

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Antonio Ćuk**

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Fleksibilna primjena industrijskih robota temeljena na  
senzorima sile

Mentor:

Doc. dr. sc. Marko Švaco, mag. ing. mech.

Student:

Antonio Ćuk

Zagreb, 2024.

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

Zahvaljujem mentoru doc.dr.sc. Marku Švaci na pruženoj pomoći, savjetima i uputama pri izradi rada.

Posebno zahvaljujem kolegi Jurici Cvetiću i djelatnicima CRTA-e na pomoći i korisnim uputstvima tijekom izrade rada.

Zahvaljujem obitelji, prijateljima i djevojci na podršci tijekom studiranja.

Antonio Ćuk



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**  
Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne i diplomske ispite studija mehatronika i robotika



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 24 – 06 / 1	
Ur.broj: 15 – 24 –	

## ZAVRŠNI ZADATAK

Student: **Antonio Čuk**

JMBAG: **0035239603**

Naslov rada na hrvatskom jeziku: **Fleksibilna primjena industrijskih robota temeljena na senzorima sile**

Naslov rada na engleskom jeziku: **Flexible application of industrial robots based on force sensors**

Opis zadatka:

Broj industrijskih robota ima stabilan rast te sve veću primjenu u različitim industrijama, od tradicionalne primjene robota u automobilskoj industriji do sve šire primjene u industriji elektroničkih proizvoda. Industrijski roboti mogu se programirati na klasičan način gdje je svaki program orijentiran zadanoj putanji i radnim točkama koje su jasno definirane u radnom prostoru robota.

Industrijski roboti s naprednim senzorima kao što su sustavi strojnog vida ili senzori sile i momenata omogućavaju prilagodbu naučenih putanja i robotskih zadatak na temelju konfiguracije i stanja okoline. U ovom završnom radu potrebno je izraditi programski podršku za šest osnog industrijskog robota rotacijske strukture. Programsku podršku treba izraditi u proceduralnom programskom jeziku KAREL koristeći povratne informacije s integriranog senzora sile na priрубnici robota.

U radu je potrebno:

- upoznati se s proceduralnim programskim jezikom KAREL
- prema mogućnostima istražiti napredne primjene programskog jezika KAREL
- upoznati se s osnovnim funkcijama zadanog senzora sile
- izraditi programsku podršku koja će omogućiti relativnu transformaciju naučenih robotskih putanja, točaka i primjene senzora sile, prema identificiranom položaju kalibracijskih značajki pomoću senzora sile.

Funkcionalnosti iz završnog rada potrebno je demonstrirati na odabranom industrijskom robotu u Laboratoriju za autonomne sustave, Regionalnog centra izvrsnosti za robotske tehnologije.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

24. 4. 2024.

Zadatak zadao:

doc. dr. sc. Marko Švaco

Datum predaje rada:

2. rok (izvanredni): 11. 7. 2024.  
3. rok: 19. i 20. 9. 2024.

Predviđeni datumi obrane:

2. rok (izvanredni): 15. 7. 2024.  
3. rok: 23. 9. – 27. 9. 2024.

Predsjednik Povjerenstva:

izv. prof. dr. sc. Petar Čurković

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
POPIS TABLICA.....	III
POPIS KRATICA .....	IV
SAŽETAK.....	V
SUMMARY .....	VI
1. UVOD.....	1
2. OPĆENITO O INDUSTRIJSKOJ ROBOTICI .....	2
2.1. Industrijski roboti.....	3
2.1.1. Konfiguracije industrijskih robota .....	4
2.2. Kolaborativni roboti .....	6
2.2.1. FANUC CR-15iA .....	8
3. SENZORI SILE I MOMENTA .....	10
3.1. Senzor sile FS-15iA .....	12
4. LABORATORIJSKA VALIDACIJA .....	13
4.1. FUNKCIJE KONTROLE SILE.....	13
4.1.1. Constant Push.....	14
4.1.2. Shaft Insert .....	15
4.1.3. Square Insert .....	17
4.1.4. Face Match.....	18
4.1.5. Hole Search .....	19
4.1.6. Phase Search .....	20
4.2. SIMULACIJSKI SOFTWARE <i>ROBOGUIDE</i> .....	21
4.2.1. Dual Check Safety .....	22
4.3. O PROGRAMSKOM JEZIKU <i>KAREL</i> .....	23
4.3.1. Tipovi podataka u KAREL-u.....	23
4.3.2. Koordinatni sustavi robota .....	24
4.3.3. Osnovne naredbe za razumijevanje koda.....	25
4.3.4. Opis postava.....	29
4.3.5. Upravljački programi .....	31
4.3.6. Rezultati rada .....	39
5. ZAKLJUČAK.....	40
LITERATURA.....	41
PRILOZI.....	42

## POPIS SLIKA

Slika 2.1.	Industrijski robot <i>Unimate</i> [1] .....	2
Slika 2.2.	Kartezijeva struktura robota[2] .....	4
Slika 2.3.	Cilindrična struktura robota[2] .....	4
Slika 2.4.	Kvazicilindrična struktura robota[2] .....	4
Slika 2.5.	Sferna struktura robota[2] .....	5
Slika 2.6.	Rotacijska struktura robota[2] .....	5
Slika 2.7.	SCARA struktura robota[2] .....	5
Slika 2.8.	Delta robot, CRTA .....	6
Slika 2.9.	Kolaborativni robot, CRTA .....	6
Slika 2.10.	Vrste interakcija između robota i čovjeka[4] .....	7
Slika 2.11.	FANUC CR-15iA, CRTA .....	8
Slika 2.12.	Specifikacije robota CR-15iA[5] .....	9
Slika 3.1.	Veza senzora s ostatkom sustava[8] .....	11
Slika 3.2.	Senzor sile i momenta FS-15iA[9] .....	12
Slika 4.1.	Primjer točke prilaska za funkciju <i>Constant Push</i> [13] .....	14
Slika 4.2.	Primjer točke prilaska za pozivanje funkcije <i>Shaft Insert</i> [13] .....	16
Slika 4.3.	Primjer greške pri pozicioniranju robota[12] .....	16
Slika 4.4.	Korisničko sučelje ROBOGUIDE-a .....	21
Slika 4.5.	DCS prikaz na privjesku za učenje .....	22
Slika 4.6.	Korisnički sustavi robota[14] .....	25
Slika 4.7.	Prikaz rada FRAME funkcije[14] .....	28
Slika 4.8.	Tlocrt stola na kotačima .....	29
Slika 4.9.	Fiksni stol .....	30
Slika 4.10.	Konstrukcija za izmjenu alata .....	30
Slika 4.11.	Proces izvršavanja programa <i>Hole Search</i> .....	34
Slika 4.12.	Proces izvršavanja programa <i>Square Insert</i> .....	35
Slika 4.13.	Proces izvršavanja funkcije <i>Phase Search</i> .....	36
Slika 4.14.	Proces izvršavanja funkcije montaže ekstrudera .....	38
Slika 4.15.	Sile tijekom vođenja robota .....	39

---

**POPIS TABLICA**

Tablica 4.1.	Glavni parametri funkcije <i>Constant Push</i> .....	14
Tablica 4.2.	Glavni parametri funkcije <i>Shaft Insert</i> .....	15
Tablica 4.3.	Glavni parametri funkcije <i>Squire Insert</i> .....	17
Tablica 4.4.	Glavni parametri funkcije <i>Face Match</i> .....	18
Tablica 4.5.	Glavni parametri funkcije <i>Hole Search</i> .....	19
Tablica 4.6.	Glavni parametri funkcije <i>Phase Search</i> .....	20
Tablica 4.7.	Tipovi podataka u <i>KAREL-u</i> .....	23

---

**POPIS KRATICA**

<b>Oznaka</b>	<b>Opis</b>
TP	engl. <i>Teach pendant</i>
CRTA	Regionalni centar izvrsnosti za robotske tehnologije
CNC	engl. <i>Computer Numerical Control</i>



---

**SAŽETAK**

U ovom radu opisane su i demonstrirane mogućnosti primjene industrijskog robota kolaborativne prirode FANUC CR-15iA i ugrađenog senzora sile FS-15iA. Kolaborativni roboti rade u okruženju s ljudima te je bitno da mogu detektirati kolizije s okolinom kako bi pružili što veću sigurnost okolini. Za detekciju kolizija koristi se senzor sile montiran na vrhu prirubnice robota, međutim senzor sile osim detektiranja kolizije omogućuje robotu izvršavanje raznih zadataka koji su bitni u industriji. Neke glavne primjene robota s ugrađenim senzorima sile su poliranje, brušenje i montaža. Svaka od ovih zadaća zahtjeva praćenje vrijednosti sile te njeno održavanje na zadanoj vrijednosti. U praktičnom dijelu rada napisan je programski kod za vođenje robota, traženje referentnih točaka te izradu novog korisničkog koordinatnog sustava na kojem se mogu demonstrirati neke od funkcija ugrađenog senzora sile. Proces je napisan kodom u programskom jeziku *KAREL*, a potprogrami su napisani pomoću programske podrške na privjesku za učenje (engl. *teach pendant*)

**Ključne riječi:** industrijski robot, kolaborativni robot, CR-15iA, senzor sile, *KAREL*, privjesak za učenje

---

## SUMMARY

This bachelor thesis describes and demonstrates the potential applications of the FANUC CR-15iA industrial robot, which is of a collaborative nature, and its built-in FS-15iA force sensor. Collaborative robots work in environments with humans, and it is important that they can detect collisions with their surroundings to ensure maximum safety from the environment. The robot uses a force sensor mounted at the top of its end-effector to detect collisions; however, in addition to detecting collisions, the force sensor enables the robot to perform various tasks essential in industry. Some main applications of robots with built-in force sensors include polishing, grinding and assembly. Each of these tasks requires monitoring and maintaining a specific force value. In the practical part of the thesis, code was programmed to guide the robot, search for reference points, and create a new user coordinate system where some functions of the built-in force sensor can be demonstrated. The process was programmed using code written in the *KAREL* programming language, while subprograms were written using software support on the teach pendant.

**Key words:** industrial robot, collaborative robot, CR-15iA, force sensor, *KAREL*, teach pendant

## 1. UVOD

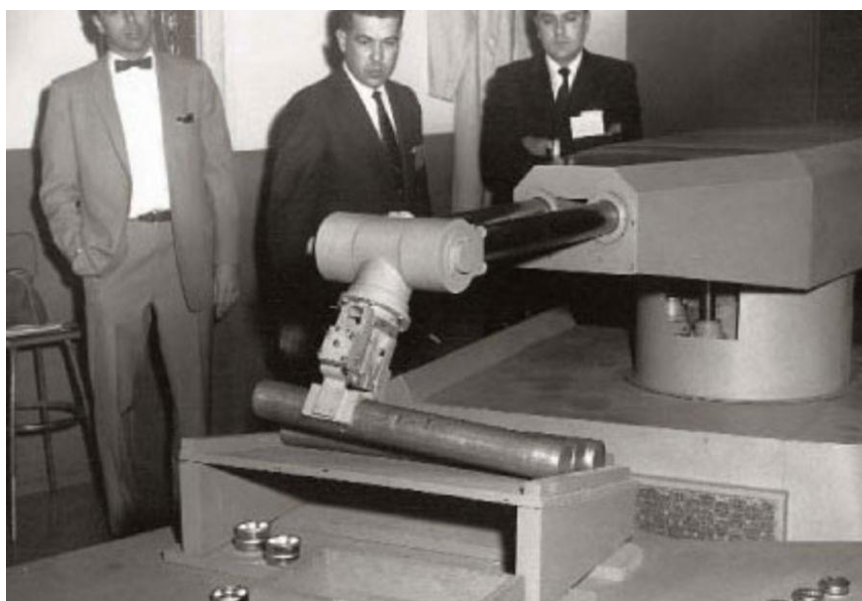
Robotika je u današnje vrijeme jedna od najzastupljenijih grana moderne industrije i njezini aspekti nalaze se u gotovo svakom većem tehnološkom pogonu. Sa svakidašnjem napretkom, roboti postaju sve više opremljeni za izravan rad s ljudima u njihovim okruženjima. Jedan od ključnih senzora koji im omogućuje takav rad je senzor sile i momenta.

Industrijski roboti često koriste unaprijed definirane korisničke koordinatne sustave koji se postavljaju s obzirom na neku određenu komponentu ili radno područje. Međutim, ako se ta komponenta pomakne u prostoru, robot ne može obavljati zadatke koji su bili definirani pomoću koordinatnog sustava te komponente. Cilj ovog završnog rada je razviti program koji će pomoću ljudske interakcije i senzora sile omogućiti robotu samostalno stvaranje koordinatnog sustav koristeći određene prostorne značajke. U radu su također demonstrirane ugrađene funkcije kontrole sile s kojima robot raspolaže te su opisani njihovi najbitniji parametri.

Svi programski kodovi i backup robota napravljeni su unutar sklopa software-a *ROBOGUIDE verzija 9.4.* sa *R-30iB Plus* kontrolerom te se nalaze u priloženim *GitHub* repozitorijima.

## 2. OPĆENITO O INDUSTRIJSKOJ ROBOTICI

Robotika je multidisciplinarno područje koje obuhvaća više različitih znanstvenih disciplina i grana od kojih su najistaknutije strojarstvo, elektrotehnika, programiranje te automatika. Krajnji je cilj industrijske robotike stvoriti strojeve koji mogu automatski ili poluautomatski obavljati zadatke, najčešće u okruženjima koja su neprikladna za ljude ili u zadacima koji zahtijevaju veliku preciznost ili su repetitivni. Grana robotike počela se razvijati sredinom 20. st. kada je tvrtka *Unimation* 1961. ugradila svog prvog industrijskog robota *Unimate*-a u tvornici *General Motors*-a. *Unimate* je bio hidraulički manipulator koji bi rukovao dijelovima za aute te obavljao zadatke zavarivanja slijedeći *step-by-step* komande pohranjene na magnetnom bubnju[1].



**Slika 2.1.** Industrijski robot *Unimate*[1]

Nakon dolaska *Unimate*-a ostale tvornice su zbog povećane produktivnosti, preciznosti i ubrzanja procesa ubrzo prepoznale prednosti upotrebe robota u proizvodnim linijama. Većina robota prve generacije obavljala je jednostavna uputstva i pokrete koji nisu zahtijevali kompleksnu strukturu robota i zahtjevan kod. Rastućim napretkom ljudske rase napredovali su i roboti, kod kojih se počela koristiti tehnologija raznih senzora i mikroprocesora. Ovo je robotima dalo dodatni stupanj samostalnog rada i kompleksnosti zadataka koji se mogu obavljati. Na početku 21. st. počela se razvijati umjetna inteligencija, strojno učenje te znanost vizijskih sustava što je omogućilo robotima veću razinu autonomije i prilagodbu kompleksnim okolinama i zadacima.

Iako je robotika iznimno kompleksno područje, može se podijeliti u nekoliko osnovnih grana; ovisno o primjeni i svrsi robota.

