imaju tri linearne osi koje se pokreću svaka svojim aktuatorom (obično servomotorom), iako iznimno može postojati i rotacijska os na prihvatnici. Neki od primjera korištenja ovakvih robota u industriji uključuju paletizaciju dijelova, slaganje limenih ploča, sklapanje modula, separacija dijelova, pick and place operacije, polaganje ljepila i dr. Zahtijevaju veliki radni prostor ali mogu raditi s velikim teretima. Takav se robot nalazi i na Fakultetu strojarstva i brodogradnje te je predmet ovog diplomskog rada.



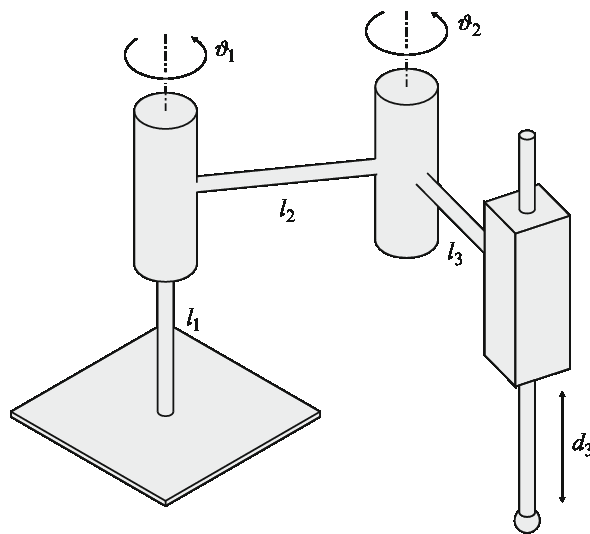
Slika 13. Kartezijev robot [10]

### 4.2. SCARA roboti

SCARA (*Selective Compliance Assembly Robot Arm* ili *Selective Compliance Articulated Robot Arm*) [Slika 14] se sastoje od 2 rotacijska zgloba i jednog linearnog zgloba u vertikalnom



smjeru. U usporedbi s jednakom vrijednim kartezijskim robotom brži su i jednostavniji za montažu pri kojoj ne zahtijevaju puno prostora, ali su zato skuplji.



Slika 14. SCARA robot [11]

### 4.3. DELTA roboti

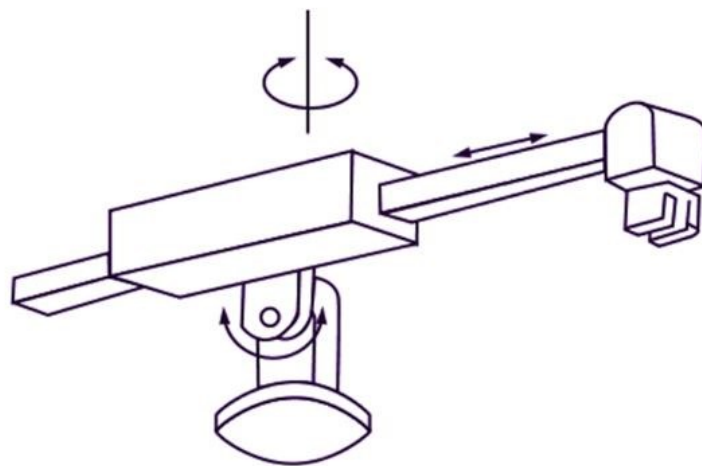
Zbog svojeg izgleda, delta roboti [Slika 15] su poznati još kao spider roboti. Sastoje se od 3 članka koji su povezani na istu prihvatnicu. Najčešće se koriste u pick and place operacijama zbog iznimne brzine, što ne čudi s obzirom na to da je motivacija za njihovu izradu proizašla iz potrebe tvornice čokolade za pakiranjem pralina. Svi se aktuatori nalaze u bazi robota koja je postavljena direktno iznad radnog prostora. Takav položaj aktuatora omogućuje da se članci izrade od laganog materijala, što rezultira malom inercijom i velikom brzinom.



Slika 15. DELTA robot [12]

#### 4.4. Polarni roboti

Roboti s polarnom konfiguracijom ili sferni roboti sastoje se od dva rotirajuća zgloba i jednog linearnog zgloba, što rezultira sfernim radnim prostorom. Najčešće se koriste za injekcijsko prešanje, bojanje, elektrolučno zavarivanje i točkasto zavarivanje. Prvi industrijski robot Unimate o kojem je bilo riječi u prvom poglavlju imao je ovu konfiguraciju. Iako se još uvijek koriste u industriji, polarni roboti [Slika 16] se najčešće zamjenjuju manjim i bržim SCARA robotima.



Slika 16. Polarni robot [13]

#### 4.5. Zglobni roboti

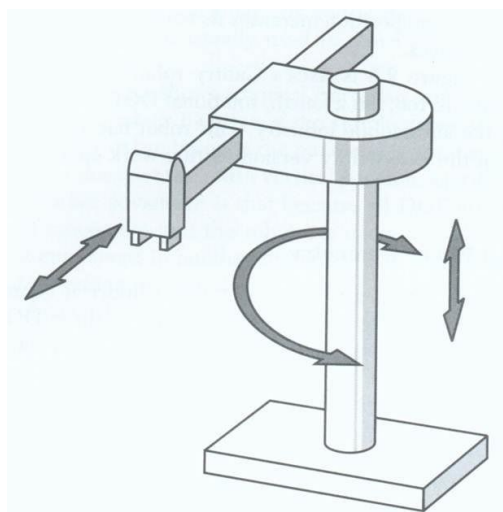
Jedni od najčešćih robota u industriji svojom mehaničkom konfiguracijom podsjećaju na ljudsku ruku. Zglobni roboti [Slika 17] mogu imati različit broj rotirajućih zglobova odnosno stupnjeva slobode, ali najčešće ih ima šest što je dovoljno za obavljanje gotovo svih zadataka jer omogućuje robotu da unutar svog radnog volumena izvršnim elementom dosegne bilo koju točku iz bilo kojeg kuta. Dodatnim stupnjevima slobode robot postaje kinematički redundantan. To znači da unutar svog radnog volumena može dosegnuti bilo koju točku iz bilo kojeg kuta na više od jednog načina. Koriste se u velikom broju industrija, ali od posebnog su značaja u autoindustriji zbog toga što mogu izvoditi vrlo složene pokrete. Logično je da su zbog svoje kompleksnosti složeniji za programiranje i imaju složenu kinematiku.



Slika 17. Zglobni robot sa 6 stupnjeva slobode gibanja [14]

#### 4.6. Cilindrični roboti

Najčešće se koriste za montažu, točkasto zavarivanje i rukovanje kod strojeva za tlačno lijevanje, a imaju jedan rotacijski i dva linearna zgloba. Vertikalni linearni pomak se najčešće postiže pneumatskim cilindrom, dok se rotacija odvija pomoću servomotora i prijenosnika. Vrlo su kompaktni i mehanički jednostavni pa je malo vremena potrebno za sastavljanje i održavanje.



Slika 18. Cilindrični robot [15]

## **5. POGON ROBOTSКИH ČLANAKA**

Najelegantnije rješenje za pogon robotskih članak predstavljaju razne vrste elektromotora. Mogu biti pogonjeni istosmjernom ili izmjeničnom strujom, s ili bez četkica te se mogu koristiti servomotori s enkoderom ili koračni motori. Moguće ih je primjenjivati za rotirajuće zglobove kao i linearne. Vrlo su tihi i jednostavni za održavanje te omogućuju visoku preciznost i ponovljivost, a imaju i nisku cijenu.

Hidraulički pogoni se koriste kod robota koji moraju raditi s velikim teretima, a omogućuju zadovoljavajuću brzinu rada i uslijed nestlačivosti ulja osiguravaju točan položaj. Nedostaci hidrauličkog pogona uključuje potrebu za filtriranjem ulja, mogućnost onečišćenja i curenja ulja, visoke razine buke i visoku cijenu.

Pneumatski pogoni rade pomoću komprimiranog zraka koji mora biti filtriran i sušen kako ne bi došlo do narušavanja performansi pogona. Oni su čišći od hidrauličkih, jeftiniji su i mogu postići veće brzine, no također su bučni. Ovaj se pogon često koristi za vertikalni pomak kod cilindričnih robota.