- **Industrijska robotika:** Grana robotike čiji je glavni fokus na robote koji obavljaju repetitivne zadatke u proizvodnji i industriji
- **Uslužna robotika:** Obuhvaća robote koji obavljaju zadatke u domaćinstvima, medicini i drugim područjima gdje obitavaju ljudi
- **Mobilna robotika:** Roboti koji se kreću kroz poznato ili nepoznato okruženje koristeći napredne senzore i kompleksne algoritme
- **Humanoidna robotika:** Roboti dizajnirani da oponašaju ljudske pokrete i interakcije
- **Medicinska robotika:** Fokusira se na primjenu robota u medicini i zdravstvu, gdje omogućuju operacije s većom preciznošću nego što je moguće postići ljudskom rukom

## 2.1. Industrijski roboti

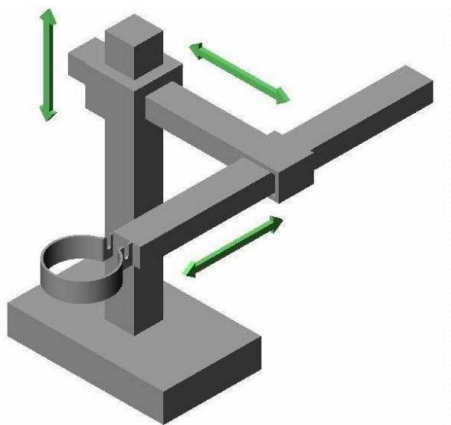
Industrijski roboti su višenamjenski automatizirani strojevi dizajnirani za obavljanje repetitivnih i preciznih zadataka u industriji poput zavarivanja, montaže, paletizacije i manipulacije objektima. Glavna je prednost industrijskih robota ekstremno visoka produktivnost i preciznost u zadacima gdje ljudska radna snaga može biti podložna umoru i greškama. Najčešća je primjena industrijskih robota u automobilskim, elektroničkim i ostalim proizvodnim industrijama gdje je ubrzanje procesa ključno. Industrijski roboti sastoje se od četiri glavna sustava [2]:

- Mehanički sustav – fizička struktura robota. Omogućuje robotu da se kreće i manipulira objektima
- Energetski sustav – opskrbljuje robota energijom potrebnom za ispravan rad. Omogućuje pogon motora i elektroničkih komponenti
- Mjerni sustav – sustav zaslužan za prikupljanje informacija o okolini robota i unutarnjih veličina robota
- Upravljački sustav – sustav koji koristi upravljajuće algoritme i softver za koordinaciju ostalih komponenti kako bi se postigao željeni cilj

### 2.1.1. Konfiguracije industrijskih robota

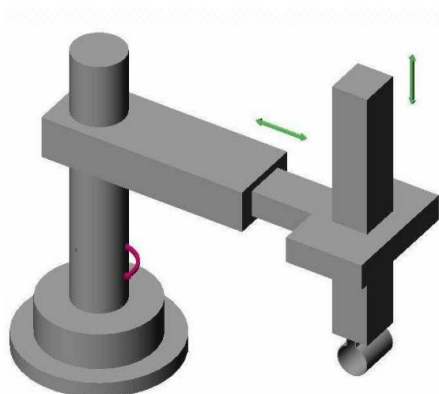
Roboti se najčešće dijele s obzirom na njihovu kinematsku strukturu. Najčešće konfiguracije su sljedeće[2]:

- 1) Kartezijska (engl. *cartesian*)



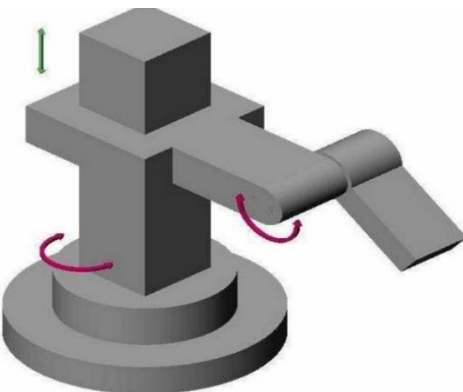
Slika 2.2. Kartezijska struktura robota[2]

- 2) Cilindrična (engl. *cylindrical*)

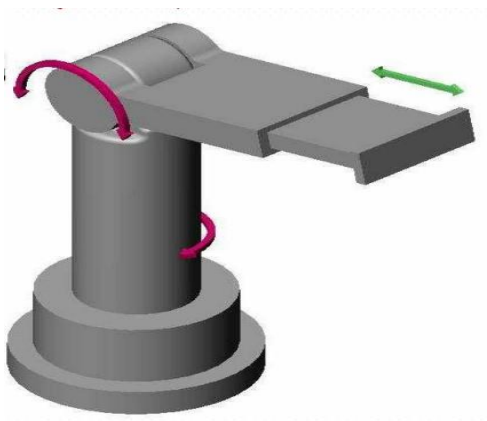
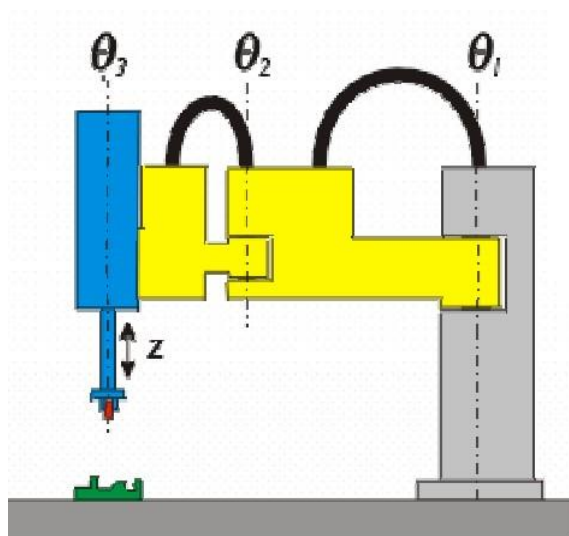


Slika 2.3. Cilindrična struktura robota[2]

- 3) Kvazicilindrična (engl. *quasi-cylindrical*)



Slika 2.4. Kvazicilindrična struktura robota[2]

4) Sferna (engl. *spherical*)**Slika 2.5.** Sferna struktura robota[2]5) Rotacijska (engl. *articulated*)**Slika 2.6.** Rotacijska struktura robota[2]6) SCARA (engl. *Selective Compliance Articulated Robot Arm*)**Slika 2.7.** SCARA struktura robota[2]

7) Delta robot (engl. *Delta robot*)

Slika 2.8. Delta robot, CRTA

**2.2. Kolaborativni roboti**

U praksi je većina industrijskih robota okružena ogradama, svjetlosnim zavjesama i ostalim sigurnosnim mjerama koje sprječavaju interakciju između čovjeka i robota. Međutim, kolaborativni roboti ili koboti su opremljeni s više senzora i mjerama sigurnosti te time omogućuju suradnju i međudjelovanje robota s ljudima. Oni s pomoću senzora detektiraju moguće kolizije te nakon detekcije u potpunosti zaustavljaju svoje kretanje. Omogućavanjem interakcije čovjek – robot izvršavanje procesa je brže i efikasnije. Svi sigurnosni zahtjevi su propisani i moraju poštivati određene norme (*ISO 10218-1*, *ISO 10218-2*, *ISO/TS 15066*). Osim samim mjernim sustavom robota, sigurnost se povećava i samom izvedbom konstrukcije gdje se smanjuje broj točaka kod kojih bi moglo doći do uklještenja dijelova tijela. Također se koriste mekši i lakši materijali za vanjske površine robota te neke dodatne meke obloge kako bi se smanjila sila koja se prenosi na čovjeka ili okolinu[3].

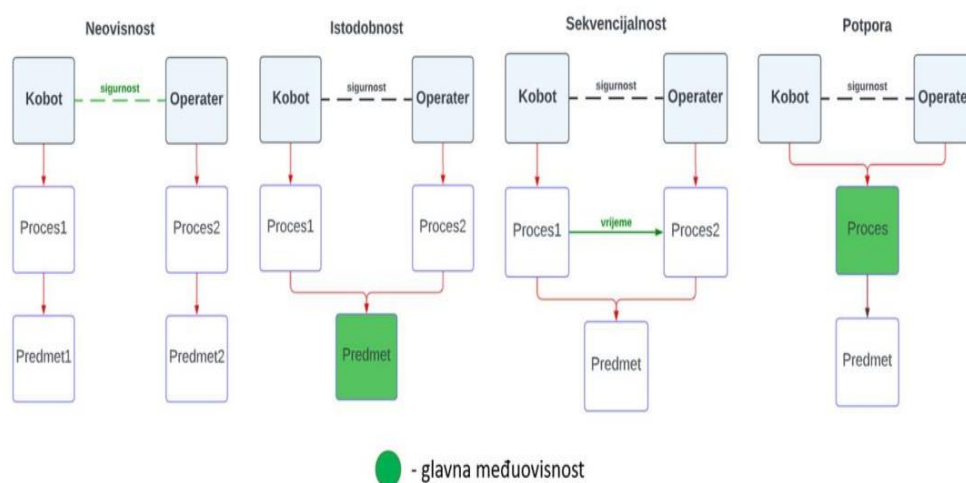


Slika 2.9. Kolaborativni robot, CRTA



Kako bi se bolje razumjela interakcija između čovjeka i robota, definirane su neke kategorije kolaborativnih slučajeva. Ove kategorije razvrstane su prema razini autonomije robota u odnosu na čovjeka. Ovakva kategorizacije olakšava odabir robota za određeni zadatak unutar tvrtke. Postoje četiri različite razine interakcije čovjeka i robota[4].

- 1) Neovisna interakcija – čovjek i robot rade s različitim predmetima u odvojenim procesima, međutim dijele isti radni prostor bez sigurnosnih mjera. Kobot koristi vanjske senzore i programske alate kako bi percipirao okolinu
- 2) Istodobna interakcija – čovjek i robot istodobno rade s istim predmetom u isto vrijeme, ali su im zadaci različiti. Potrebno je osigurati dovoljno prostora kako bi se omogućile istovremene akcije te kako bi se smanjilo vrijeme rada
- 3) Sekvencijalna interakcija – čovjek i robot naizmjenično obavljaju zadatke na istom predmetu. Ovdje postoji međuovisnost i vremenska ograničenja. Kobot u ovom slučaju preuzima zamorne i repetitivne zadatke kako bi se olakšao ljudski rad
- 4) Potporna interakcija – čovjek i kobot zajedno izvršavaju isti proces na istom predmetu. Proces se može uspješno dovršiti samo stalnom suradnjom čovjeka i robota



**Slika 2.10.** Vrste interakcija između robota i čovjeka[4]

### 2.2.1. FANUC CR-15iA

FANUC je japanska kompanija osnovana 1956. godine koja je specijalizirana za automatizaciju i robotiku. Njeno ime nosi akronim za *Fuji Automatic Numerical Control*. FANUC je pionir u razvoju CNC sustava i industrijskih robota, s globalnim prisustvom u brojnim industrijama. FANUC ima široki raspon industrijskih robota koji pokrivaju različite primjene poput zavarivanja, obrade materijala, montaže i mnoge druge. Glavne prednosti njihovih robota su velika preciznost, dug vijek trajanja, visoka pouzdanost te uz sve to zahtijevaju minimalne zahtjeve za održavanjem. Jedan od robota koji proizvode nalazi se u CRTA-i na kojem je izvršen praktični dio rada. FANUC-ov CR-15iA (Slika 2.11.) je kolaborativni robot visoke točnosti i ponovljivosti iz serije CR (*Collaborative Robot*), dizajniran za rad u ljudskoj okolini bez potrebe za sigurnosnim pregradama i barijerama.



Slika 2.11. FANUC CR-15iA, CRTA

Ovo je robot revolutne strukture sa 6 stupnjeva slobode gibanja. Ima integriran osjetljiv senzor koji se nalazi u bazi robota s pomoću kojeg spoznaje vlastitu okolinu te pruža sigurnosni rad. Neke osnovne specifikacije robota su sljedeće[5]:

Model		CR-15iA
Type		6axes (J1,J2,J3,J4,J5,J6)
Reach		1441 mm
Installation		Floor, Upside-down, Angle mount
Motion range (Note 1)	J1 axis	340° / 370° 5.93 rad / 6.46 rad
	J2 axis	180° 3.14 rad
	J3 axis	312° 5.45 rad
	J4 axis	380° 6.63 rad
	J5 axis	280° 4.89 rad
	J6 axis	900° 15.71 rad
Max load capacity at wrist		15 kg
Maximum speed (Note 2)		800 mm/s (Note 3) (Max 1500 mm/s (Note 4))
Allowable load moment at wrist	J4 axis	26.0 N·m
	J5 axis	26.0 N·m
	J6 axis	11.0 N·m
Allowable load inertia at wrist	J4 axis	0.90 kg·m <sup>2</sup>
	J5 axis	0.90 kg·m <sup>2</sup>
	J6 axis	0.30 kg·m <sup>2</sup>
Repeatability(Note 5)		± 0.02 mm
Mass (Note 6)		255 kg
Installation environment		Ambient temperature : 0 to 45°C Ambient humidity : Normally 75%RH or less (No dew nor frost allowed), Short time (within one month) Max.95%RH

Slika 2.12. Specifikacije robota CR-15iA[5]

### 3. SENZORI SILE I MOMENTA

U današnje vrijeme većina industrijskih robota izvršava gotove programe ponavljajući unaprijed definirane sekvence pokreta. Međutim, takvi roboti nisu adekvatni za zadatke gdje se veličine i položaji predmeta mogu mijenjati. Neki od zadataka zahtijevaju programiranje i ponavljanje pokreta s velikom preciznošću, koja se ne može svaki put tako jednostavno postići. Takvi zadaci mogu se izvesti korištenjem senzorskih informacija za prilagodbu pokreta robota.

Vanjski senzori korišteni u robotici dijele se u dvije velike skupine: kontakti i beskontaktni.

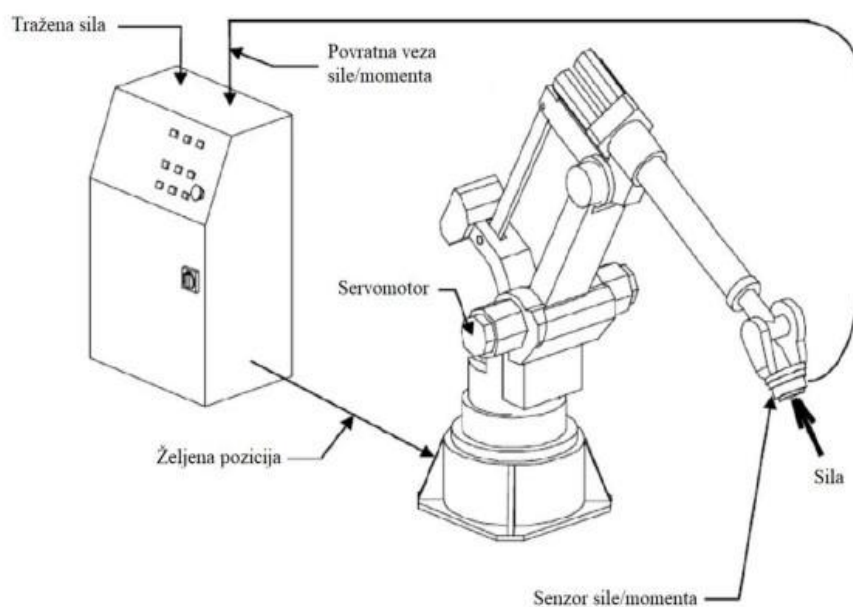
Najbolji je primjeri beskontaktnog senzora vizijski sustav robota s kojim FANUC CR-15iA također raspolaže, ali se ne koristi u ovom radu. Senzor sile koji pruža informacije o silama i momentima detektiranim na priрубnici robota; najrašireniji je i najkorisniji kontakti senzor.