## 6. IZVRŠNI ELEMENTI ROBOTA

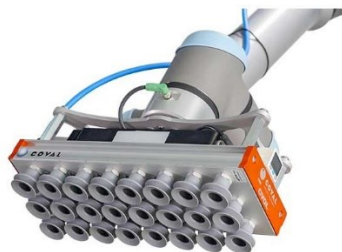
Izvršni elementi robota su alati, uređaji ili sustavi koji se pričvršćuju na prirubnicu robota i služi robotu za direktnu interakciju s predmetima u njegovoj okolini. Odabir izvršnog elementa ovisi o zadatku kojeg robot mora obavljati. Ukoliko robot treba obavljati više različitih zadataka, moguće je promijeniti njegov izvršni član ili koristiti adapter koja na sebi ima pričvršćeno više izvršnih članova, te se njegovom rotacijom odabire onaj koji je u tome trenutku adekvatan. Neki od mogućih izvršnih članova su: hvataljke, magneti, kamere, alati za bušenje, odvijači, alati za zavarivanje, pištolji za boju, senzori sile, alati za rezanje, alati za brušenje. Među najčešćim izvršnim članovima su hvataljke s obzirom na učestalost potrebe za pomicanjem predmeta rada, o čijim karakteristikama ovisi konkretan odabir hvataljke. Prema izvedbi hvataljke dolaze u velikom broju varijanata, ali mogu se podijeliti na:

- Krute hvataljke - koriste se za prihvat čvrstih predmeta s ravnim plohama prihvaća
- Mekane hvataljke - koriste se za prihvat osjetljivih i lako lomljivih predmeta, te predmeta nepredvidljivih i složenih oblika
- Posebne hvataljke - izrađuju se za specifične namjene

Prema načinu rada mogu se podijeliti na vakuumske, pneumatske, hidrauličke i električne.

### 6.1. Vakuumske hvataljke

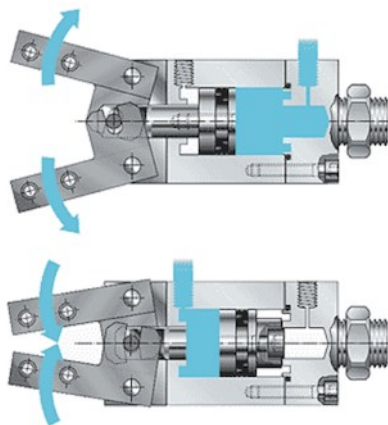
Vakuumske hvataljke [Slika 19] obično se sastoje od niza usisnih kapica iz kojih se nakon prijanjanja na površinu pumpom isisava zrak. Time se postiže razliku u tlaku i omogućuje podizanje i manipulacija predmeta. Zbog načina rada vakuumske hvataljke mogu se u pravilu primjenjivati samo na predmetima s ravnim površinama.



Slika 19. Vakuumska hvataljka [16]

## 6.2. Pneumatske hvataljke

Pneumatske hvataljke [Slika 20] se najčešće koriste za podizanje i pomicanje manjih predmeta jer imaju ograničenu silu hvatanja. Predmeti se hvataju s dva, tri ili više prstiju čiji se mehanizam otvaranja i zatvaranja pogoni komprimiranim zrakom. Ovakve su hvataljke jeftine za proizvesti, ali nedostatak im je ograničena snaga i potreba za pumpom i provođenjem cjevčica do hvataljke.



Slika 20. Način rada pneumatske hvataljke [17]

## 6.3. Električne hvataljke

Električne hvataljke [Slika 21] pogodne su za rad s osjetljivim i predmetima rada malih dimenzija. Princip rada je sličan pneumatskim hvataljkama – za prihvata predmeta koriste se prsti hvataljke, samo što su u ovom slučaju oni pogonjeni elektromotorom. U pravilu imaju manju silu prihvata od pneumatskih hvataljki, ali finiju kontrolu brzine, sile i pozicije prstiju.



Slika 21. Električna hvataljka [18]

#### 6.4. Hidrauličke hvataljke

Hidrauličke hvataljke [Slika 22] su najbolji izbor za hvatanje i manipulaciju velikih i teških predmeta rada. Mnogo su složenije od ostalih jer je za njihovu primjenu potrebna pumpa s pripadnim instalacijama koje ne samo što otežavaju montažu ovakve hvataljke, već i dugoročno dovode do neizbježnih problema za curenjem radne tekućine koji mogu biti skupi za popravak.



Slika 22. Hidraulička hvataljka [19]

## 7. SENZORI U ROBOTICI

Senzori su elementi koji zahvaljujući svojim električkim, kemijskim ili mehaničkim značajkama podražaje ili stanja u okolini pretvaraju u informaciju koja je značajna za robota, i time obogaćuju njegovu svijest o okolini i mogućnost interakcije s njom. Ulazna veličina se najčešće pretvara u napon. Senzore možemo podijeliti u dvije grupe: unutarnji senzori i vanjski senzori. Unutarnji senzori mjere veličine unutar samog robota, primjerice pozicija, akceleracija, brzina, moment motora i slično. Vanjski senzori poput kamera, senzori blizine i senzora sile sakupljaju informacije o okolini robota.

Senzorima se mogu definirati neke zajedničke značajke:

- Osjetljivost - omjer promjene izlazne i ulazne veličine.
- Linearnost - opisuje kako se promjena izlazne veličine za istu promjenu ulazne veličine mijenja duž mjerljivog raspona.
- Vrijeme odziva - vrijeme potrebno da se detektirana promjena ulaza u sustav obradi i pošalje robotu.
- Preciznost – razlika između stvarne i mjerene vrijednosti.
- Ponovljivost - razlike očitanih vrijednosti među ponovljenim mjerenjima.
- Mjerni raspon – razlika između najveće i najmanje vrijednosti koju senzor može detektirati.
- Rezolucija - najmanja primjetljiva promjena ulazne veličine.

Neki od najčešće korištenih senzora su opisani u nastavku.

### 7.1. Svjetlosni senzori

Svjetlosni senzori detektiraju svjetlost te ovisno o njenom intenzitetu stvaraju različit napon. Jedan od načina za to postići je korištenje fotootpornika, čiji je otpor obrnuto proporcionalan intenzitetu svjetlosti koji na njega pada. Drugi način je korištenje fotonaponskih ćelija koje pretvaraju energiju sunčeve svjetlosti u električnu energiju, ali njihova primjena nije česta kod industrijskih robota.



## 7.2. Senzori zvuka

Senzori zvuka su u osnovi mikrofoni koji detektiraju zvuk i stvaraju napon proporcionalan jačini zvučnog podražaja. U složenijoj varijanti može se koristiti i za raspoznavanje glasovnih naredbi. Zbog niskog napona potrebno je koristiti pojačalo da bismo dobili mjerljive vrijednosti.

## 7.3. Senzori blizine

Senzor blizine može detektirati okolne predmete bez direktnog kontakta s njima. Dva su osnovna dijela, odašiljač koji šalje zvučne ili svjetlosne impulse i prijemnik koji analizira povratni signal. Postoji mnogo izvedbi ovog senzora, a one koje se najčešće koriste u robotici su :

- Infracrveni senzor – odašiljač šalje infracrveni svjetlosni signal koji se reflektira kada naiđe na prepreku. Mogu se koristiti i za mjerenje udaljenosti.
- Ultrasonični senzor – Ovi senzori kreiraju zvučne signale visokih frekvencija. Također se mogu koristiti za mjerenje udaljenosti.
- Fotootpornik – Kao senzor blizine moguće je koristiti i fotootpornik, s obzirom na to da se intenzitet svjetlosti koji pada na fotootpornik smanjuje kada se on nalazi u neposrednoj blizini prepreke.

## 7.4. Taktilni senzori

Kao što ime predlaže, taktilni senzori su osjetljivi na kontakt senzora s drugim predmetom. Omogućuju robotu da bude nježan pri interakciji s okolinom. Mogu se podijeliti na dvije grupe: Senzori dodira obično imaju binarni način rada, šalju povratni signal kada su u dodiru s nekim predmetom. Na taj signal robot se može primjerice zaustaviti . Senzori sile vraćaju povratni signal u određenom rasponu napona, ovisno o sili koja djeluje na senzor. Jedan od primjera korištenja je pri montaži, kada je potrebno postići određenu silu.

## 8. ROBOT IZ ZADATKA

Riječ je o kartezijskom robotu s 3 linearna zgloba i jednim rotacijskim zglibom [Slika 23]. Robot se nalazi u sjevernoj zgradi na Fakultetu strojarstva i brodogradnje u Zagrebu u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava. Svrha robota je da se mladi inženjeri educiraju i nauče osnovama rada i upravljanja jednim kartezijskim robotom. Dimenzije postolja robota i pomaci osi dani su u tablici 1.

**Tablica 1. Dimenzije postolja robota i pomaci osi**

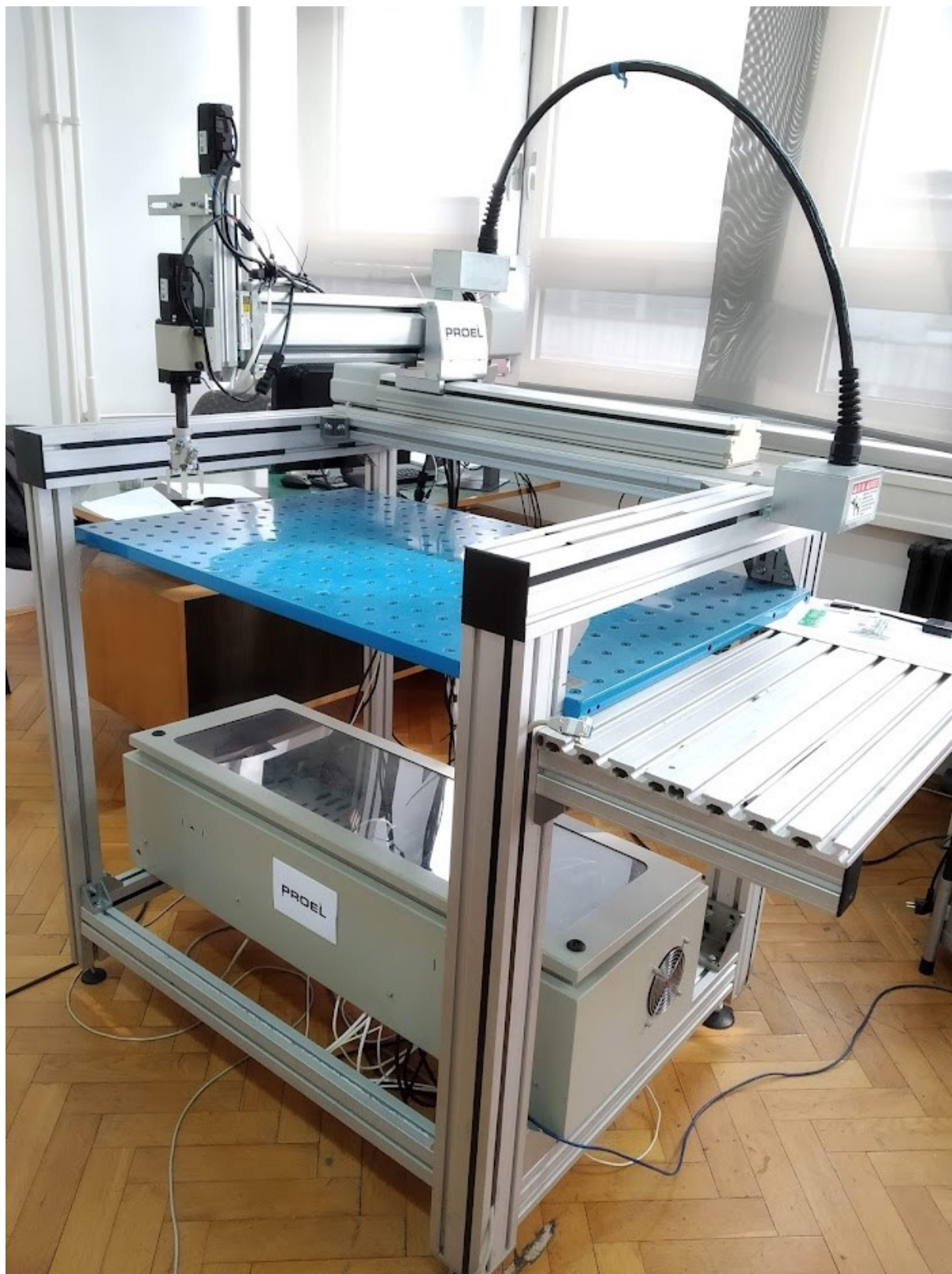
Visina	110 cm
Širina	108 cm
Dužina	90 cm
Pomak u smjeru osi x	600 mm
Pomak u smjeru osi y	400 mm
Pomak u smjeru osi z	100 mm

Pomičnost linearnih osi ostvarena je pomoću vretena i servomotora bez prijenosnika s 20-bitnim enkoderom, dok je rotacijska os izvedena pomoću harmoničkog prijenosnika s omjerom 1:25. Vertikalna os je specifična po tome što je pod konstantnim utjecajem gravitacije, pa se na njoj nalazi kočnica koja sprječava propad z osi u slučaju gubitka napona.

Servomotorima se upravlja PLC-om preko 4 frekvencijska pretvarača, svaki dodijeljen jednom servomotoru. Programsko rješenje rađeno je u ljestvičastom (*engl. ladder*) dijagramu u softveru „CANOpen Builder“ koji je povezan s upravljačkim programom rađenim u programskom jeziku Python u svrhu jednostavnijeg upravljanja robotom, mogućnosti izrade grafičkog sučelja te povezivanja s nekim drugim softverom (u ovom slučaju RoboDK). Snage motora dane su u tablici 2.

**Tablica 2. Snage servomotora za svaku os**

Snaga servomotora za os x	200 kW
Snaga servomotora za os y	200 kW
Snaga servomotora za os z	100 kW
Snaga servomotora za os R	100 kW



**Slika 23. Kartezijev robot u Laboratoriju za projektiranje izradbenih i montažnih sustava**

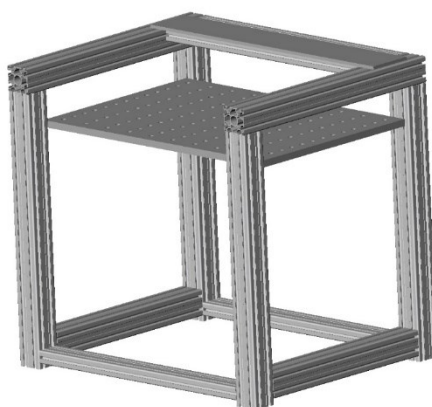
## 9. MODELIRANJE ROBOTA

Sve komponente i njihovi međusobni odnosi najprije su precizno izmjereni kako bi se robot vjerno mogao prikazati u virtualnom okruženju. Odabrani CAD softver za modeliranje bio je Solidworks. Postolje se sastoji od 8 aluminijskih profila dimenzija 90x90, te 2 aluminijska profila dimenzija 90x45 [Slika 24] koji služe za dodatno ukrućenje podnožja postolja. Ovi su dijelovi postolja kreirani pomoću opcije *extrude*, a zatim su dijelovi povezani u sklop.



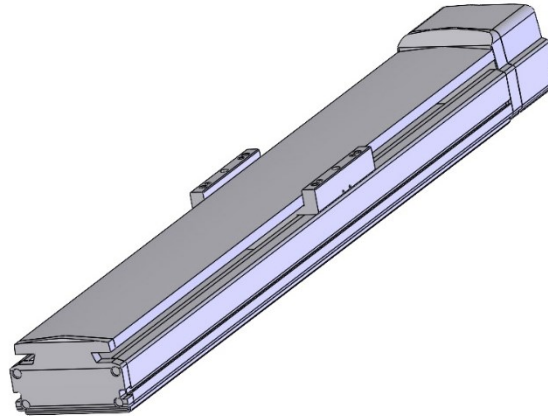
**Slika 24.** Aluminijski profil 90x90 (lijevo) i aluminijski profil 90x45 (desno)

Na postolju [Slika 25] se još nalazi ploča na koju se pričvršćena jedna od osi robota ispod koje se nalazi još jedan aluminijski profil kako bi se smanjilo momentno opterećenje, te radni stol koji predstavlja površinu na kojoj se nalazi predmeti rada robota.



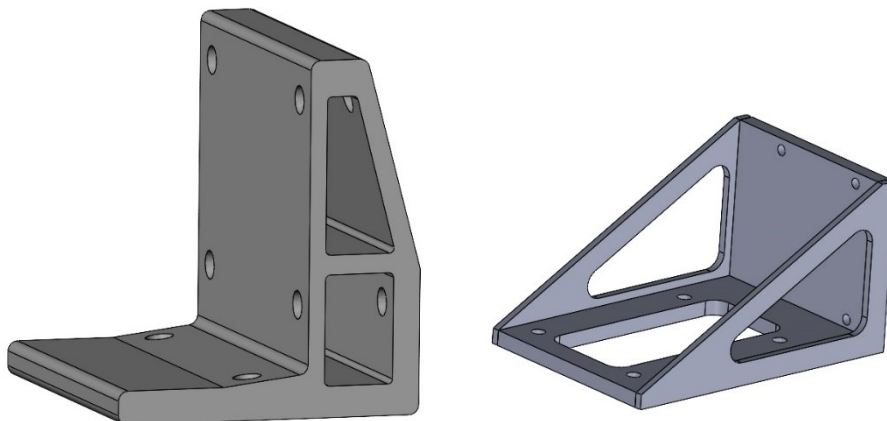
**Slika 25.** Postolje robota s radnim stolom

Sve tri osi [Slika 26] su u osnovi iste ali različitih dimenzije. Tako je os x duljine 760 mm, os y duljine 560 mm a os z je najmanja s duljinom 200 mm. Na osima se nalazi vodilica koja se kreće po vretenu.