Korištenjem informacija o sili, senzor robotu pruža optimalna rješenja za raznovrsne zadatke. Najjednostavnija primjena senzora sile kod robota je vaganje predmeta ili zaustavljanje kretanja robota kada se detektira kolizija. Informacije sa senzora također se mogu koristiti za rukovanje radnim predmetima koji su nasumično orijentirani. Robot također može pratiti prethodno nepoznati i neodređeni oblik tijela. Ovo se svojstvo koristi pri zavarivanju, brušenju i drugim proizvodnim zadacima koji zahtijevaju kretanje duž površine, konture ili spoja više radnih komada. Kod svih senzora sile, sile i momenti mjere se putem deformacija ili pomaka u elastičnom elementu senzora, a za pretvorbu tih veličina u izlazni signal koriste se različiti načini poput[6]:

- potencijometarskih pretvarača pomaka – promjene pomaka mijenjaju vrijednost otpora
- induktivnih pretvarača pomaka – mjerenjem promjene međusobne induktivnosti dviju zavojnica koje dijele pomičnu željeznu jezgru može se detektirati pomak
- kapacitivnih pretvarača pomaka – mjerenjem promjene kapaciteta uzrokovane pomicanjem jedne ploče kondenzatora u odnosu na drugu može se detektirati pomak
- piezoelektričnih pretvarača – pretvaraju deformaciju u električni izlaz, međutim ne može se mjeriti DC komponenta sile te zbog toga moraju biti opremljeni dodatnim pojačalima za mjerenje signala niske frekvencije
- elektrovodljivim polimerima – materijali koji mijenjaju električni otpor pod opterećenjem

- tenzometarskim pretvaračima – ovo su najrašireniji senzori za mjerenje sile. U ovim pretvaračima deformacija uzrokuje promjenu električnog otpora tenzometra koji su postavljeni na elastični element. Ovi senzori imaju visoku osjetljivost i preciznost mjerenja, ali su jako osjetljivi na temperaturu.

Mjerni sustav robota uobičajeno uključuje senzor sile i momenta zajedno s mehaničkom i upravljačkom jedinicom robota koji djeluju kao zasebni sustav. Senzor sile i momenta postavljen je kao 6-osni mjerni pretvarač na jedan od zglobova robota; na koji se pričvršćuje alat robota. Da bi se ostvarila određena sila, robot pritišće alat na površinu predmeta dok sklop unutar senzora pretvara očitavanje vrijednosti sile i momenta u električne signale koji se prenose do upravljačke jedinice. Ovi signali predstavljaju iznose sile duž X, Y i Z osi te njihove pripadajuće momente. Upravljačka jedinica robota prima te signale, vrši potrebne proračune, uspoređuje ih s postavljenim vrijednostima sile te određuje eventualna odstupanja[8].



**Slika 3.1.** Veza senzora s ostatkom sustava[8]

Ove karakteristike senzora sile i momenta ključne su kod raznovrsnih industrijskih zadataka kao što su: bušenje, brušenje, poliranje, zatezanje vijaka, kontrole kvalitete, primjene ljepila ili silikona, detekcija kolizije i mnogi drugi.

### 3.1. Senzor sile FS-15iA

U ovom radu koristio se FANUC-ov senzor sile i momenta FS-15iA. Ovaj 6-osni senzor koristi silikonske tenzometarske trake koje šalju informacije o sili elektroničkom sklopu. Sklop tada pretvara te informacije u 6 kanalne signale napona; svaki kanal za pojedinu os. U slici dolje navedene su neke karakteristike navedenog senzora.

	DOF	Dim Ø [mm]	Weight [kg]	Rated load		Static overload tolerance		Resolution		Accuracy [%]
				Fx, Fy, Fz [N / kgf]	Mx, My, Mz [Nm / kgfcm]	Fx, Fy, Fz [N / kgf]	Mx, My, Mz [Nm / kgfcm]	Fx, Fy, Fz [N / kgf]	Mx, My, Mz [Nm / kgfcm]	
<b>FS-15iA</b>										
	6 axis	94 x 43	0.57	147 / 15	11.8 / 120	1570 / 160	125 / 1280	0.39 / 0.04	0.016 / 0.16	less than 2

**Slika 3.2.** Senzor sile i momenta FS-15iA[9]

## 4. LABORATORIJSKA VALIDACIJA

U eksperimentalnom dijelu ovog rada opisuju se osnovne funkcije kontrole sile s kojima možemo raspolagati pomoću senzora FS-15iA. Te funkcije su također korištene za realizaciju naprednijeg koda pisanog u programu KAREL koji korisniku omogućuje vođenje robota rukom, traženje utora na stolu koji predstavljaju referentne točke korisničkog koordinatnog sustava te spremanje točaka u pozicijske registre. Na kraju izrade novog koordinatnog sustava, moguće je pozvati potprograme napisane pomoću programske podrške na privjesku za učenje u kojima se demonstriraju funkcije kontrole sile.

### 4.1. FUNKCIJE KONTROLE SILE

FANUC je razvio integrirane funkcije koje omogućuju izvođenje naprednih operacija poput montaže proizvoda, poliranja i sklapanja zupčanika. Svaka od ovih funkcija uključuje osnovne i napredne parametre koje treba prilagoditi specifičnom procesu kako bi se poboljšala učinkovitost rada. Za detaljnije informacije o postavljanju parametara, preporučena je literatura [12]. Naredbe vezane uz funkcije kontrola sile su sljedeće:

- *FORCE CONTROL* – naredbe vezane uz izvršavanje 12 različitih funkcija kontrole sile, ovisne o procesu i predmetu rada
- *SENSOR DIAGNOSIS* – naredba koja se koristi za nadzor i dijagnozu različitih senzora ugrađeni na robota
- *GET DIAG DATA* – naredba koja daje povratnu informaciju o stanju senzora sile
- *AUTO TUNING ON/OFF* – naredba koja automatski mijenja pojačanje o kojoj ovisi odziv funkcije kontrole sile
- *TRQ ERROR ON/OFF* – naredba koja omogućava sustavu da nadzire trenutni moment i prepozna abnormalnosti; time sprječavajući oštećenja i kvarove tijekom rada
- *END COND ON/OFF* – naredba koja omogućuje automatsko određivanje dubine na kojoj se nalazi umetnuti predmet rada

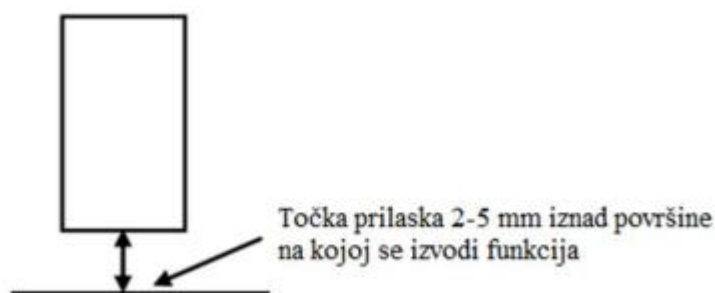
#### 4.1.1. *Constant Push*

Funkcija kontrole sile *Constant Push* koristi se u primjenama gdje je potrebno da robot primjenjuje konstantnu silu na površinu tijekom izvršavanja zadatka. Najbolji primjeri za to su neke montaže predmeta ili konfirmacija kontakta s površinom. Sila koju robot stvara nije kontrolirana pozicijskim informacijama, već pomoću povratnih informacija sa senzora sile. Senzori mjere koliko robot stvara silu, a upravljačka jedinica pomoću tih informacija prilagođava kretanje motora robota.

**Tablica 4.1.** Glavni parametri funkcije *Constant Push*

Stavka	Opis
Korisnički koordinatni sustav (engl. <i>User Frame</i> )	Odabrani korisnički koordinatni sustav za izvršavanje funkcije kontrole sile.
Koordinatni sustav alata (engl. <i>Tool Frame No.</i> )	Odabrani koordinatni sustav alata za izvršavanje funkcije kontrole sile.
Smjer primjene sile (engl. <i>Pushing direction</i> )	Odabir smjera primjene sile s obzirom na postavljeni korisnički koordinatni sustav.
Granična sila ostvarenja kontakta (engl. <i>Approach Dist. Limit</i> )	Najmanja vrijednost sile kod koje robot prepoznaje da je uspostavio kontakt.
Primijenjena sila $a$ (engl. <i>Pushing force</i> )	Sila koja se primjenjuje nakon uspješnog uspostavljenog kontakta s površinom.
Vrijeme primjene sile (engl. <i>Pushing time</i> )	Trajanje primjene sile nakon uspostave kontakta s površinom.

Bitno je napomenuti da je pri izvođenju ove funkcije ključno dovesti robota u položaj nekoliko milimetara iznad površine na koju želimo pozvati funkciju *Constant Push*.



**Slika 4.1.** Primjer točke prilaska za funkciju *Constant Push*[13]



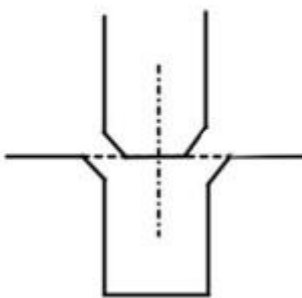
#### 4.1.2. *Shaft Insert*

Sljedeća funkcija koja je dostupna i ključna za realizaciju ovog rada je *Shaft Insert*. Ova funkcija koristi se za precizno umetanje osovine ili cilindričnih predmeta u odgovarajuće rupe ili uture. Primjene takvog zadatka možemo naći u autoindustriji pri montaži osovine ili vratila. Robot s pomoću ove funkcije može kontrolirano i s točno definiranom silom umetati osovine uz prilagodbu putanje i pozicije s obzirom na detektirani otpor tijekom izvršavanja zadatka. Tijekom procesa robot pokušava umetnuti osovinu u utor te koristi povratne informacije sa senzora kako bi prepoznao trenutni otpor. Detekcijom otpora i prevelike sile, robot si automatski može prilagoditi poziciju kako bi ispravno umetnuo dio i spriječio oštećenje dijelova.

**Tablica 4.2.** Glavni parametri funkcije *Shaft Insert*

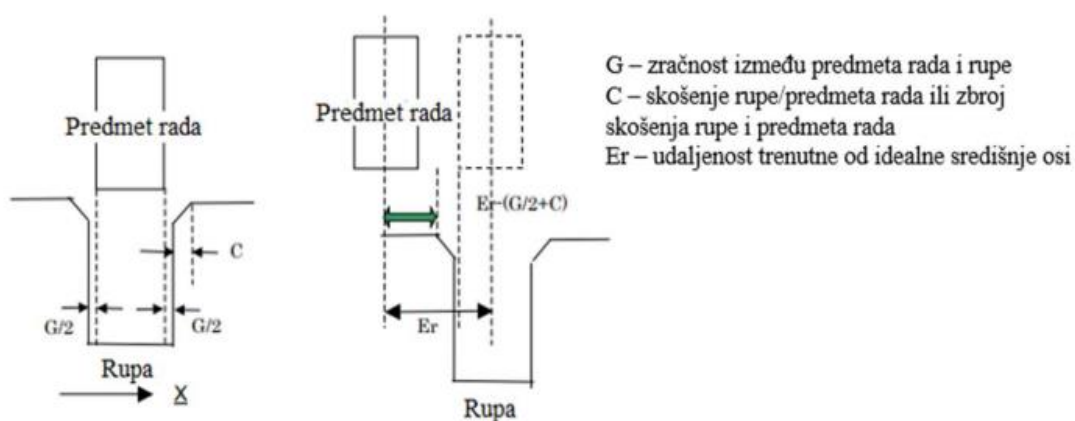
Stavka	Opis
Korisnički koordinatni sustav (engl. <i>User Frame</i> )	Odabrani korisnički koordinatni sustav za izvršavanje funkcije kontrole sile.
Koordinatni sustav alata (engl. <i>Tool Frame No.</i> )	Odabrani koordinatni sustav alata za izvršavanje funkcije kontrole sile.
Smjer umetanja (engl. <i>Insert Direction</i> )	Odabir smjera umetanja s obzirom na postavljeni korisnički koordinatni sustav.
Dubina umetanja (engl. <i>Insert Depth</i> )	Numerička vrijednost dubine umetanja mjerene od točke prilaska do krajnje točke.
Brzina umetanja (engl. <i>Insert Velocity</i> )	Numerička vrijednost koja određuje brzinu kretanja predmeta rada nakon ostvarenja kontakta.
Sila umetanja (engl. <i>Insert Force</i> )	Numerička vrijednost koja označava vrijednost sile pri izvršavanju procesa umetanja predmeta rada.
Ograničenje vremena umetanja (engl. <i>Insert Time Limit</i> )	Numerička vrijednost koja označava period unutar kojeg se mora izvršiti funkcija umetanja. Ako se funkcija ne izvrši u zadanom periodu, javlja se alarm.

Kod ove funkcije treba paziti da se robotska ruka ili predmet kojom rukuje dovede u točku, tako da se donja površina predmeta ili ruke poklapa s površinom gdje želimo pozvati funkciju.



**Slika 4.2.** Primjer točke prilaska za pozivanje funkcije *Shaft Insert*[13]

Također je važno napomenuti da ova funkcija može tolerirati određenu grešku u smjeru određene osi. Ako greška pozicioniranja u smjeru osi prelazi vrijednost  $(C+G/2)$ , funkcija *Shaft Insert* ne može se izvršiti[12].



**Slika 4.3.** Primjer greške pri pozicioniranju robota[12]

### 4.1.3. *Square Insert*

Ova se funkcija koristi za umetanje četverokutne prizme u pravokutni utor. Iznimno je slična funkciji umetanja osovine s jedinom razlikom u korištenom predmetu rada. Međutim, za razliku od umetanja cilindričnih dijelova kao u prošlom primjeru, kvadratni dijelovi zahtijevaju ne samo pozicioniranje u središte već i pravilno poravnavanje kuta.

Funkcija koristi informacije sa senzora kako bi osigurala da kvadratni dio prvo bude poravnat s otvorom prije nego što započne proces umetanja.

**Tablica 4.3.** Glavni parametri funkcije *Square Insert*

Stavka	Opis
Korisnički koordinatni sustav (engl. <i>User Frame</i> )	Odabrani korisnički koordinatni sustav za izvršavanje funkcije kontrole sile.
Koordinatni sustav alata (engl. <i>Tool Frame No.</i> )	Odabrani koordinatni sustav alata za izvršavanje funkcije kontrole sile.
Smjer umetanja (engl. <i>Insert Direction</i> )	Odabir smjera umetanja s obzirom na postavljeni korisnički koordinatni sustav.
Dubina umetanja (engl. <i>Insert Depth</i> )	Numerička vrijednost dubine umetanja mjerene od točke prilaska do krajnje točke.
Brzina umetanja (engl. <i>Insert Velocity</i> )	Numerička vrijednost koja određuje brzinu kretanja predmeta rada nakon ostvarenja kontakta.
Sila umetanja (engl. <i>Insert Force</i> )	Numerička vrijednost koja označava vrijednost sile pri izvršavanju procesa umetanja predmeta rada.
Ograničenje vremena umetanja (engl. <i>Insert Time Limit</i> )	Numerička vrijednost koja označava period unutar kojeg se mora izvršiti funkcija umetanja. Ako se funkcija ne izvrši u zadanom periodu, javlja se alarm.
Provjera orijentacije (engl. <i>Check Orientation Change</i> )	Mogućnost odabira provjere orijentacije tijekom izvođenja funkcije usklađivanja površina. Ako je uključeno, promjena orijentacije tijekom izvođenja funkcije uspoređuje se s orijentacijom netom prije početka izvođenja funkcije kontrole sile.

Kod ove se funkcije također pozicija robota ili predmeta pozicionira kao i kod *Shaft Inserta*; nekoliko milimetara iznad površine te također postoji greška koja se može tolerirati u smislu odstupanja od određene osi. Međutim, budući da je riječ u pravokutnom presjeku, postoji i kutno odstupanje koje se može tolerirati, a iznosi oko  $\pm 2^\circ$ .

#### 4.1.4. Face Match

Još jedna bitna funkcija koju FANUC pruža je funkcija *Face Match*. Ona se koristi za precizno poravnavanje dviju površina kako bi se postigla njihova međusobna paralelnost. Funkcija koristi povratne informacije iz senzora sile kako bi robot mogao osjetiti kad su dvije površine u kontaktu. Ako površine nisu pravilno poravnate, robot automatski prilagođava svoju poziciju sve dok ne postigne pravilno poravnanje. Nakon što robot uskladi površine, koristi kontrolu sile kako bi primijenio odgovarajući pritisak.