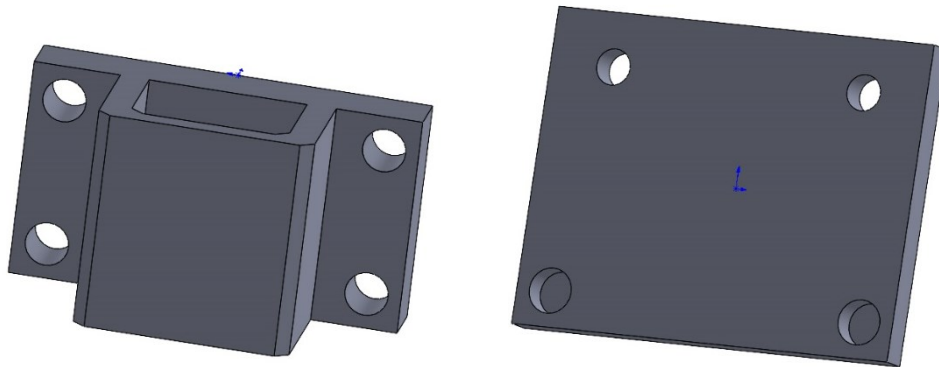


**Slika 26. Os robota**

Na vodilicu osi x je postavljen nosač [Slika 27 lijevo] koji je izrađen pomoću opcije *extrude* na osnovu zadanog profila. Nosač se kreće po vodilici a na sebi nosi drugu os robota. Na nosaču se nalazi šupljine zbog mogućnosti pristupa alatom i smanjenja mase. Na y os je pričvršćen još jedan nosač [Slika 27 desno] na kojeg je fiksirana vodilica z osi po kojoj ona klizi. Nepotreban materijal je uklonjen zbog smanjenja mase, a veze se ostvaruju vijcima. Izrađen je i nosač servomotora koji pokreće rotacijsku os motora, te pločica na koju je isti pričvršćen [Slika 28].

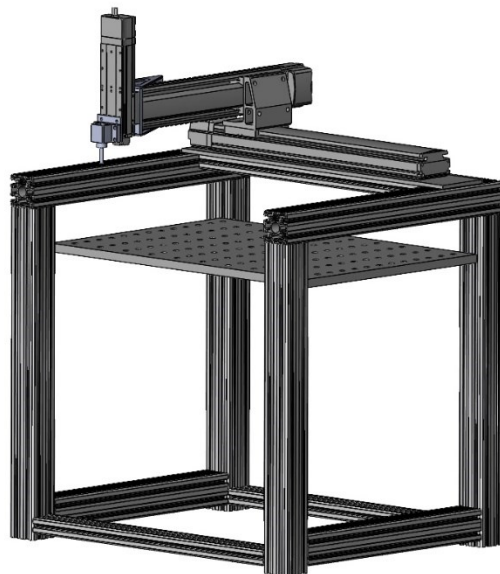


**Slika 27. Nosač y osi (lijevo) i nosač klizača z osi (desno)**



**Slika 28. Nosač servomotora (lijevo) i pločica (desno)**

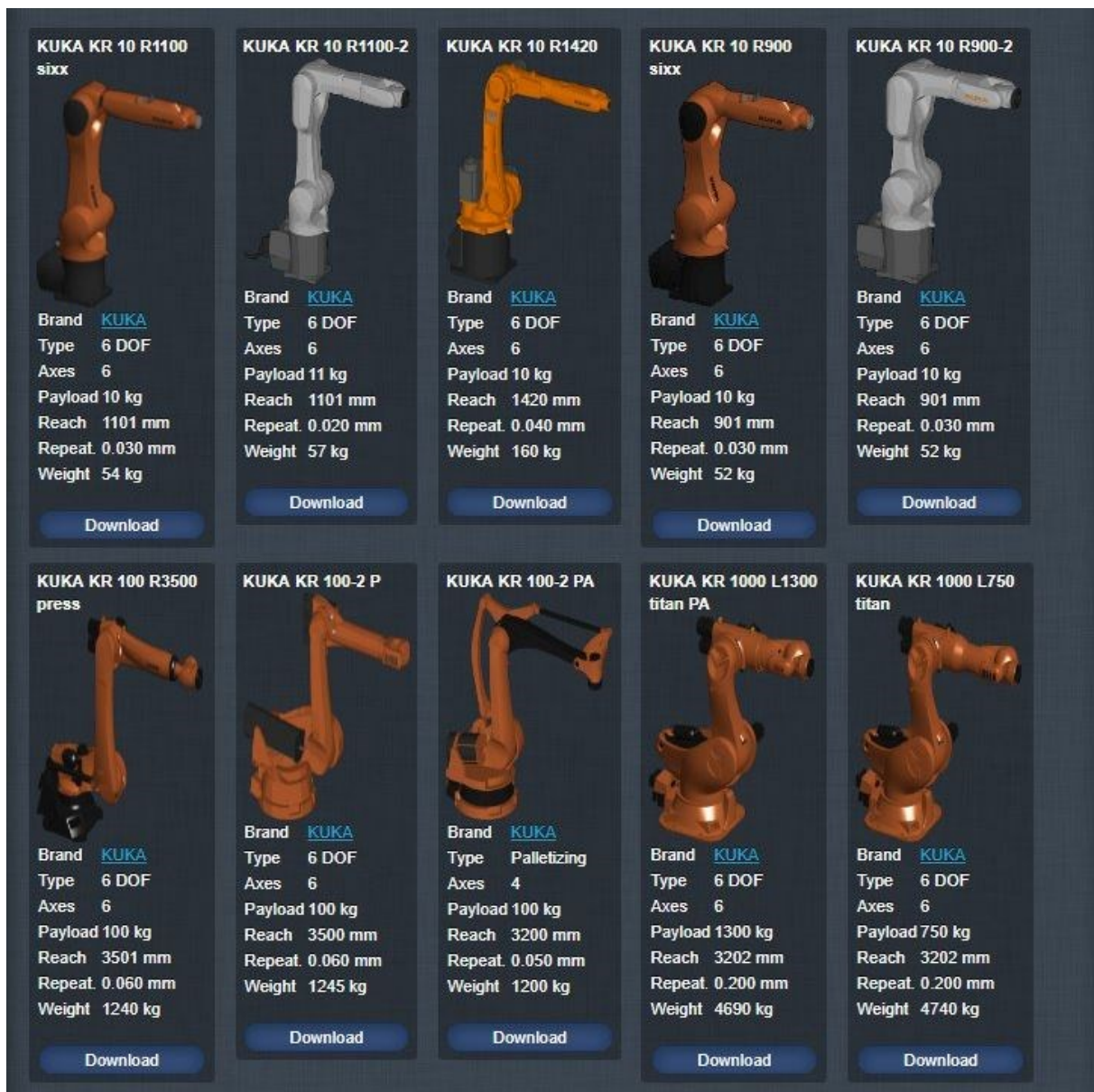
Kompletan sklop robota prikazan je na slici 29. Robot je prije implementacije u RoboDK postavljen u nultu poziciju.



**Slika 29. 3D sklop robota**

## 10. RoboDK

RoboDK je računalni program koji omogućuje simulaciju robotskih programa u virtualnom okruženju. Moguće je također generirati programski kod za određeni zadatak te ga poslati robotu, kao i ostvariti direktnu komunikaciju s robotom te u realnom vremenu pokretati robota iz samog softvera. RoboDK u svojoj knjižnici posjeduje preko 600 robota [Slika 30] raznih proizvođača čiji se modeli mogu učitati u softver, a za mnoge od njih postoje i upravljački programi za povezivanje s pravim robotom.



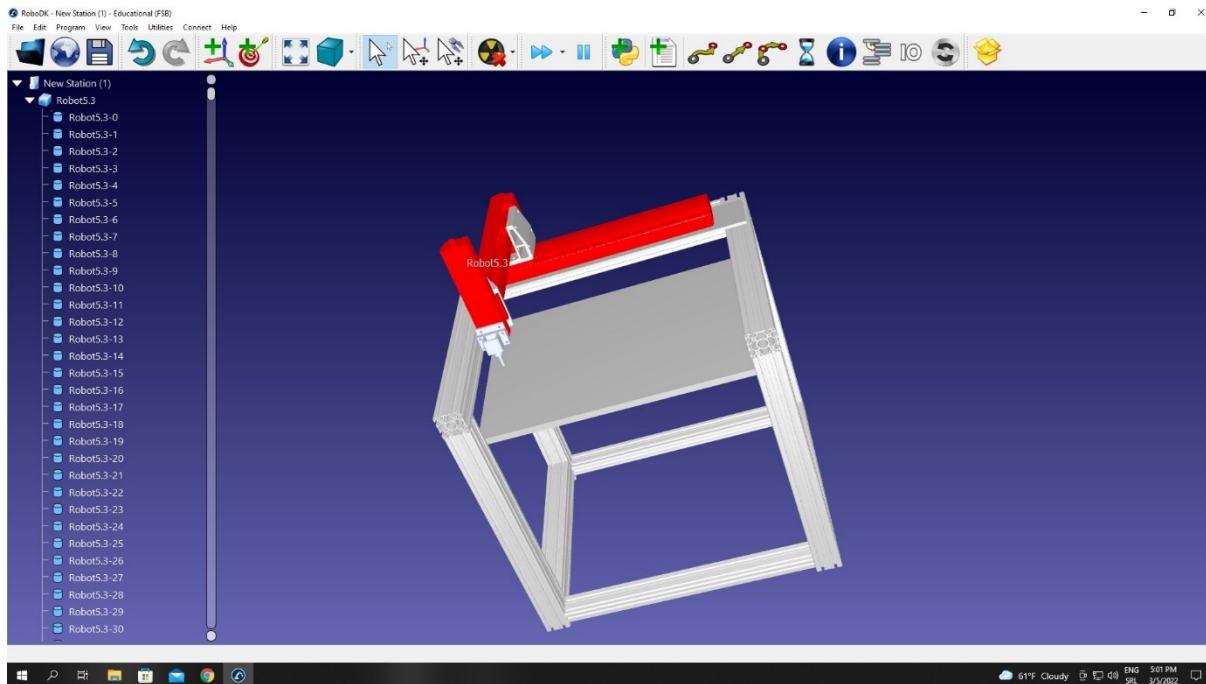
Slika 30. Isječak knjižnice robota u softveru RoboDK

Moguće je također (uz određena ograničenja) kreirati vlastiti mehanizam koji će omogućiti simulaciju robota koji ne postoji u RoboDK knjižnici, te isprogramirati vlastiti upravljački program za njegovo upravljanje. To se može napraviti u programskom jeziku Python, na kojem je baziran cijeli softverski paket.



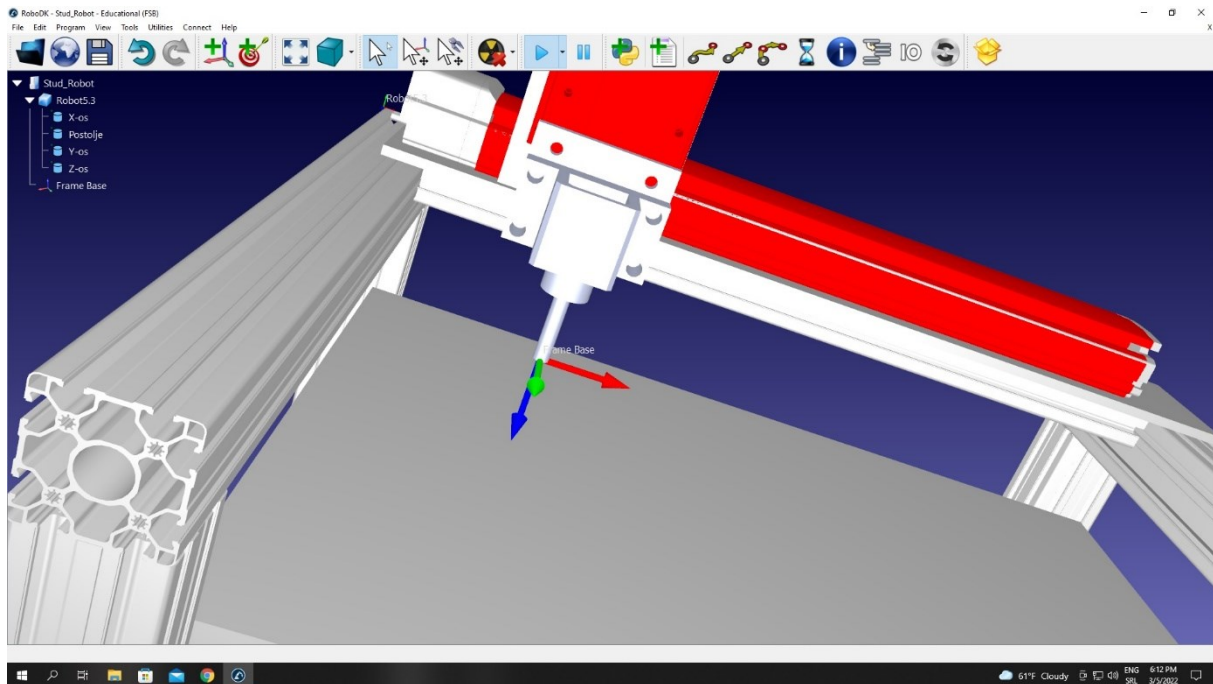
## 11. IMPLEMENTACIJA ROBOTA U RoboDK

Robota modeliranog u Solidworksu ili njemu sličnom programu se u step formatu može otvoriti u RoboDK, međutim taj je robot prikazan kao jedan jedini objekt. Da bismo stvorili mehanizam potrebno je robota podijeliti u nekoliko smislenih cjelina. Proces je pomalo nezgrapnan, potrebno je naredbom *split* podijeliti robota na stotine djelića (kao što je vidljivo na lijevom izborniku) koji se zatim označavaju i povezuju po potrebi.



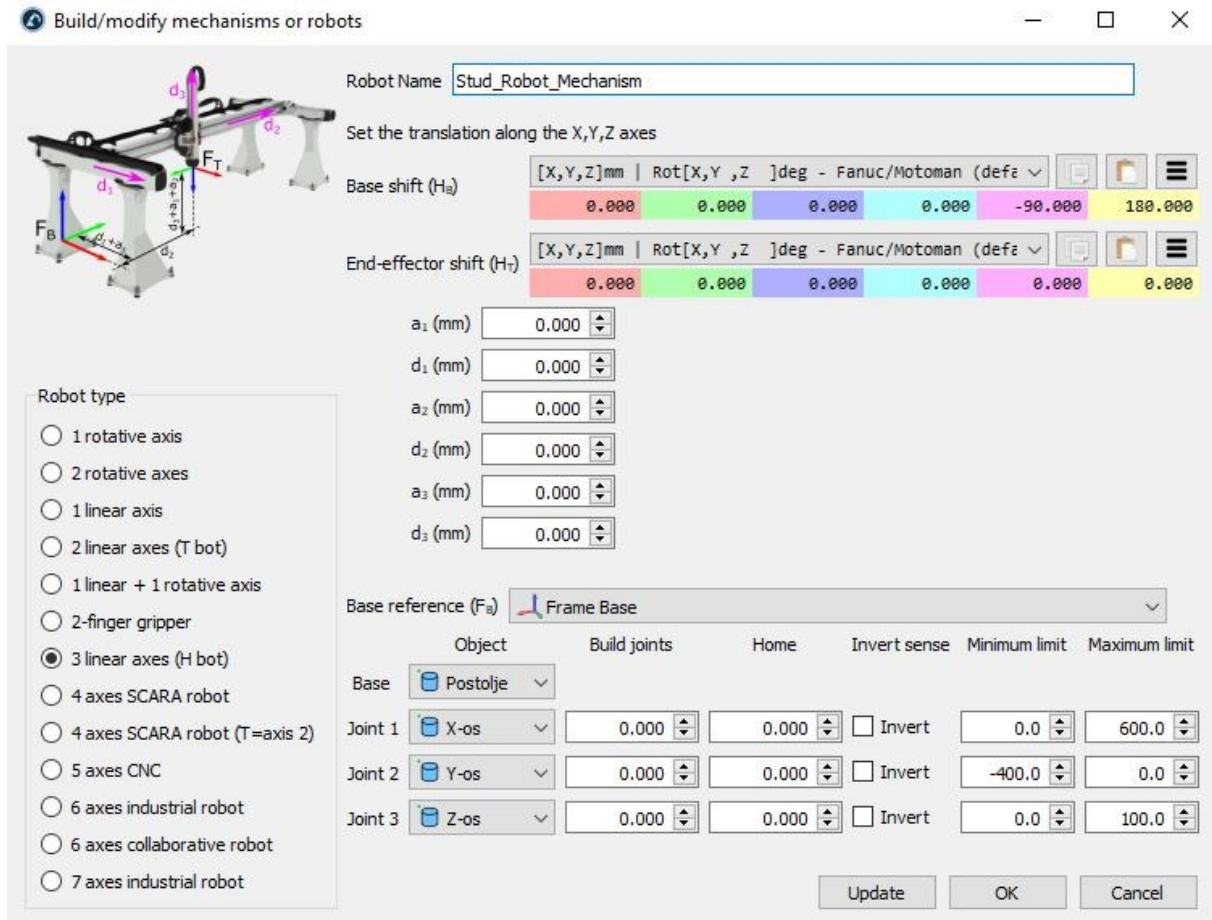
**Slika 31. Model robota u programu RoboDK nakon naredbe *split***

Prije kreiranja mehanizma još je pametno kreirati koordinatni sustav čije se ishodište nalazi na prihvatnici [Slika 32], kako bi sve bilo spremno za pristupiti izradi mehanizma.



**Slika 32. Koordinatni sustav prihvatnice**

Kada je taj posao obavljen možemo pristupiti izradi mehanizma [Slika 33] odnosno kinematike. RoboDK omogućuje 13 različitih tipova mehanizama u verziji programa koja je korištena pri izradi ovog rada. Kartezijski robot s jednom rotacijskom osi nije među njima, već je ta mogućnost dodana u novijim verzijama uz pristup razvojnom načinu rada. Stoga je ovaj robot napravljen kao klasični kartezijski robot s tri linearne osi, još poznat pod nazivima H-bot ili Gantry robot.