**Tablica 4.4.** Glavni parametri funkcije *Face Match*

Stavka	Opis
Korisnički koordinatni sustav (engl. <i>User Frame</i> )	Odabrani korisnički koordinatni sustav za izvršavanje funkcije kontrole sile.
Koordinatni sustav alata (engl. <i>Tool Frame No.</i> )	Odabrani koordinatni sustav alata za izvršavanje funkcije kontrole sile.
Smjer primjene sile (engl. <i>Pushing direction</i> )	Odabir smjera primjene sile s obzirom na postavljeni korisnički koordinatni sustav.
Provjera orijentacije (engl. <i>Check Orientation Change</i> )	Mogućnost odabira provjere orijentacije tijekom izvođenja funkcije usklađivanja površina. Ako je uključeno, promjena orijentacije tijekom izvođenja funkcije uspoređuje se s orijentacijom netom prije početka izvođenja funkcije kontrole sile.
Prekid primjene momenta (engl. <i>Torque End Judgment Switch</i> )	Omogućuje korištenje funkcije za prekid izvođenja funkcije kontrole sile nakon provjere je li količina generiranog momenta smanjena na odgovarajuću razinu.
Najveći dopušteni moment (engl. <i>Max Torque</i> )	Numerička vrijednost koja se koristi definiranje graničnog momenta koji se koristi za stavku prekida primjene momenta. S tom se vrijednošću uspoređuje količina generiranog momenta.
Granična sila ostvarenja kontakta (engl. <i>Contact Force Threshold</i> )	Najmanja vrijednost sile kod koje robot prepoznaje da je uspostavio kontakt.
Primijenjena sila (engl. <i>Pushing force</i> )	Sila koja se primjenjuje nakon uspješnog uspostavljenog kontakta s površinom.
Najveća dopuštena promjena orijentacije (engl. <i>Change MAX Limit</i> )	Numerička vrijednost koja izražava najveću dopuštenu vrijednost promjene orijentacije u odnosu na početnu orijentaciju. U slučaju prekoračenja iste, javlja se alarm.

Kod ove funkcije pozicija se dovodi kao i kod funkcije *Constant Push*; par milimetara iznad same površine na kojoj se vrši funkcija.

#### 4.1.5. *Hole Search*

Sljedeća funkcija s kojom raspolazemo u FANUC-ovom asortimanu je *Hole Search*. Ova funkcija omogućuje robotu da pronade i precizno locira rupe na objektima. Najčešće primjene ovakvih potreba mogu se vidjeti kod linija koje zahtijevaju montažu vijaka ili u proizvodnji elektronike. Robot pri izvođenju ove funkcije prati otpor i kada alat ili predmet naiđe na rub rupe, robot prilagođava svoju putanju kako bi pronašao središte.

**Tablica 4.5.** Glavni parametri funkcije *Hole Search*

Stavka	Opis
Korisnički koordinatni sustav (engl. <i>User Frame</i> )	Odabrani korisnički koordinatni sustav za izvršavanje funkcije kontrole sile.
Koordinatni sustav alata (engl. <i>Tool Frame No.</i> )	Odabrani koordinatni sustav alata za izvršavanje funkcije kontrole sile.
Smjer primjene sile (engl. <i>Pushing direction</i> )	Odabir smjera primjene sile s obzirom na postavljeni korisnički koordinatni sustav.
Dubina završetka pretrage (engl. <i>Search End Depth</i> )	Numerička vrijednost koja označava udaljenost točke prilaska i dubinu završetka izvođenja funkcije pretrage.
Brzina pretrage umetanja (engl. <i>Search Insert Vel</i> )	Numerička vrijednost koja označava brzinu umetanja predmeta rada za vrijeme izvođenja funkcije pretrage.
Primjenjena sila (engl. <i>Pushing force</i> )	Numerička vrijednost sile koja se primjenjuje na površinu na kojoj se vrši izvođenje funkcije pretrage. Povećati ovu silu u slučaju neuspješnog umetanja unatoč pronalasku provrta/uvrta.
Veličina prostora pretrage (engl. <i>Size of Search Range</i> )	Jedan od osnovnih parametara pretrage. Numerička vrijednost koja određuje duljinu pretrage u određenom smjeru definiranih osi.
Najveća vrijednost trajanja pretrage (engl. <i>Search Time MAX Limit</i> )	Numerička vrijednost koja označava period od trenutka dodira površine na kojoj se izvodi funkcija do uspješnog završetka izvođenja funkcije. U slučaju prelaska definirane granice, javlja se alarm.

Kod ove funkcije pozicija se dovodi kao i kod funkcije *Constant Push*; nekoliko milimetara iznad same površine na kojoj se vrši funkcija.

#### 4.1.6. Phase Search

Posljednja funkcija koja se obradila u ovom radu je *Phase Search*. Ova funkcija koristi se za pronalaženje specifične faze rotacije ili kuta pozicije tijekom izvršavanja zadatka. Ovo omogućuje robotu da precizno locira određeni kut ili fazu rotacijskog predmeta kako bi se moglo postići točno usklađivanje. Ovakva primjena može se vidjeti u industrijama gdje je bitno precizno poravnavanje rotacijskih dijelova kao što je pri montaži zupčanika. Robot pri operaciji očitava vrijednosti sa senzora kako bi pronašao određeni kut. Nakon pronalaska kuta robot automatski prilagođava svoju poziciju kako bi se što bolje uskladio.

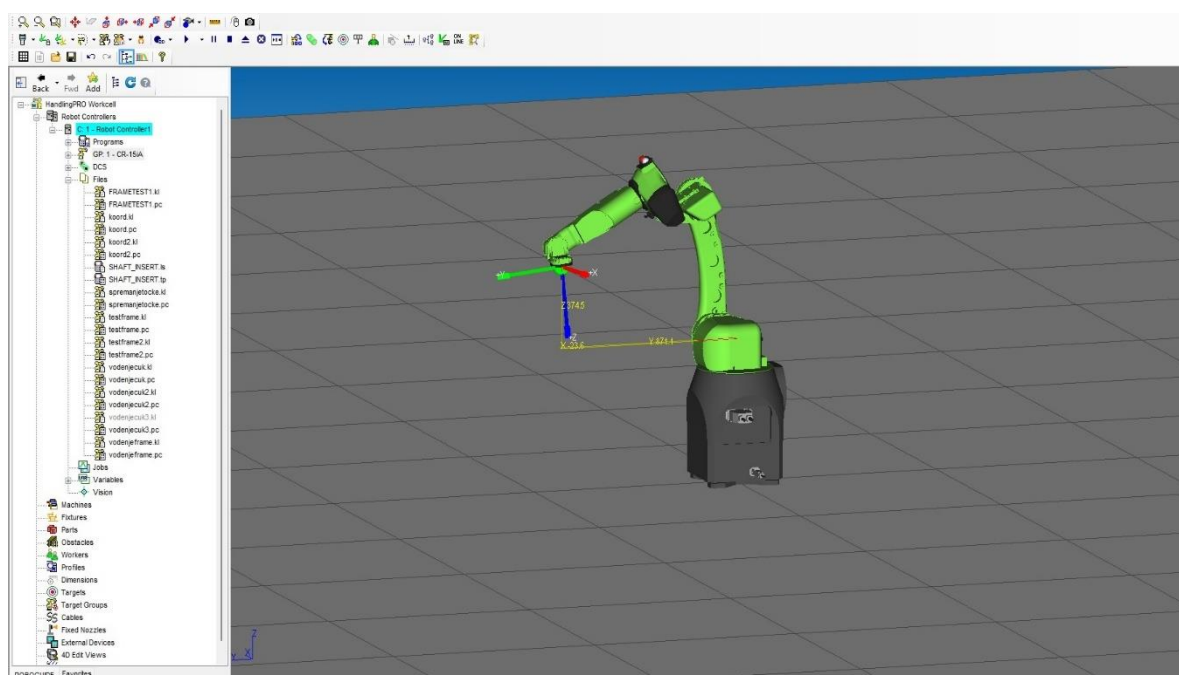
**Tablica 4.6.** Glavni parametri funkcije *Phase Search*

Stavka	Opis
Korisnički koordinatni sustav (engl. <i>User Frame</i> )	Odabrani korisnički koordinatni sustav za izvršavanje funkcije kontrole sile.
Koordinatni sustav alata (engl. <i>Tool Frame No.</i> )	Odabrani koordinatni sustav alata za izvršavanje funkcije kontrole sile.
Smjer primjene sile (engl. <i>Pushing direction</i> )	Odabir smjera primjene sile s obzirom na postavljeni korisnički koordinatni sustav.
Kutna brzina usklađivanja faza (engl. <i>PhaseMatch Ang. Vel</i> )	Numerička vrijednost koja određuje brzinu rotacije predmeta rada u svrhu usklađivanja faza. Povećati ovu vrijednost ako proces dugo traje, a funkcija je uspješno izvršena. Smanjiti ovu vrijednost u slučaju neuspjeha usklađivanja faza.
Moment usklađivanja faza (engl. <i>PhaseMatch Torque</i> )	Numerička vrijednost iznosa momenta koji se primjenjuje na površinu prilikom usklađivanja faza. Ako je prijedena točka podudaranja faza, smanjiti ovu vrijednost u koracima 10 – 20 %.
Ograničenje kuta rotacije usklađivanja faza (engl. <i>PhaseMatch Ang. Limit</i> )	Numerička vrijednost najvećeg dopuštenog zakreta oko rotacijske osi usklađivanja faza.
Promjena sile pritiska (engl. <i>Push Force Change Switch</i> )	Omogućuje uključanje funkcije koja izvršava variranje sile pritiska u smjeru umetanja u danim intervalima.

Kod ove funkcije pozicija se dovodi kao i kod funkcije *Constant Push*; nekoliko milimetara iznad same površine na kojoj se vrši funkcija.

## 4.2. SIMULACIJSKI SOFTWARE *ROBOGUIDE*

*FANUC*-ov *ROBOGUIDE* iznimno je koristan i ključan alat u industrijskoj automatizaciji koji inženjerima diljem svijeta daje mogućnosti projektiranja, optimizacije i provjere robotskih sustava. *ROBOGUIDE* djeluje kao virtualni prostor za razvoj i testiranje robotskih aplikacija u digitalnom okruženju prije nego što se primjene dovedu u stvarni svijet. Jedna od glavnih značajki *ROBOGUIDE*-a je njegova mogućnost izrade detaljnih 3D modela *FANUC* robota i njihovih okruženja. Inženjeri mogu simulirati kretanje robota, interakciju s objektima i cijeli proces proizvodnje s visokom točnošću i preciznošću. Osim toga, *ROBOGUIDE* podržava *offline* programiranje, što znači da se robotski programi mogu razviti i usavršiti bez potrebe za fizičkim robotom ili zaustavljanjem proizvodnje. Ova karakteristika ubrzava proces programiranja i skraćuje vrijeme potrebno za implementaciju u stvarni svijet. Uz *ROBOGUIDE*, inženjeri mogu optimizirati robotske tijekove rada, vremena ciklusa te korištenje resursa kroz iterativnu simulaciju i analizu. Validacijom dizajna u virtualnom svijetu korisnici mogu prepoznati i ispraviti potencijalne probleme prije nego što se pojave u stvarnom svijetu, smanjujući rizik i povećavajući učinkovitost.



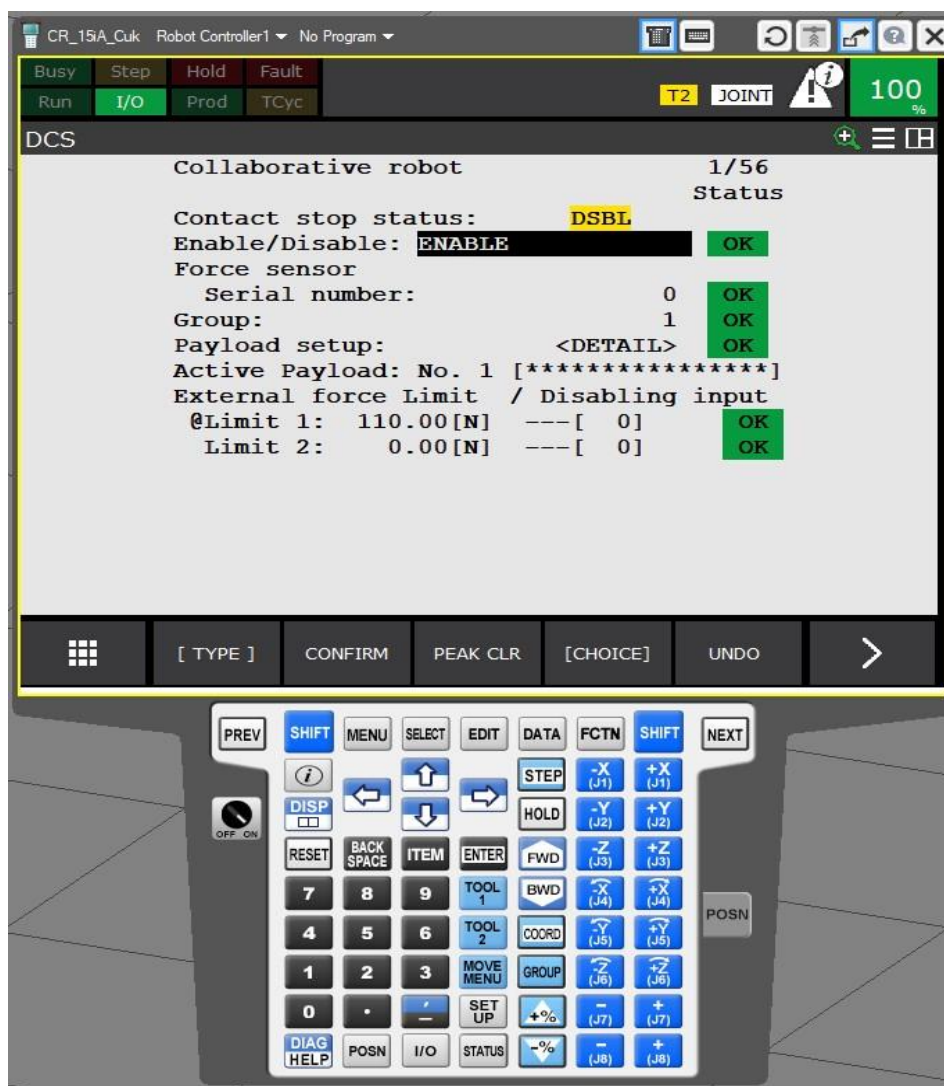
**Slika 4.4.** Korisničko sučelje *ROBOGUIDE*-a



#### 4.2.1. Dual Check Safety

*DCS (Dual Check Safety)* je sigurnosni sustav koji je razvila tvrtka *FANUC*, a dostupan je unutar softverskog alata *ROBOGUIDE* i preko privjeska za učenje. *DCS* omogućuje korisnicima da definiraju sigurnosne zone i postave ograničenja za kretanje robota kako bi se osigurala sigurnost radnika i opreme unutar proizvodnih pogona.

Kako bi robot funkcionirao kako treba, kod svakog paljenja robota treba uključiti *DCS* protokol kako bi se robot mogao kalibrirati te kako bi bio spreman za rad.



Slika 4.5. DCS prikaz na privjesku za učenje



### 4.3. O PROGRAMSKOM JEZIKU KAREL

U ovom se radu za programiranje robota koristi *FANUC*-ov *KAREL* programski jezik dizajniran specifično za programiranje i upravljanje industrijskim robotima. *KAREL* pruža napredno upravljanje robotima, komunikaciju s vanjskim uređajima te kompleksnije operacije i logiku u usporedbi s osnovnim metodama programiranja robota kao što je preko TP-a. *KAREL*-ova sintaksa slična je programskom jeziku *Pascal*. Takvi programski jezici u osnovi se sastoje od programske logike i podataka. Logika određuje redoslijed i način na koji se kod izvršava, dok podaci omogućuju realizaciju te logike. Na početku programa, unutar dijela *VAR*, potrebno je deklarirati sve podatke koji će se kasnije koristiti tijekom izvršavanja programa. Logika se izvodi između dijelova označenih s *BEGIN* i *END*. Nakon što je program napisan, potrebno ga je "izgraditi" pritiskom na tipku *Build*, čime se kod prevodi u oblik razumljiv robotskom sustavu. Funkcija *Build* generira p-kod (datoteka s nastavkom "*pc*") koji se može prenijeti na privjesak za učenje robota. *KAREL* se najčešće koristi unutar softvera *ROBOGUIDE*, iz kojeg se napisani program jednostavno može poslati na TP. Važno je napomenuti da se *KAREL* program može koristiti s programima napisanim na TP-u kao što ćemo i vidjeti kasnije.