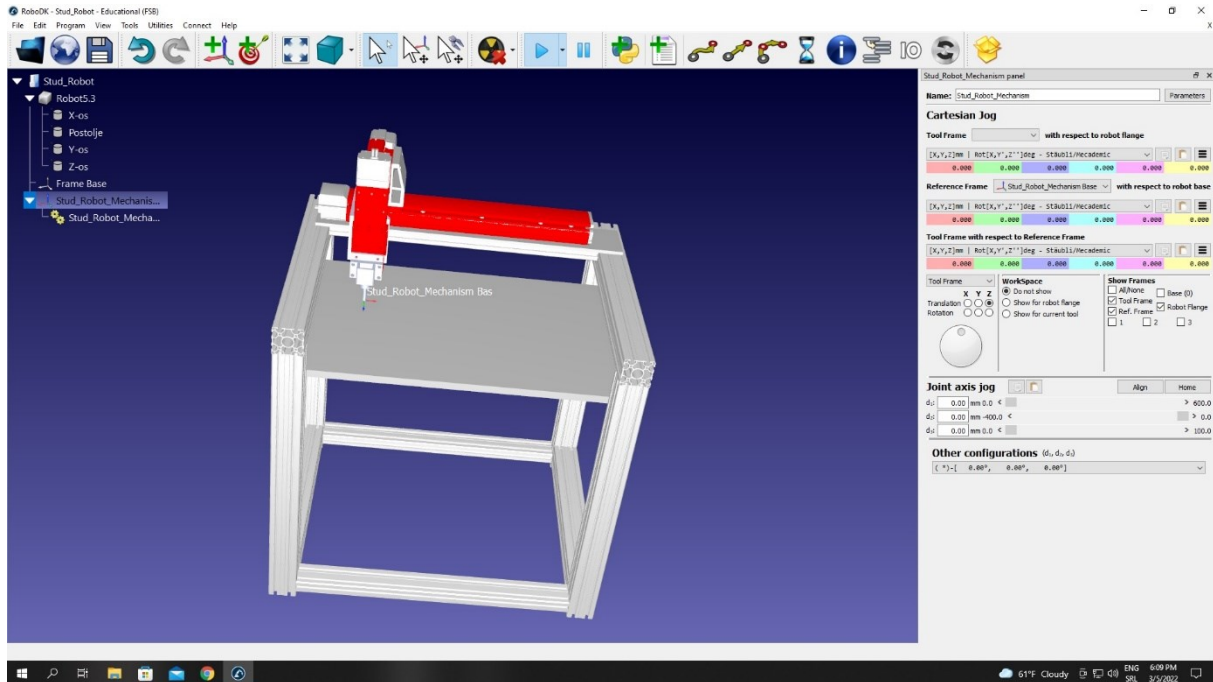


**Slika 33. Prozor za stvaranje mehanizma**

U izborniku je potrebno odabrati prethodno kreirane dijelove robota u skladu sa slikom u gornjem lijevom kutu izbornika. Također je potrebno ograničiti pomake svake od osi kako bi odgovarale mogućnostima robota u stvarnom svijetu.

U stvarnosti je kinematika robota bazirana na lijevokretnom koordinatnom sustavu, kojeg je u RoboDK nemoguće napraviti. No iz perspektive funkcionalnosti to se može vrlo lako riješiti softverskim putem, tako da upravljački program samo zamijeni predznak jednoj od osi i takvu izmijenjenu šalje robotu, pa je tako i napravljeno.

Sada je mehanizam stvoren te se u desnom izborniku mogu pomicati sve osi robota [Slika 34], a robot se može spremati i na jednostavan način koristiti u virtualnom okruženju kao i svi ostali roboti u knjižnici robota RoboDK.



Slika 34. Robot s kreiranim mehanizmom

### 11.1. Vrste putanja gibanja

Unutar RoboDK softvera moguće je dati naredbu pomaka na tri načina:

- Zglobni pomak može se smatrati osnovnim tipom pomaka jer najbrže pomiče robota od polazne do ciljne točke. Putanja ne mora uvijek biti unaprijed poznata, ali najčešće nije linearna.
- Linearni pomak rezultira linearnim pomakom robota odnosno vrha alata, i u pravilu se koristi samo kada je striktno potrebna takva vrsta pomaka. Za razliku od zglobnog pomaka, može doći do problema sa singularnosti.
- Kružni pomak zahtjeva 3 definirane točke i prostoru, a odvija se tako da se robot kreće u luku od polazišne točke kroz središnju točku do ciljne točke.

## 12. UPRAVLJAČKI PROGRAM

RoboDK je softver pomoću kojeg se može upravljati stotinama robota od raznih proizvođača, a neminovno je da svaki od proizvođača ima drugačije zahtjeve za upravljanje robota. Stoga postoje upravljački programi koji povezuju izlazni kod iz softvera RoboDK i pretvaraju ga u kod koji je razumljiv robotu i koji može njime upravljati. Drugim riječima, upravljački program omogućuje komunikaciju između hardvera i softvera. Kod upravljačkog programa s komentarima dan je u prilogu, a ovdje je objašnjen glavni dio koda koji daje naredbu za pomake.

```
# Funkcija koju poziva RoboDK prilikom pomicanja robota
def RunCommand(cmd_line):
    global ROBOT_IP
    global ROBOT_PORT
    global ROBOT
    global ROBOT_MOVING

    # funkcija služi za pretvorbu string varijabli u float varijable
    def line_2_values(line):
        values = []
        for word in line:
            try:
                number = float(word)
                values.append(number)
            except ValueError:
                pass
        return values

    cmd_words = cmd_line.split(' ') #rezanje teksta na dijelove
    cmd = cmd_words[0] #U cmd se sprema naredba (npr move linear, move
joints...)
    cmd_values = line_2_values(cmd_words[1:]) #u cmd_values se spremaju samo
koordinatne tipa float
    n_cmd_values = len(cmd_values) #broj elemnata u cmd_values
    n_cmd_words = len(cmd_words) #broj elemnata u cmd words
    print_message(str(cmd_words)) #ispis poruke (za test)
    received = None

    #Stvaranja rječnika s koordinatama
    cord = {
        "X": cmd_values[0],
        "Y": -cmd_values[1], #negativna vrijednost jer je pravi robot
napravljen u lijevokretnom koordinatnom sustavu
        "Z": cmd_values[2],
        "Rz": 0, }

```

### Slika 35. Dio koda upravljačkog programa za stvaranje rječnika s koordinatama

Poziva se funkcija RunCommand s parametrom cmd\_line. U toj su varijabli zapisane željene koordinate robota od strane RoboDK u tekstualnom obliku. Pomoću naredbe cmd\_line.split to

pretvorimo u listu s varijablama tipa string. Zatim se poziva funkcija `line_2_values` koja string varijable pretvara u float varijable.

Naposljetku se numeričke vrijednosti koordinata zapisuju u rječnik te se kasnije u tom obliku šalju robotu

```
if cmd_line == "":
    # Preskoči ako nije dana naredba
    return

elif n_cmd_values >= nDOFs_MIN and cmd_line.startswith("MOVJ"):
    # Obnova statusa
    UpdateStatus(ROBOTCOM_WORKING)
    # Pozivanje funkcije .move_linear na strani robota s parametrom coord
    ROBOT.move_linear(relative=False, **cord)
    ROBOT_MOVING = True

elif n_cmd_values >= nDOFs_MIN and cmd_line.startswith("MOVL"):
    # Obnova statusa
    UpdateStatus(ROBOTCOM_WORKING)
    ROBOT_MOVING = True
    print_message(str(cord)) #ispis poruke
    # Pozivanje funkcije .move_linear na strani robota s parametrom coord
    ROBOT.move_linear(relative=False, **cord)
    UpdateStatus(ROBOTCOM_READY)
```

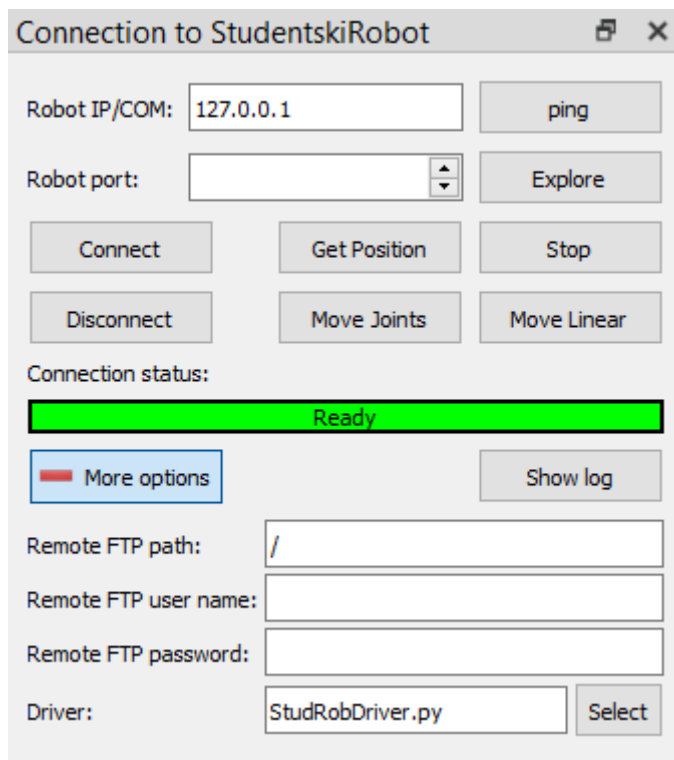
### Slika 36. Dio koda upravljačkog programa za pozivanje funkcija pomaka na strani robota

U ovom se dijelu koda pozivaju linearni ili zglobovi pokret robota u ovisnosti o početnoj riječi naredbe do koje se dolazi funkcijom `cmd_line.startswith()`. Status robota se obnovi te se pozove funkcija `move_linear` ili u koju se pošalje prethodno dobiveni parametar `cord`. Ta funkcija se nalazi na strani robota i rezultira njegovim pomakom.