#### 4.3.1. Tipovi podataka u KAREL-u

Kao i skoro svaki programski jezik, *KAREL* koristi varijable za spremanje podataka u memoriju. U tablici su navedeni osnovni tipovi podataka koji se koriste u *KAREL*-u pri programiranju robota te njihov kratki opis.

**Tablica 4.7. Tipovi podataka u KAREL-u**

Tip podatka	Opis
INTEGER	Tip podatka koji može poprimiti vrijednost cijelih brojeva u vrijednosti od -2147483647 do +2147483647
REAL	Tip podatka koji može poprimiti vrijednost realnih brojeva u vrijednosti od -2147483647 do +2147483647
ARRAY	Tip podatka koji može poprimiti vrijednosti niza podataka iste vrste.

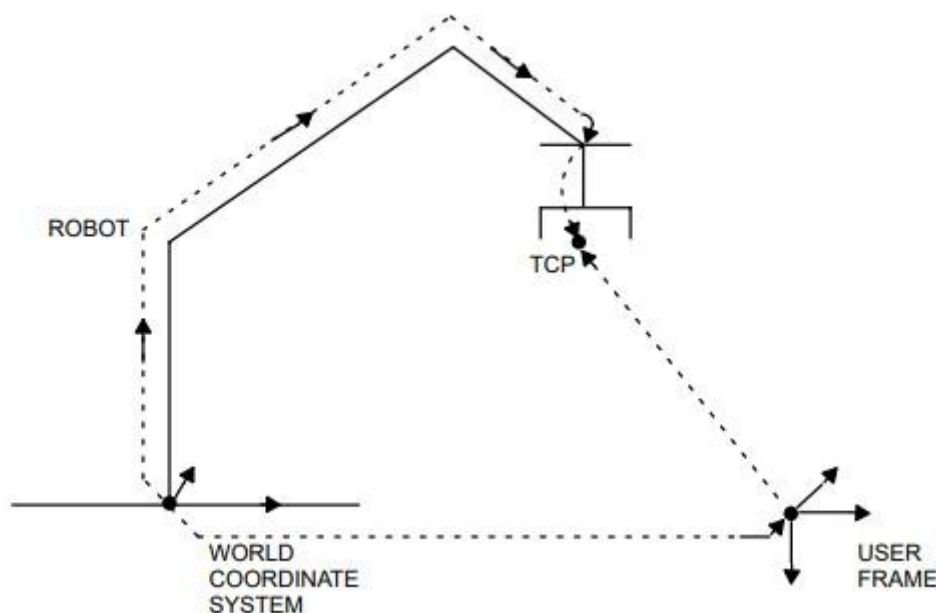
STRING	Tip podatka koji može primiti vrijednosti niza znakova, to jest tekst.
BYTE	Tip podatka koji može poprimiti vrijednost od 0 do 255.
BOOLEAN	Tip podatka koji može poprimiti vrijednosti <i>TRUE</i> (1) ili <i>FALSE</i> (0).
CONFIG	Tip podatka koja se koristi za pohranjivanje informacija o konfiguraciji robota. On pohranjuje specifične podatke koji definiraju položaj i orijentaciju robota u radnom prostoru, uključujući informacije o tome kako su postavljeni zglobovi robota i parametre za pravilno izvođenje kretanja.
XYZWPR	Tip podatka koji se sastoji od 6 realnih komponenti( <i>X</i> , <i>Y</i> i <i>Z</i> kartezijske pozicije, te <i>W</i> , <i>P</i> i <i>R</i> orijentacijske pozicije) te komponente koja specificira CONFIG tip podatka.
POSITION	Tip podatka koji sadrži matricu koja se sastoji od vrijednosti vektora normala, orijentacijskih vektora, lokacijskih vektora i vektora prilaza, te još uključuje komponentu za specificiranje “konfiguracijskog stringa”. Vrijedi napomenuti da je ovaj tip uvijek referenciran na određeni koordinatni sustav.

#### 4.3.2. Koordinatni sustavi robota

Kod industrijskih robota koristi se nekoliko različitih koordinatnih sustava koji omogućuju precizno definiranje pozicija robota u prostoru. Ti koordinatni sustav ključni su za ispravno programiranje i kontrolu robota jer omogućuju jednostavno određivanje kretanja bez obzira na složenost mehanike robota i njegove okoline. Glavni koordinatni sustavi kod industrijskih robota su sljedeći[14]:

- Kartezijski koordinatni sustav (engl. *World Coordinate Frame*) – najosnovniji i najintuitivniji sustav u kojem se robot kreće u pravcima duž tri osi: *X*, *Y* i *Z*

- Koordinatni sustav alata (engl. *Tool Centre Point*) – koordinatni sustav vezan uz alat na prihvatnici robota. Smjerovi  $X, Y$  i  $Z$  su definirani s obzirom na alat koji se nalazi na robotu
- Zglobni koordinatni sustav – Ovaj koordinatni sustav koristi zglobove robota za definiranje kretanja. Svaki zglob predstavlja jedan stupanj slobode i kontrolira se individualno.
- Korisnički koordinatni sustav (engl. *User Frame*) – Korisnik može definirati vlastiti koordinatni sustav koji se nalazi bilo gdje u radnom okruženju robota. Ovaj koordinatni sustav je ključan za realizaciju praktičnog zadatka.



Slika 4.6. Korisnički sustavi robota[14]

#### 4.3.3. Osnovne naredbe za razumijevanje koda

U ovom odjeljku izvedene su i objašnjene ključne naredbe za lakše razumijevanje i praćenje logike izvršavanja napisanog programskog koda.

- `FORCE_LINK(id_panela, url)` – uspostavlja vezu između dva objekta  
 Ulazni/izlazni parametri  
 [ulazni][obavezan parametar] `id_panela` : INTEGER  
 [ulazni][obavezan parametar] `url` : STRING
- `$GROUP[broj_grupe].$UFRAME=$MNUFRAME[broj_grupe, broj_korisnickog_koordinatnog_sustava]` – postavljenje korisničkog koordinatnog sustava

Ulazni/izlazni parametri

[ulazni][obavezan parametar] broj\_grupe : INTEGER

[ulazni][obavezan parametar] broj\_korisnickog\_koordinatnog\_sustava:  
INTEGER

- \$GROUP[broj\_grupe].\$UTOOL = \$MNUTOOL[broj\_grupe,  
broj\_koordinatnog\_sustava\_alata] – postavljanje koordinatnog sustava alata  
Ulazni/izlazni parametri  
[ulazni][obavezan parametar] broj\_grupe : INTEGER  
[ulazni][obavezan parametar] broj\_koordinatnog\_sustava\_alata:  
INTEGER
- \$GROUP[broj\_grupe].\$MOTYPE – postavljanje načina gibanja robota kroz pojedine točke  
Ulazni/izlazni parametri  
[ulazni][obavezan parametar] broj\_grupe : INTEGER
- \$GROUP[broj\_grupe].\$SPEED – zadavanje brzine robota  
Ulazni/izlazni paramteri  
[ulazni][obavezan parametar] broj\_grupe : INTEGER
- \$GROUP[broj\_grupe].\$TERMTYPE – zadavanje načina gibanja robota  
Ulazni/izlazni parametri  
[ulazni][obavezan parametar] broj\_grupe : INTEGER
- GET\_VAR(entry, ime\_programa, ime\_varijable, vrijednost, status) – omogućuje KAREL-u očitavanje vrijednosti određene varijable  
Ulazno/izlazni parametri:  
[ulazno/izlazni][obavezan parametar] entry : INTEGER  
[ulazni][obavezan parametar] ime\_programa: STRING  
[ulazni][obavezan parametar] ime\_varijable : STRING  
[izlazni][obavezan parametar] vrijednost: Bilo koji tip podatka iz KAREL-a osim PATH

---

[izlazni][obavezan parametar] status: INTEGER

- CURPOS(axis\_limit\_mask, ovr\_trv\_mask, broj\_grupe) – vraća kartezijsku poziciju ishodišta alata za specificiranu grupu osi

Ulazni/izlazni parametri:

[izlazni][obavezan parametar] axis\_limit\_mask : INTEGER

[izlazni][obavezan parametar] ovr\_trv\_mask : INTEGER

[ulazni][opcionalan parametar] broj\_grupe : INTEGER

- CALL\_PROG(ime\_programa, prog\_index) – omogućuje poziv drugog KAREL ili TP programa

Ulazni/izlazni parametri

[ulazni][obavezan parametar] ime\_programa: STRING

[ulazno/izlazni][opcionalan parametar] prog\_index : INTEGER

- GET\_POS\_REG(broj\_registra, status, broj\_grupe) – omogućuje čitanje određene pozicije robota iz zadanog pozicijskog registra

Ulazni/izlazni parametri

[ulazni][obavezan parametar] broj\_registra: INTEGER

[izlazni][obavezan parametar] status: INTEGER

[ulazni][opcionalan parametar] broj\_grupe: INTEGER

- SET\_POS\_REG(broj\_registra, pozicija, status, broj\_grupe) – omogućuje spremanje određene pozicije robota u zadani pozicijski registar

Ulazni/izlazni parametri

[ulazni][obavezan parametar] broj\_registra: INTEGER

[ulazni][obavezan parametar] pozicija: XYZWPR

[izlazni][obavezan parametar] status: INTEGER

[ulazni][opcionalan parametar] broj\_grupe: INTEGER

- POS(x, y, z, w, p, r, c) – iz zadanih vrijednosti stvara tip podataka POSITION

Ulazni/izlazni parametri

[ulazni][obavezan parametar] x, y, z, w, p i r : REAL

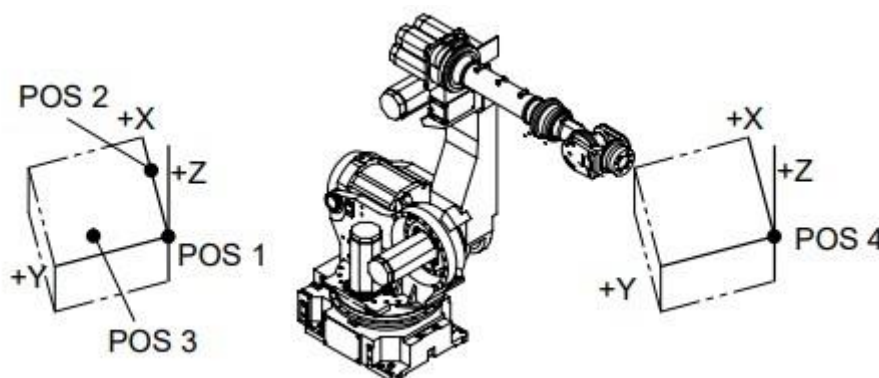
[ulazni][obavezan parametar] c: CONFIG

- FRAME(pozicija\_1, pozicija\_2, pozicija\_3, pozicija\_4) – koristi 3 POSITION argumenta za stvaranje novog koordinatnog sustava. Valja napomenuti da je pri pozivu ove funkcije bitan redoslijed i postav argumenata koji ulaze u nju. Upute za uspješno korištenje FRAME funkcije[14]:
  - Prva točka se smatra ishodištem sustava
  - X os je paralelna liniji od točke 1 do točke 2
  - XY ravnina je definirana da bude ravnina koja sadrži točku 1,2 i 3; tako da je točka 3 u pozitivnom djelu ravnine
  - Y os je okomita na X os te se nalazi u XY ravnini
  - Z os prolazi točkom 1 i okomita je na XY ravninu. Pozitivan smjer Z osi definiran je pravilom desne ruke.
  - Točke 1 i 2 moraju biti međusobno udaljene barem 10 mm, a točka 3 mora biti barem 10 mm udaljena od linije koja spaja točku 1 i 2

Ulazni/izlazni parametri:

[ulazni][obavezan parametar] pozicija\_1, pozicija\_2,  
pozicija\_3 : POSITION

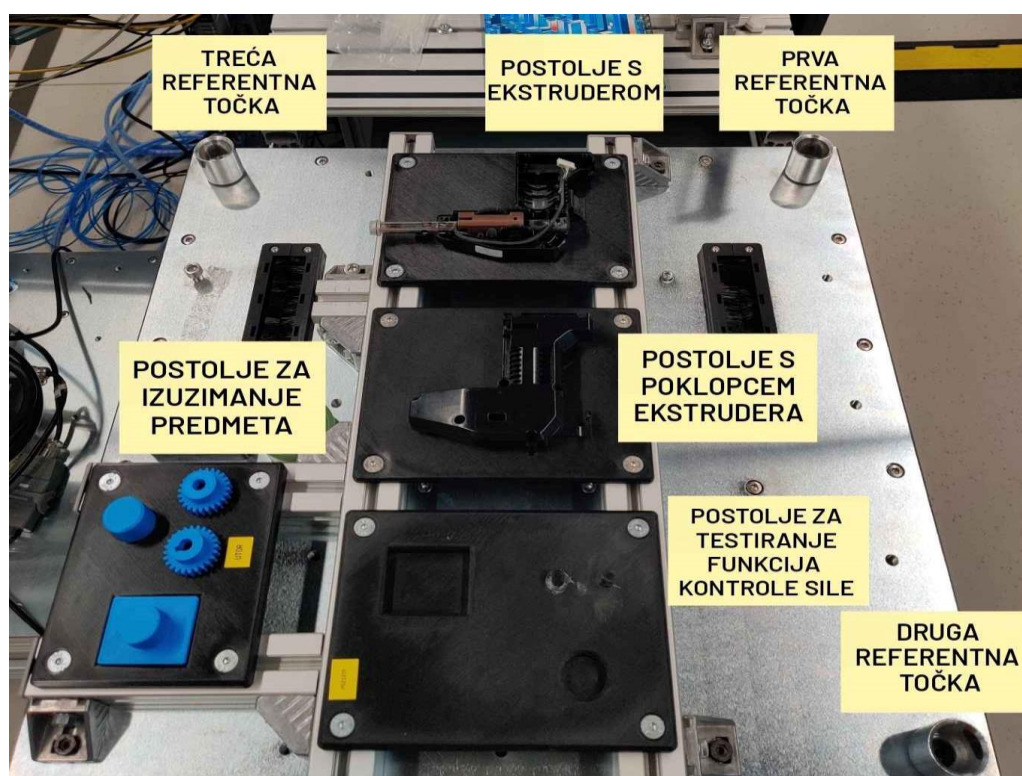
[ulazni][opcionalan parametar] pozicija\_4: POSITION



**Slika 4.7.** Prikaz rada FRAME funkcije[14]

#### 4.3.4. Opis postava

Postav na kojem robot izvodi programe sastoji se od radnog stola na kotačima (Slika 4.8.), fiksiranog stola (Slika 4.9.) te od konstrukcije za izmjenu alata (Slika 4.10.). Na gornjoj strani radnog stola na kotačima nalazi se metalna ploča na kojoj su montirana postolja za izuzimanje predmeta te za izvršavanje funkcija kontrole sile. U trima kutovima metalne ploče nalaze se metalni utori za cilindar. Ovi utori predstavljaju referentne točke korisničkog sustava kolica. Na postolju za izuzimanje nalaze se pravokutni i cilindrični utor, te dva cilindrična stupića. Za ispravan rad zadatka, kvadratna prizma, cilindar i zupčanici trebaju se nalaziti na ovome postolju. Postolje za testiranje funkcija također sadrži kvadratni i cilindrični utor te dva stupića. Međutim jedan od stupića na ovom postolju sadrži pravokutno izbočenje i na njega se tijekom izvršavanja zadatka montira zupčanic s utorom za pero. Ostala dva postolja služe se funkciju montaže ekstrudera. Jedno postolje drži poklopac za ekstruder, a na drugom postolju je pričvršćen sami ekstruder. Na fiksiranom stolu nalazi se utor za izuzimanje cilindra s pomoću kojeg pronalazimo referentne točke. Bitno je da se pri pokretanju programa cilindar nalazi u tom utoru. Na konstrukciji za izmjenu alata bitno je da se hvataljka za montažu ekstrudera nalazi u prvom mjestu za odlaganje s lijeva, te da je pravokutna izbočina na alatu okrenuta dalje od nas.

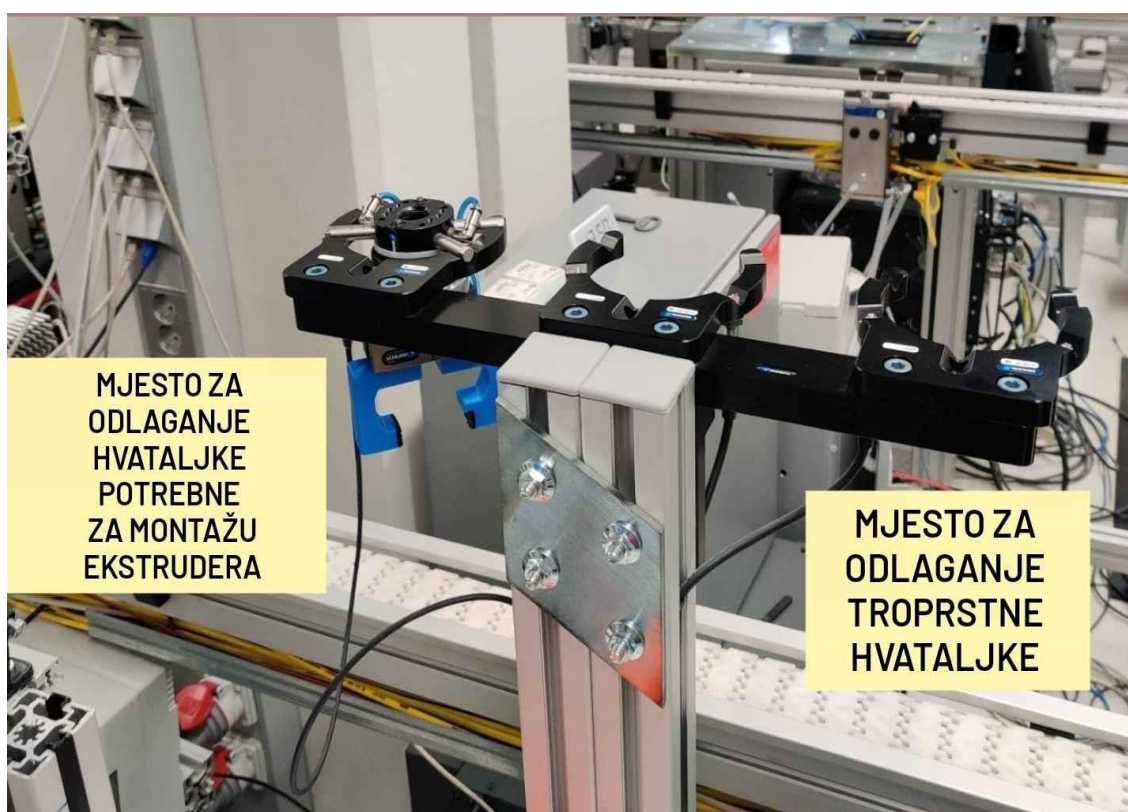


**Slika 4.8.** Tlocrt stola na kotačima





Slika 4.9. Fiksni stol



Slika 4.10. Konstrukcija za izmjenu alata



#### 4.3.5. Upravljački programi

U ovom se odjeljku prolazi i objašnjava kod dan u prilogu te se ukratko opisuju potprogrami koji se koriste u realizaciji ovog zadatka. Na samom početku koda definiramo varijable koje se kasnije koriste u izvođenju logike programa. Nakon toga se izvršava forsiranje korisničkog zaslona na *TP* te se definiraju početni koordinatni sustavi robota, način kretanja i brzina. Uz to se također definiraju i neki parametri vezani uz vođenje robota rukom. Nakon toga slijede uputstva koja ukratko objašnjavaju korisniku kako da se snalazi tijekom korištenja ovog koda. Nakon svakog odlomka korisnik je dužan pritisnuti tipku *F1* kako bi kod mogao nastaviti dalje.

Nakon uputstava, izvršava se potprogram na *TP*-u koji uzima cilindar iz utora te vrši funkciju *Face Match* kako bi uspostavio paralelnost površina i postigao što manje greške tijekom traženja referentnih točaka. Krajem završetka potprograma, vrši se postavljanje pomičnog koordinatnog sustava koji služi kako bi se realizirala funkcija vođenja robota. Ovaj se koordinatni sustav u svakom trenutku nalazi u vrhu hvataljke, a pomak ovisi o generiranoj sili koju detektira senzor sile.

Sljedeći dio koda ulazi u glavnu petlju koja se stalno izvršava sve dok se zadatak ne izvrši. Svakim prolaskom kroz ovu petlju osvježavaju se vrijednosti sila očitane sa senzora. Te se vrijednosti oduzimaju od prijašnjih kako bi se ta vrijednost pretvorila u pomak robota i kako bi se realiziralo vođenje. Zadane su neke maksimalne i minimalne vrijednosti koje se mogu dostići kako robot ne bi radio prevelike pomake.

```

IF FX-FX_0>M_MAX THEN p1.X= p1.X+KP*(R_X) ENDIF
IF FX-FX_0<M_MIN THEN p1.X= p1.X-KP*(R_X) ENDIF

IF FY_0-FY>M_MAX THEN p1.Y= p1.Y+KP*(R_Y) ENDIF
IF FY_0-FY<M_MIN THEN p1.Y= p1.Y-KP*(R_Y) ENDIF

IF FZ-FZ_0>M_MAX THEN p1.Z= p1.Z-KP*(R_Z) ENDIF
IF FZ-FZ_0<M_MIN THEN p1.Z= p1.Z+KP*(R_Z) ENDIF

MOVE TO p1 NOWAIT

```

U ovoj se petlji konstantno može mijenjati faktor pojačanja kretanja tijekom vođenja. Za brže kretanje je potrebno pritisnuti tipku *F5*, a za sporiju kretanju treba pritisnuti tipku *F3*. Također se može osvježiti *threshold* u slučaju nakupljanja greške gdje se robot samostalno giba u malim pomacima. Ova greška javlja se zbog fluktuacija senzora te zbog konstantnog uspoređivanja trenutne i prijašnje vrijednosti senzora. Razlika tih dviju vrijednosti koristi se pretvorbu pomaka i

ako ona nije približno jednaka nuli robot može raditi male pomake. Za izvršavanje ove funkcije potrebno je pritisnuti tipku *F4*.

Sada je na korisniku da robota dovede iznad prve referentne točke na stolu. Kada je korisnik zadovoljan s vođenjem, potrebno je pritisnuti tipku *NEXT*. Program pita korisnika za potvrdu zadovoljstva željene pozicije robota u usporedbi s referentnom točkom. Ako je odgovor negativan, robot se vraća u stanje vođenja. Ako je odgovor pozitivan, robot poziva drugi potprogram koji vrši funkciju *Shaft Insert* iznad trenutne pozicije. Kada se funkcija obavi, od korisnika se traži potvrda je li cilindar do kraja u utoru. U slučaju negativnog odgovora robot se podiže u pozitivnom smjeru osi Z te se vraća u funkciju vođenja. Ako je odgovor pozitivan, robot sprema točku u određeni pozicijski registar, izlazi iz utora i nastavlja s funkcijom vođenja. Ovaj proces pronalaska referentne točke treba se obaviti 3 puta.

```
$GROUP[1].$UFRAME = $MNUFRAME[1,2]
$GROUP[1].$UTOOL = $MNUTOOL[1,9]
$GROUP[1].$MOTYPE = JOINT
$GROUP[1].$SPEED = 400
pozicija=CURPOS(0,0) --čitanje trenutne pozicije robota

SET_POS_REG(80+broj, pozicija, STATUS) --spremanje pozicije u
određeni registar

WRITE('TOCKA zapamcena',CR)

$GROUP[1].$MOTYPE = LINEAR
$GROUP[1].$SPEED = 100
pozicija.z=pozicija.z+80
MOVE TO pozicija --izlaz iz utora
DELAY(300)
```

Nakon što je pronađena treća referentna točka, program korisnika obavještava kako je spremna izrada novog korisničkog sustava te se očekuje pritisak tipke 5 kako bi se to izvršilo. Nakon pritiska tipke 5, robot odlaže cilindar i poziva se potprogram koji služi za izradu novog koordinatnog sustava.

```
PROGRAM koord2

%NOLOCKGROUP

--deklaracija varijabli
VAR
    poz1,poz2,poz3:XYZWPR
    poz4,poz5,poz6,p_force: POSITION
    conf: CONFIG
    STATUS : INTEGER

BEGIN
    FORCE_LINK(TP_PANEL,'config=single')
    FORCE_SPMENU(TP_PANEL, SPI_TPUSER, 1)
    CNV_STR_CONF('nut', conf, STATUS)
    WRITE TPDISPLAY(CHR(128), CHR(137))
    $GROUP[1].$UFRAME = $MNUFRAME[1,2];
    $GROUP[1].$UTOOL = $MNUTOOL[1,9];

    poz1=GET_POS_REG(81,STATUS) -spremanje vrijednosti pozicijskog registra
    poz2=GET_POS_REG(82,STATUS) -spremanje vrijednosti pozicijskog registra
    poz3=GET_POS_REG(83,STATUS) -spremanje vrijednosti pozicijskog registra

    --pretvorba XYZWPR podataka u POSITION tip podatka
    poz4=POS(poz1.x, poz1.y, poz1.z, poz1.w, poz1.p, poz1.r, conf)
    poz5=POS(poz2.x, poz2.y, poz2.z, poz2.w, poz2.p, poz2.r, conf)
    poz6=POS(poz3.x, poz3.y, poz3.z, poz3.w, poz3.p, poz3.r, conf)

    p_force=FRAME(poz4,poz5,poz6) --poziv FRAME funkcije
    WRITE('Frame uspjesan', CR)
    DELAY(300)
    SET_POS_REG(99,p_force,status)
    $MNUFRAME[1,9]=p_force --postavljanje novog frame-a na TP

END koord2
```

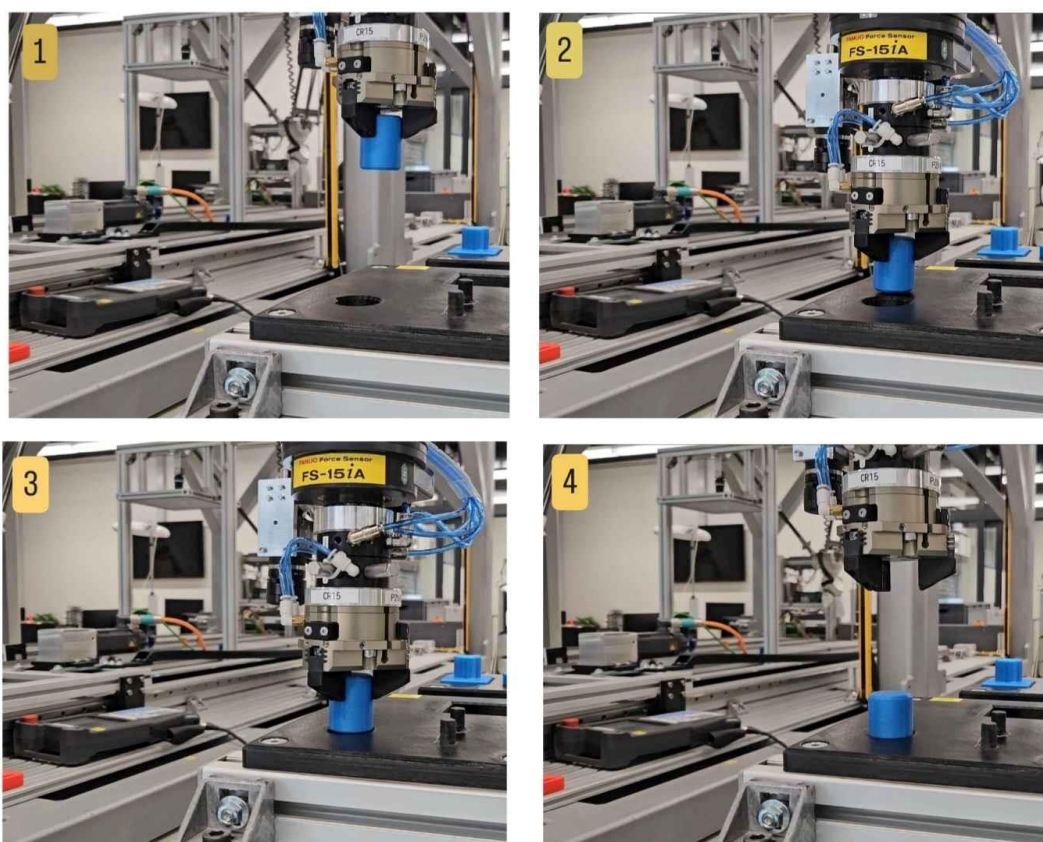
Nakon izrade novog korisničkog sustava, korisnika se pita ako je željan testirati radi li sve kako treba pozivom jednih od navedenih TP programa.