### 13. SIMULACIJA

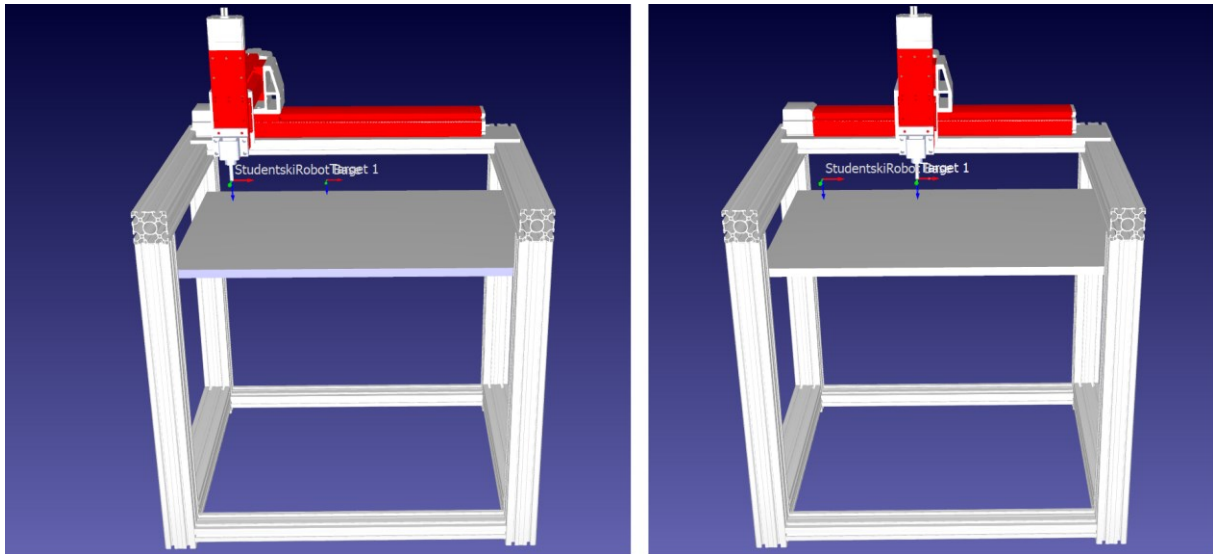
Funkcionalnost izrađenog modela robota, njegove kinematike i pripadajućeg upravljačkog programa testirani su simulacijom.

Prije svega je potrebno ostvariti vezu s robotom putem bežične mreže. Desnim klikom na mehanizam otvara se padajući izbornik te se odabire naredba *connect to robot*. U novootvorenom izborniku [Slika 37] potrebno je upisati IP adresu robota i odabrati željeni upravljački program. Klikom na gumb *Connect* u upravljačkom programu se pozivaju naredbe zadužene za komunikaciju te ako je sve prošlo bez poteškoća, RoboDK će javiti da je spreman za daljnje naredbe.

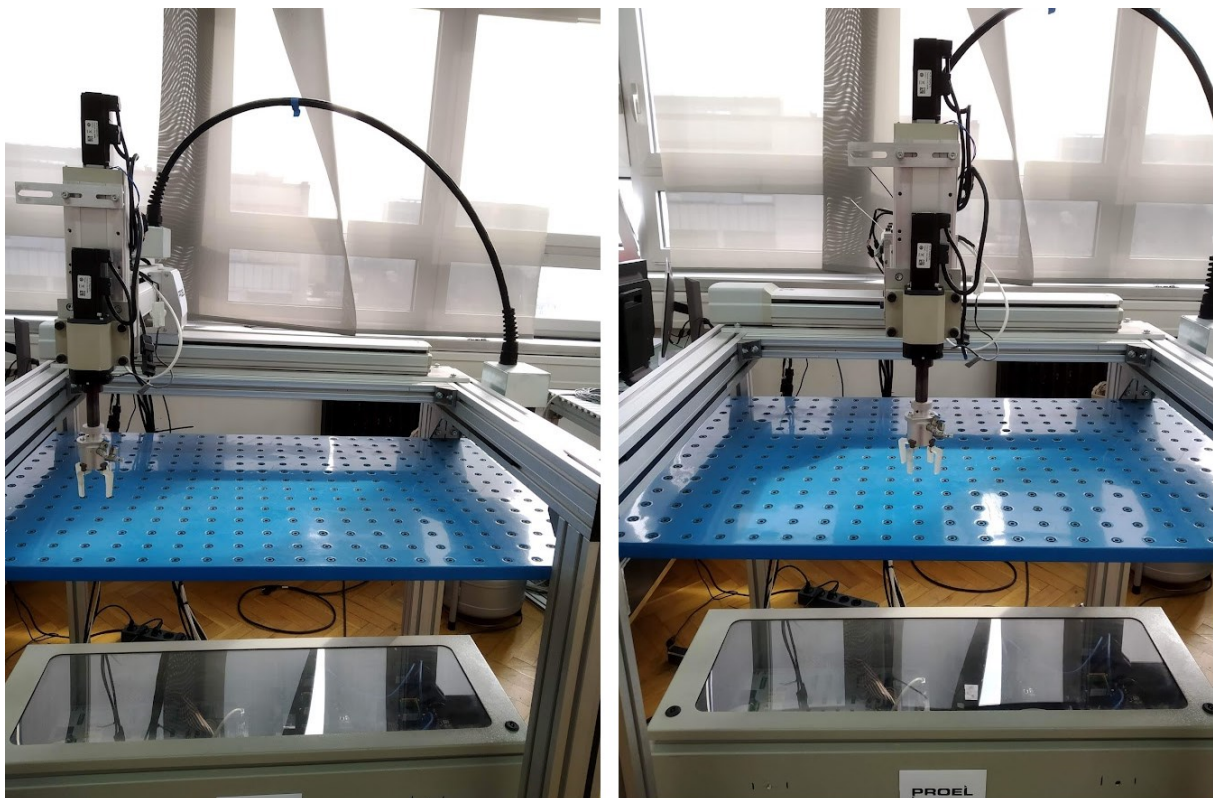


Slika 37. Izbornik za povezivanje s robotom

U softveru RoboDK stvorena je ciljna točka robota koja se nalazi na udaljenosti 200 mm u smjeru x osi u odnosu na nultu poziciju, te je dana naredba za linearni pomak. Početna i završna stanja virtualnog odnosno fizičkog robota prikazana su na slikama 38 i 39. Vidljivo je da se robot ponaša prema očekivanju, odnosno pomaknuo se 200 mm u smjeru osi x, baš kao i robot u virtualnom okruženju.



**Slika 38.** Stanje robota u virtualnoj okolini na početku simulacije (lijevo) i na kraju simulacije (desno)



**Slika 39.** Stanje robota u stvarnoj okolini na početku simulacije (lijevo) i na kraju simulacije (desno)



U kontekstu ovog rada korisno je spomenuti koncept digitalnog blizanca (*engl.* digital twin), relativno nov pojam u svijetu tehnologije koji se pojavio početkom 21. stoljeća. On u kontekstu robotike predstavlja ideju potpuno vjerne reprezentacije robota u virtualnoj okolini u svakom vremenskom trenutku. Za izvedbu digitalnog blizanca potrebno je putem senzora i kontrolnog sustava ostvariti i konstantno održavati povratnu vezu prema softveru. Grafički prikaz može se dodatno obogatiti raznim informacijama kao što su stanje i zdravlje komponenata, učinkovitost ili usporedba s prijašnjim radnjama. Pomoću tih informacija sustav se može analizirati, simulirati i optimirati. Moguće je iskoristiti i proširenu stvarnost (*engl.* augmented reality) za prikaz virtualnog robota u stvarnoj okolini.

U ovom se radu informacije o poziciji robota šalju samo u jednom smjeru (od softvera prema robotu) tako da ne možemo govoriti o digitalnom blizancu, ali je ostvaren prvi korak u tom smjeru.

## 14. ZAKLJUČAK

U industrijskom okruženju može doći do situacije u kojoj je potrebno izraditi nestandardnog robota za odrađivanje specifičnih zadataka. U ovom je radu pokazano kako jednog takvog robota modelirati i vjerno prikazati u virtualnom svijetu te ostvariti kinematiku kako bi se mogao simulirati njegov rad prije puštanja u pogon. Također je ostvarena komunikacija s fizičkim robotom te je izrađen upravljački program putem kojeg se simulirani pokreti ostvaruju u realnom okruženju.

Ovaj rad pokazuje interdisciplinarnost inženjerskog posla jer su za potpunu izradu, programiranje i upravljanje nestandardnog robota potrebna znanja iz strojarstva, elektrotehnike i programiranja. Time se podupire težnja o važnosti cjeloživotnog učenja koje bi trebalo biti nit vodilja svakom ozbiljnom inženjeru.