- *Hole Search*: Robot uzima cilindar iz utora na radnom stolu te odlazi do sljedećeg postolja kako bi pokušao pronaći rupu. Proces izvođenja funkcije *Hole Search* može se vidjeti u prilogu [1].

```

1:  UFRAME_NUM=9 ;
2:  UTOOL_NUM=9 ;
3:
4:  J P[1] 100% CNT100 ;
5:  J P[6] 100% FINE Offset,PR[22] ;
6:  L P[6] 100mm/sec FINE ;
7:  RO[5:OFF:GRIPPER CLOSE]=OFF ;
8:  WAIT .50(sec) ;
9:  RO[4:OFF:GRIPPER OPEN]=ON ;
10: WAIT .50(sec) ;
11: L P[6] 100mm/sec FINE Offset,PR[22] ;
12:
13: J P[3] 100% CNT100 ;
14: L P[2] 100mm/sec FINE ;
15: FORCE CTRL[8:osovinaHS]
    : ErrorLBL[1] ;
16: LBL[1] ;
17: FORCE CTRL[12:shafinsert2]
    : ErrorLBL[2] ;
18: LBL[2] ;
19: RO[4:OFF:GRIPPER OPEN]=OFF ;
20: WAIT .50(sec) ;
21: RO[5:OFF:GRIPPER CLOSE]=ON ;
22: L P[5] 100mm/sec FINE ;

```



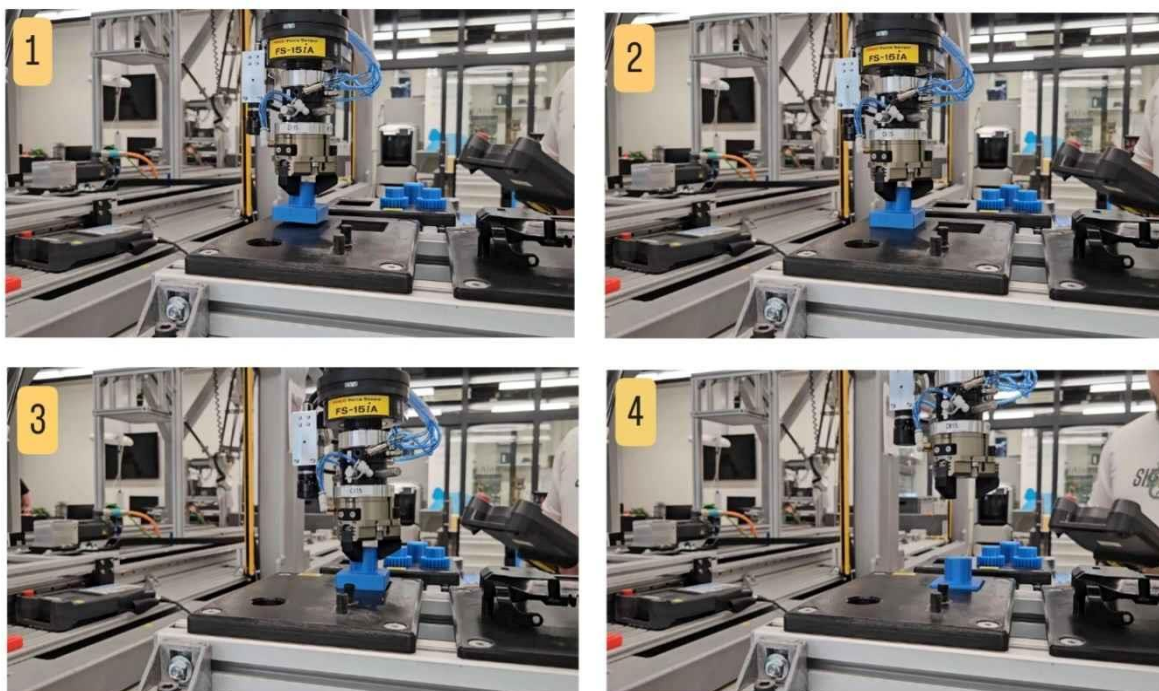
**Slika 4.11.** Proces izvršavanja programa *Hole Search*

- *Square Insert*: Robot uzima kvadratnu prizmu iz utora, poravnava je kako bi uskladio površine te ju stavlja u određeni utor. Proces izvođenja funkcije *Square Insert* može se vidjeti u prilogu [2].

```

1:  UFRAME_NUM=9 ;
2:  UTOOL_NUM=9 ;
3:  J P[1] 100% CNT100 ;
4:  L P[2] 100mm/sec FINE Offset,PR[22] ;
5:  L P[2] 100mm/sec FINE ;
6:  RO[5:OFF:GRIPPER CLOSE]=OFF ;
7:  RO[4:OFF:GRIPPER OPEN]=ON ;
8:  WAIT 1.00(sec) ;
9:  L P[2] 100mm/sec FINE Offset,PR[22] ;
10: J P[3] 100% CNT100 ;
11: L P[4] 100mm/sec FINE ;
12: FORCE CTRL[2:prizmaFM]
   : ErrorLBL[1] ;
13: LBL[1] ;
14: J P[5] 50% FINE ;
15:
16: L P[7] 50mm/sec FINE ;
17: FORCE CTRL[10:prizmaSI]
   : ErrorLBL[0] ;
18: RO[4:OFF:GRIPPER OPEN]=OFF ;
19: RO[5:OFF:GRIPPER CLOSE]=ON ;
20: WAIT 1.00(sec) ;
21: L P[6] 100mm/sec FINE Offset,PR[22] ;

```



**Slika 4.12.** Proces izvršavanja programa *Square Insert*

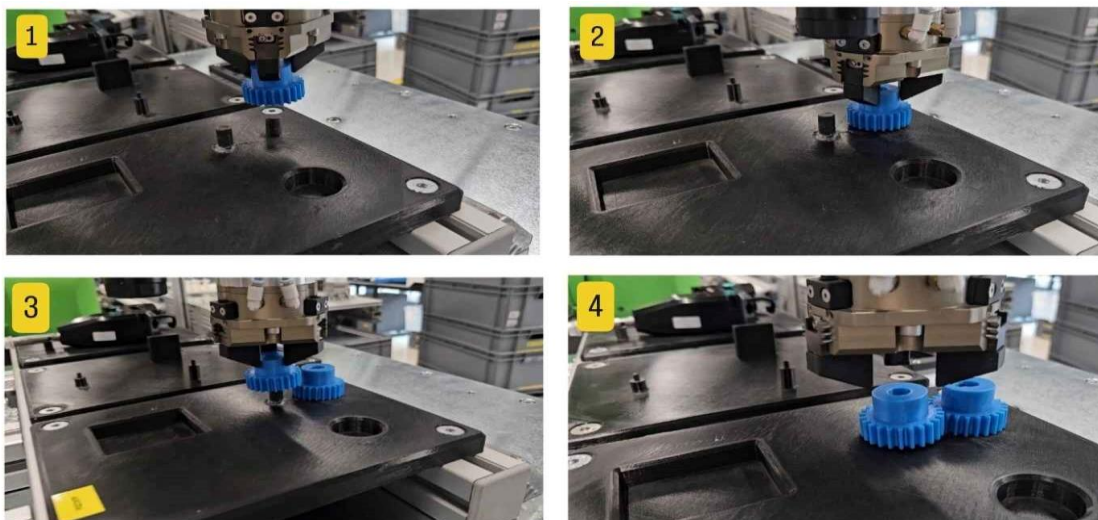


- *Phase Search*: Robot uzima zupčanik s utorom za pero te ga pokušava montirati na za to određenu osovinu. Nakon toga uzima drugi zupčanik te ga montira tako da pokuša ući u utor između zubaca prvog zupčanika. Proces izvođenja funkcije *Phase Search* može se vidjeti u prilogu [3].

```

1:  UFRAME_NUM=9 ;
2:  UTOOL_NUM=9 ;
3:
4: J P[1] 100% FINE ;
5: L P[8] 100mm/sec FINE Offset,PR[22] ;
6:
7: L P[8] 100mm/sec FINE ;
8: RO[5:OFF:GRIPPER CLOSE]=OFF ;
9: RO[4:OFF:GRIPPER OPEN]=ON ;
10: WAIT .50(sec) ;
11: L P[8] 100mm/sec FINE Offset,PR[22] ;
12: L P[3] 100mm/sec CNT100 Offset,PR[22] ;
13: J P[4] 100% CNT100 ;
14: L P[6] 100mm/sec FINE ;
15: FORCE CTRL[6:zupcanikPS]
    : ErrorLBL[1] ;
16: LBL[1] ;
17: FORCE CTRL[7:zupcanikSI]
    : ErrorLBL[2] ;
18: LBL[2] ;
19: RO[4:OFF:GRIPPER OPEN]=OFF ;
20: RO[5:OFF:GRIPPER CLOSE]=ON ;
21: L P[6] 100mm/sec FINE ;
22: J P[5] 100% CNT100 ;
23: L P[2] 100mm/sec FINE ;
24: RO[5:OFF:GRIPPER CLOSE]=OFF ;
25: RO[4:OFF:GRIPPER OPEN]=ON ;
26: L P[2] 100mm/sec FINE Offset,PR[22] ;
27: L P[7] 100mm/sec FINE ;
28: L P[10] 100mm/sec FINE ;
29: FORCE CTRL[6:zupcanikPS]
    : ErrorLBL[3] ;
30: LBL[3] ;
31: FORCE CTRL[7:zupcanikSI]
    : ErrorLBL[4] ;
32: LBL[4] ;
33: RO[4:OFF:GRIPPER OPEN]=OFF ;
34: RO[5:OFF:GRIPPER CLOSE]=ON ;
35: WAIT 1.00(sec) ;
36: L P[7] 100mm/sec FINE ;

```



**Slika 4.13.** Proces izvršavanja funkcije *Phase Search*

- Montaža ekstrudera: Robot vrši zamjenu hvataljke, dolazi do radnog stola te montira i pritišće poklopac na glavni dio ekstrudera kako bi se ustanovio dobar spoj. Nakon izvršenja montaže, ponovno mijenja hvataljku. Proces izvođenja montaže ekstrudera može se vidjeti u prilogu [4].

```

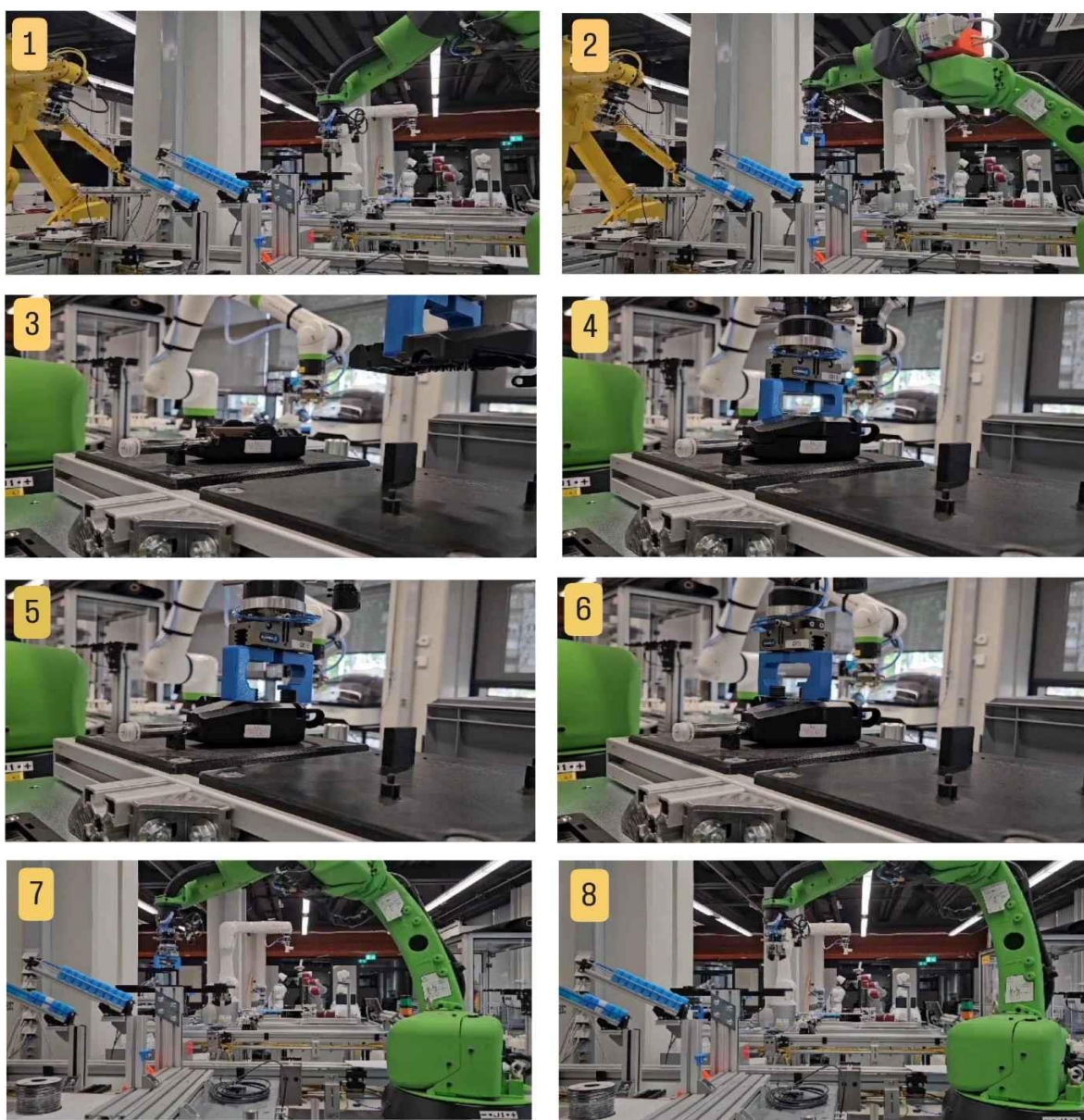
1:  UFRAME_NUM=2 ;
2:  UTOOL_NUM=9 ;
3: J P[1] 100% CNT100 ;
4: J P[2] 100% CNT100 ;
5: L P[3] 100mm/sec FINE ;
6: L P[4] 100mm/sec FINE ;
7: L P[5] 100mm/sec FINE ;
8:  RO[4:OFF:GRIPPER OPEN]=OFF ;
9:  RO[5:OFF:GRIPPER CLOSE]=OFF ;
10: RO[6:OFF:TOOL LOCK]=OFF ;
11: RO[7:OFF:TOOL UNLOCK]=ON ;
12: L P[6] 100mm/sec FINE ;
13: L P[7] 100mm/sec FINE ;
14: L P[10] 100mm/sec FINE ;
15: L P[11] 100mm/sec FINE ;
16: LBL[1] ;
17: RO[7:OFF:TOOL UNLOCK]=OFF ;
18: RO[6:OFF:TOOL LOCK]=ON ;
19: L P[9] 100mm/sec FINE ;
20: J P[25] 100% FINE ;
21: L P[12] 100mm/sec FINE ;
22: J P[8] 100% CNT100 ;
23: J P[13] 100% CNT100 ;
24: UFRAME_NUM=9 ;
25: RO[4:OFF:GRIPPER OPEN]=OFF ;
26: RO[5:OFF:GRIPPER CLOSE]=ON ;
27: J P[14] 100% CNT100 ;
28: L P[15] 100mm/sec FINE ;
29: L P[16] 100mm/sec FINE ;
30: RO[5:OFF:GRIPPER CLOSE]=OFF ;
31: RO[4:OFF:GRIPPER OPEN]=ON ;
32: L P[17] 100mm/sec FINE ;
33: J P[18] 100% CNT100 ;
34: L P[21] 100mm/sec FINE ;
35: L P[24] 100mm/sec FINE ;
36: WAIT 1.00(sec) ;
37: RO[4:OFF:GRIPPER OPEN]=OFF ;
38: RO[5:OFF:GRIPPER CLOSE]=ON ;
39: L P[18] 100mm/sec FINE ;
40: RO[5:OFF:GRIPPER CLOSE]=OFF ;
41: RO[4:OFF:GRIPPER OPEN]=ON ;
42: L P[19] 100mm/sec FINE ;
43: FORCE CTRL[1:CONSTpush]
    : ErrorLBL[0] ;
44: L P[20] 100mm/sec FINE ;
45: L P[22] 100mm/sec FINE ;
46: FORCE CTRL[1:CONSTpush]
    : ErrorLBL[0] ;
47: J P[23] 100% CNT100 ;
48: UFRAME_NUM=2 ;
49: J P[26] 100% CNT100 ;
50: J P[27] 100% CNT100 ;
51: J P[12] 100% FINE ;
52: L P[25] 100mm/sec FINE ;
53: L P[9] 100mm/sec FINE ;
54: L P[11] 100mm/sec FINE ;
55: RO[4:OFF:GRIPPER OPEN]=OFF ;

```

```

56:RO[5:ON:GRIPPER CLOSE]=OFF ;
57:RO[6:ON:TOOL LOCK]=OFF ;
58:RO[7:OFF:TOOL UNLOCK]=ON ;
59:L P[10] 100mm/sec FINE ;
60:L P[28] 100mm/sec FINE ;
61:J P[6] 100% CNT100 ;
62:J P[4] 100mm/sec FINE ;
63:L P[29] 100mm/sec FINE ;
64:L P[30] 100mm/sec FINE ;
65:RO[7:OFF:TOOL UNLOCK]=OFF ;
66:RO[6:ON:TOOL LOCK]=ON ;
67:L P[29] 100mm/sec FINE ;
68:L P[29] 100mm/sec FINE ;
69:J P[4] 100mm/sec CNT100 ;
70:J P[4] 100mm/sec CNT100 ;

```



**Slika 4.14.** Proces izvršavanja funkcije montaže ekstrudera



Nakon završetka potprograma, program korisnika pita želi li pozvati neki drugi potprogram. U slučaju pozitivnog odgovora korisnik se vraća na izbornik s potprogramima, a u slučaju negativnog odgovora program se završava. *Flow* dijagram za lakše razumijevanje programskog koda nalazi se u prilogu.