Jedno od mogućnosti nadogradnje ovog rada je izrada programske podrške koja bi omogućila asinkrono (*engl.* offline) programiranje robota. Time bi se u virtualnom okruženju mogao kreirati i simulirati program prije njegove implementacije u robota, što smanjuje vrijeme koje robot provodi van pogona (*engl.* downtime).

**LITERATURA**

- [1] <https://www.ancient-origins.net/history-famous-people/steam-powered-pigeon-002179>, posjećeno: 1.3.2022.
- [2] <https://www.ancient-origins.net/history-famous-people/steam-powered-pigeon-002179>, posjećeno 1.3.2022.
- [3] [https://hr.m.wikipedia.org/wiki/Datoteka:R.U.R.\\_by\\_Karel\\_%C4%8Capek\\_1939.jpg](https://hr.m.wikipedia.org/wiki/Datoteka:R.U.R._by_Karel_%C4%8Capek_1939.jpg), posjećeno 1.3.2022.
- [4] [https://www.researchgate.net/figure/a-A-picture-of-the-Unimate-first-prototype-picture-downloaded-from\\_fig3\\_340683333](https://www.researchgate.net/figure/a-A-picture-of-the-Unimate-first-prototype-picture-downloaded-from_fig3_340683333), posjećeno 1.3.2022.
- [5] <https://ifr.org/ifr-press-releases/news/record-2.7-million-robots-work-in-factories-around-the-globe>, posjećeno 1.3.2022
- [6] <https://www.therobotreport.com/top-5-countries-using-industrial-robots-2018/>, posjećeno 1.3.2022
- [7] <https://docplayer.fi/151425330-Teemu-kotila-pikarien-pakkaaminen-yhteistoimintarobotilla.html>, posjećeno 1.3.2022.
- [8] <https://docplayer.fi/151425330-Teemu-kotila-pikarien-pakkaaminen-yhteistoimintarobotilla.html>, posjećeno 1.3.2022.
- [9] <https://www.yaskawa.eu.com/products/robots/glossary>, posjećeno 1.3.2022.
- [10] <https://electricalworkbook.com/cartesian-robot/>, posjećeno 1.3.2022.
- [11] [https://www.researchgate.net/figure/SCARA-robot-manipulator-in-an-arbitrary-pose\\_fig4\\_251140027](https://www.researchgate.net/figure/SCARA-robot-manipulator-in-an-arbitrary-pose_fig4_251140027), posjećeno 1.3.2022.
- [12] <https://www.br-automation.com/en/products/machine-centric-robotics/robotics-portfolio/maximum-performance-with-delta-robots/>, posjećeno 1.3.2022.
- [13] <https://electricalworkbook.com/polar-robot/>, posjećeno 1.3.2022.
- [14] [https://www.researchgate.net/figure/Model-of-Articulated-robotic-arm\\_fig5\\_276062380](https://www.researchgate.net/figure/Model-of-Articulated-robotic-arm_fig5_276062380), posjećeno 1.3.2022.
- [15] [https://www.researchgate.net/figure/The-Configuration-of-a-Cylindrical-Arm-Robot\\_fig2\\_336702692](https://www.researchgate.net/figure/The-Configuration-of-a-Cylindrical-Arm-Robot_fig2_336702692), posjećeno 1.3.2022.
- [16] <https://www.universal-robots.com/plus/products/coval/suction-cups-vacuum-gripper-cvgl335xa50c1/>, posjećeno 10.3.2022.
- [17] <https://tameson.com/pneumatic-gripper.html>, posjećeno 10.3.2022.

- [18] <https://accu-techusa.com/shop/cobot/accessories-grippers-end-effectors/toyo-accessories-grippers-end-effectors/toyo-electric-gripper/>, posjećeno 10.3.2022.
- [19] <https://www.robots.com/articles/grippers-for-robots>, posjećeno 10.3.2022.

## **PRILOZI**

### I. Programski kod

## I. PROGRAMSKI KOD

```
import RobotDriverLeon as RDL #uvoz modula sa funkcijama na strani robota  
import sys
```

```
#Šifre za prikaz poruka:
```

```
ROBOTCOM_UNKNOWN = -1000  
ROBOTCOM_CONNECTION_PROBLEMS = -3  
ROBOTCOM_DISCONNECTED = -2  
ROBOTCOM_NOT_CONNECTED = -1  
ROBOTCOM_READY = 0  
ROBOTCOM_WORKING = 1  
ROBOTCOM_WAITING = 2
```

```
# Funkcija za povezivanje s robotom
```

```
def RobotConnect():  
    global ROBOT  
    global ROBOT_IP  
    global ROBOT_PORT  
    ROBOT.connect(ROBOT_IP) #pozivanje funkcije .connect robota  
    print("Robot is connected") #Ispis poruke  
    return ROBOT.connect(ROBOT_IP)
```

```
#Prikazivanje poruka u log-u ovisno o statusu robota
```

```
def UpdateStatus(set_status=None):  
    global STATUS  
    if set_status is not None:  
        STATUS = set_status  
    if STATUS == ROBOTCOM_CONNECTION_PROBLEMS:  
        print_message("Connection problems")  
    elif STATUS == ROBOTCOM_DISCONNECTED:  
        print_message("Disconnected")  
    elif STATUS == ROBOTCOM_NOT_CONNECTED:  
        print_message("Not connected")  
    elif STATUS == ROBOTCOM_READY:  
        print_message("Ready")  
    elif STATUS == ROBOTCOM_WORKING:  
        print_message("Working...")  
    elif STATUS == ROBOTCOM_WAITING:  
        print_message("Waiting...")  
    else:  
        print_message("Unknown status")
```

```
# Prekid veze s robotom
```

```
def RobotDisconnect():  
    global ROBOT  
    return ROBOT.disconnect()
```

```
# Funkcija koju poziva RoboDK prilikom pomicanja robota
```

```
def RunCommand(cmd_line):  
    global ROBOT_IP  
    global ROBOT_PORT  
    global ROBOT  
    global ROBOT_MOVING
```

```

# funkcija služi za pretvorbu string varijabli u float varijable
def line_2_values(line):
    values = []
    for word in line:
        try:
            number = float(word)
            values.append(number)
        except ValueError:
            pass
    return values

cmd_words = cmd_line.split(' ') #rezanje teksta na dijelove
cmd = cmd_words[0] #U cmd se sprema naredba (npr move linear, move joints...)
cmd_values = line_2_values(cmd_words[1:]) #u cmd_values se spremaju samo koordinate tipa float
n_cmd_values = len(cmd_values) #broj elemnata u cmd_values
n_cmd_words = len(cmd_words) #broj elemnata u cmd_words
print_message(str(cmd_words)) #ispis poruke (za test)
received = None

#Stvaranja rječnika s koordinatama
cord = {
    "X": cmd_values[0],
    "Y": -cmd_values[1], #negativna vrijednost jer je pravi robot napravljen u lijevokretnom koordinatnom sustavu
    "Z": cmd_values[2],
    "Rz": 0, }

if cmd_line == "":
    # Preskoči ako nije dana naredba
    return

elif n_cmd_values >= nDOFs_MIN and cmd_line.startswith("MOVJ"):
    # Obnova statusa
    UpdateStatus(ROBOTCOM_WORKING)
    # Pozivanje funkcije .move_linear na strani robota s parametrom coord
    ROBOT.move_linear(relative=False, **cord)
    ROBOT_MOVING = True

elif n_cmd_values >= nDOFs_MIN and cmd_line.startswith("MOVL"):
    # Obnova statusa
    UpdateStatus(ROBOTCOM_WORKING)
    ROBOT_MOVING = True
    print_message(str(cord)) #ispis poruke
    # Pozivanje funkcije .move_linear na strani robota s parametrom coord
    ROBOT.move_linear(relative=False, **cord)
    UpdateStatus(ROBOTCOM_READY)

elif cmd_line.startswith("t"):
    # Pozivanje custom funkcije za testiranje
    TestDriver()

```

```
elif cmd_line.startswith("CONNECT"):
    # Povezivanje s robotom
    ROBOT_IP = cmd_words[1] #iz liste cmd_words uzima prvi element koji predstavlja IP adresu ro-
    bota
    received = ROBOT.connect(ROBOT_IP)

else:
    #Za slučaj da je dana nepoznata naredba
    print("Unknown command: " + str(cmd_line))

if received is not None:
    UpdateStatus(ROBOTCOM_READY)
    ROBOT_MOVING = False

def RunDriver():
    for line in sys.stdin:
        RunCommand(line)

def RunMain():
    """Poziva se glavna funkcija"""
    # Ispis poruke o upravljačkom programu
    print_message("RoboDK Driver v2.0")

    # Poziva se RobotDisconnect ako se aborta program ili slično
    import atexit
    atexit.register(RobotDisconnect)

    print_message(DRIVER_VERSION)
    UpdateStatus()
    # Pokreće se driver iz STDIN
    RunDriver()

if __name__ == "__main__":
    """Call Main procedure"""
    # Poziva se glavna funkcija
    RunMain()
```