#### 4.3.6. Rezultati rada

Evaluacija razvijenog sustava je ključna za procjenu efikasnosti u stvaranju korisničkog koordinatnog sustava. Tijekom eksperimentiranja su uočene neke konstante koje je vrijedno napomenuti i navesti. Prilikom svakog vođenja robota primijećeno je da je za pronalaženje svih triju referentnih točaka, bez početnog poravnavanja cilindra, potrebno oko 3 do 4 minute; ovisno kolikom brzinom robot radi i kako se koriste faktori pojačanja. Vrijeme potrebno za ulazak, pamćenje pozicije i izlazak iz utora iznosi oko 30 sekundi po točki. Za precizno postavljanje korisničkog koordinatnog sustava s pomoću vođenja robota preko TP-a obično je potrebno 5 do 7 minuta, ovisno o iskustvu operatera. Također je bitno napomenuti da pri vođenju robota rukom, sila ne bi smjela preći granicu od 10 N (slika 4.11.) jer zbog toga dolazi do greške koju naknadno treba resetirati. Proces vođenja robota može se vidjeti pod prilogom [5].

Nakon dvadesetak testiranja pozicija na novoizrađenom koordinatnom sustavu, dobivena preciznost i točnost zapamćenih točaka je unutar granice od 0,1 mm do 0,5 mm.

Ovakav rezultat preciznosti iznimno je zadovoljavajuć s obzirom na to da se svaki potprogram može izvesti bez prevelikih promjena parametara. Ovakva visoka točnost postignuta je zbog fino obrađenih glodanih utora i cilindra. Preciznost ove razine teško se može postići standardnim načinom postavljanja korisničkog koordinatnog sustava gdje čovjek s pomoću TP-a robota dovodi od točke do točke. Ljudsko oko i pogreške tijekom dovođenja robota glavni su zaostaci ovog načina izrade korisničkog koordinatnog sustava.

SENSOR GRP 1		ATTACHMENTHAND
	Current	Mean
Fx	-7.567 N	-7.570 N
Fy	2.341 N	2.341 N
Fz	0.347 N	0.347 N
Mx	0.145 N*m	0.145 N*m
My	-0.050 N*m	-0.050 N*m
Mz	-0.021 N*m	-0.021 N*m

**Slika 4.15.** Sile tijekom vođenja robota

## 5. ZAKLJUČAK

U ovom radu demonstrirane su i objašnjene funkcije s kojima raspolaže *FANUC*-ov robot CR-15iA. Napisani su razni programi za praktični prikaz tih funkcija. Tijekom rada utrošeno je mnogo vremena na ispravno postavljanje parametara zbog nesavršenosti 3D printanih postolja i modela s kojima je robot rukovao. Iako je ovaj robot kolaborativne prirode, ne dolazi s već programiranom mogućnošću vođenja robota kao neki drugi *FANUC*-ovi modeli. Time smo limitirani na vlastiti kod koji robotu omogućuje kretanje po samo  $X, Y$  i  $Z$  osi; mijenjanje orijentacije i upravljanje pojedinih zglobova sadašnjim kodom nije moguće. To isto limitira mogućnosti postavljanja radnog stola jer se očekuje da je radni stol uvijek djelomično paralelan s bazom robota. Taj problem mogao bi se riješiti implementacijom vizijskog sustava kojim CR-15iA raspolaže. Kombinacija kamere i senzora sile robotu bi omogućila traženje referentnih točaka samostalno i bez vođenja.

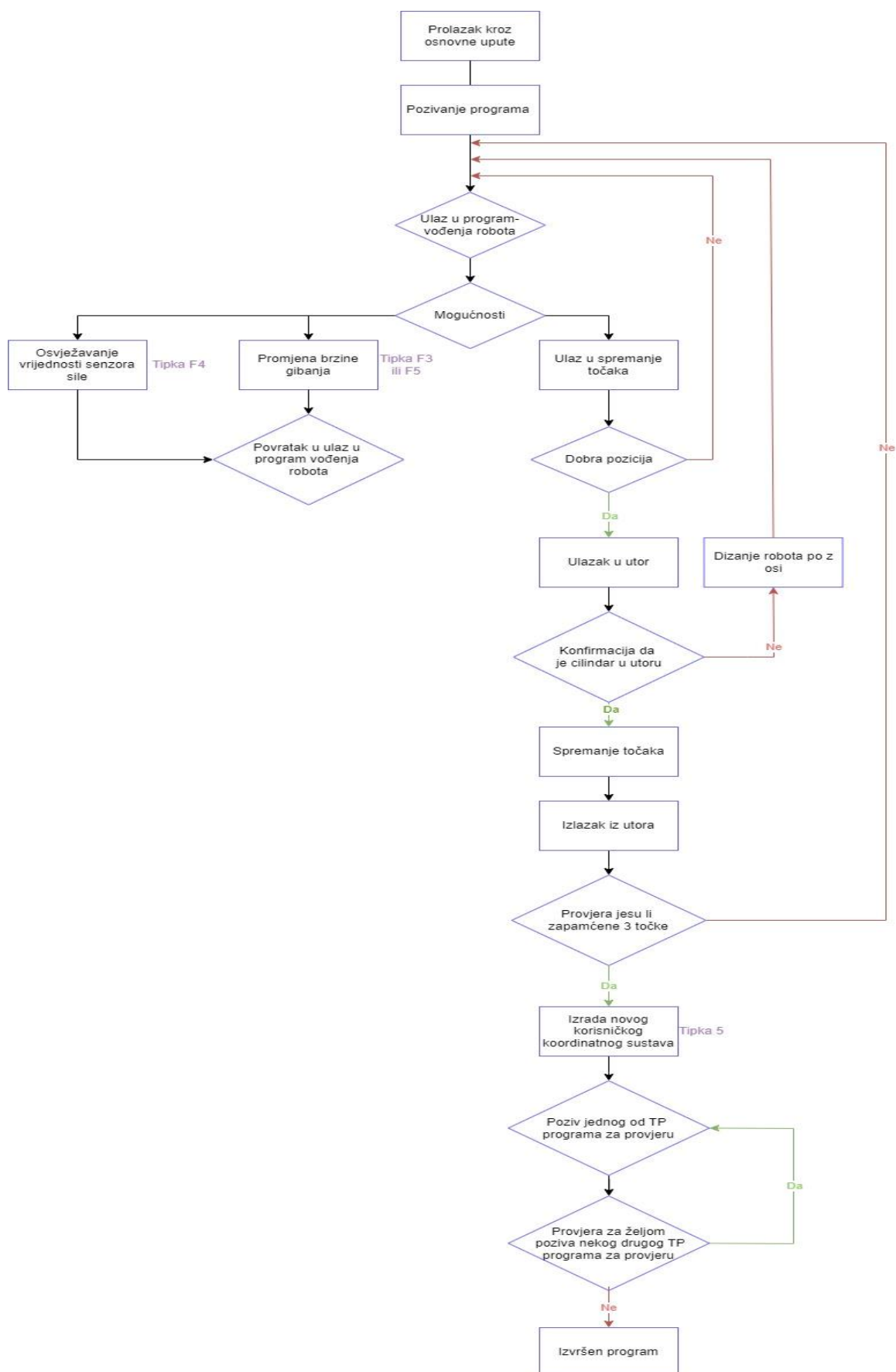
Napomena: Svi kodovi završnog rada i backup model robota nalaze se u literaturi [15].

---

**LITERATURA**

- [1] Gasparetto A., Scalera L.: From the Unimate to the Delta Robot: The Early Decades of Industrial Robotics, University of Udine, Italy, 2018., str.286, pristupljeno 09.09.2024.
- [2] Crneković M., Predavanja kolegija: Industrijski i mobilni roboti, FSB, Zagreb, 2024., pristupljeno 10.09.2024.
- [3] IFR: Demystifying Collaborative Industrial Robots, Frankfurt, 2018., pristupljeno 11.09.2024.
- [4] El Zaatari S., Marei M., Li W., Usman Z.: Cobot programming for collaborative industrial tasks: An overview, Coventry University, UK, 2018., str. 163-164, pristupljeno 11.09.2024.
- [5] Collaborative Robot CR-15iA datasheet: <https://www.fanuc.eu/be/en/robots/robot-filter-page/collaborativerobots/collaborative-cr15ia>
- [6] Gorinevski D. M., Formalski A. M., Schneider A. Y.; „Force Control Of Robotic Systems“, CRC Press, Boca Raton, New York; 1997.,pristupljeno 12.09.2024.
- [7] E. Baumann. Elektrische Kraftmesstechnik. VEB Verlag Technik, Berlin, 1976., pristupljeno 12.09.2024.
- [8] Erbacher, E. A.; „Force Control Basics“; „Industrial Robot: An International Journal“; 2000., str 20-22, pristupljeno 13.09.2024.
- [9] [https://www.fanuc.eu/~media/files/pdf/products/robots/brochures/mbr-02195-ro%20force%20sensor-v2/brochure\\_force\\_sensor\\_en.pdf?la=en](https://www.fanuc.eu/~media/files/pdf/products/robots/brochures/mbr-02195-ro%20force%20sensor-v2/brochure_force_sensor_en.pdf?la=en) , pristupljeno 13.09.2024.
- [10] <https://www.automate.org/robotics/tech-papers/working-with-robots-a-guide-to-the-levels-of-collaboration-between-humans-and-robots>, pristupljeno 13.09.2024.
- [11] <https://www.turbosquid.com/3d-models/3d-model-industrial-delta-robot-1716985>, pristupljeno 13.09.2024.
- [12] Primjena i mogućnosti kolaborativnog robota visoke ponovljivosti; Jurica Cvetić, Zagreb, 2022. , pristupljeno 14.09.2024.
- [13] FANUC Robot series; Force sensor; Operator's manual (B-83934EN/02), FANUC CORPORATION, 2017. pristupljeno 14.09.2024.
- [14] <https://therobotguylc.com/wp-content/uploads/2015/05/KAREL-Programming-Guide.pdf> ,pristupljeno 14.09.2024.
- [15] <https://github.com/ToncoBalonco/Završni>

## PRILOZI



---

```

PROGRAM vodenjecuk3
%NOLOCKGROUP
%NOPAUSESHFT
VAR  --deklaracija varijabli
    poz1,poz2,poz3: XYZWPR
    poz4,poz5,poz6, p_force: POSITION
    conf: CONFIG
    p_user,p_tool,p1,HOME,pozicija : XYZWPR
    prog_index,broj  IN DRAM: INTEGER
    KP, R_X, R_Y, R_Z,M_MAX, M_MIN: REAL
    FX_0,FY_0, FZ_0,FX, FY, FZ: REAL
    STATUS,entry : INTEGER
    UVJET: BOOLEAN

BEGIN

    FORCE_LINK(TP_PANEL,'config=single')
    FORCE_SPMENU(TP_PANEL, SPI_TPUSER, 1)
    $GROUP[1].$UFRAME = $MNUFRAME[1,2]
    $GROUP[1].$UTOOL = $MNUTOOL[1,9]
    $GROUP[1].$MOTYPE = JOINT
    --$GROUP[1].$TERMTYPE = FINE
    $GROUP[1].$SPEED = 400
    $GROUP[1].$TERMTYPE = NODECEL
    KP=1  --faktor pojaćanja
    M_MAX=0.5
    M_MIN=0.5
    broj=0
    UVJET = TRUE
    FX = 0; FY=0; FZ = 0; R_X = 0; R_Y = 0;R_Z = 0
    WRITE TPDISPLAY(CHR(128), CHR(137))
    WRITE('Pritisak F1',CR)
    WRITE('za daljnji tekst.',CR)

    WAIT FOR TPIN[129] = ON
    DELAY(200)

    WRITE TPDISPLAY(CHR(128), CHR(137))
    WRITE('Progam za izradu user frame-a.', CR)
    WRITE('Upute :', CR)
    WRITE('Robot prvo ulazi u fazu ravnanje cilindra,', CR)
    WRITE('za preciznije pamćenje pozicija', CR)

    WAIT FOR TPIN[129] = ON
    DELAY(200)

    WRITE TPDISPLAY(CHR(128), CHR(137))
    WRITE('Nakon ravnanja ulazi u fazu vodenja :', CR)
    WRITE('Robot rukom vodite do zeljenih tocaka', CR)
    WRITE('Za pamćenje zeljene tocke ', CR)
    WRITE('Pritisnite tipku NEXT', CR)
    WRITE('te potvrdite tipkom F1 ili ', CR)
    WRITE('ponovno uđete u fazu vodenja', CR)
    WRITE('pritiskom tipke F2', CR)

    WAIT FOR TPIN[129] = ON
    DELAY(200)

```

---

```

WRITE TPDISPLAY(CHR(128), CHR(137))
  WRITE('U fazi vodenja', CR)
  WRITE('mozete mjenjati brzinu .', CR)
  WRITE('robota pritiskom tipaka F3 i F5', CR)
  WRITE('Za osvjezavanje tresholda', CR)
  WRITE('pritisnite tipku F4', CR)

  WAIT FOR TPIN[129] = ON
  DELAY(200)

  WRITE TPDISPLAY(CHR(128), CHR(137))
  WRITE('Tocke pamтите redoslijedom kako', CR)
  WRITE('su napisane na ploči', CR)
  WRITE('Kada imate sve 3 tocke.', CR)
  WRITE('pritiskom tipke NEXT', CR)
  WRITE('ulazite u izradu', CR)
  WRITE('novog userframe-a.', CR)

  WAIT FOR TPIN[129] = ON
  DELAY(200)

  WRITE TPDISPLAY(CHR(128), CHR(137))
  WRITE('Nakon izrade mozete,', CR)
  WRITE('birati zeljeni program.', CR)
  WRITE('za testiranje novog framea', CR)

  WAIT FOR TPIN[129] = ON
  WRITE TPDISPLAY(CHR(128), CHR(137))
  DELAY(200)

  WRITE('RAVNANJE U TIJEKU', CR)
  CALL_PROG('RAVNANJE', prog_index)
  WRITE TPDISPLAY(CHR(128), CHR(137))
  WRITE('Vodenje omoguceno', CR)

vod::
--pocetak vodenja robota, stvaranje pomичnog koordinantnog sustava
CNV_STR_CONF('nut', conf, STATUS)
p_tool=POS(0,0,0,0,0,0,conf)
$GROUP[1].$UTOOL = p_tool
p_user=POS(0,0,0,0,0,0,conf)
$GROUP[1].$UFRAME = p_user
p1=CURPOS(0,0)
p1.w=0
$GROUP[1].$UFRAME = p1
P1=CURPOS(0,0)

$GROUP[1].$MOTYPE = JOINT
--$GROUP[1].$TERMTYPE = FINE
$GROUP[1].$SPEED = 400
$GROUP[1].$TERMTYPE = NODECEL
--čitanje vrijednosti senzora
GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[1]', FX_0, STATUS)
GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[2]', FY_0, STATUS)
GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[3]', FZ_0, STATUS)

  WHILE UVJET DO

```

---

```
--snizavanje faktora pojačanja
IF TPIN[132] = ON THEN --pritisak tipke F3
    KP=0.25
ENDIF

--povećavanje faktora pojačanja
IF TPIN[134] = ON THEN --F5
    KP=1
ENDIF

--osvježavanje vrijednosti sa senzora sile
IF TPIN[133] = ON THEN -- pritisak tipke F4

GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[1]', FX_0, STATUS)

GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[2]', FY_0, STATUS)

GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[3]', FZ_0, STATUS)
    DELAY(50)
ENDIF

GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[1]', FX, STATUS)
GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[2]', FY, STATUS)
GET_VAR(entry, '*SYSTEM*', '$CCC_GRP[1].$FS_FORCE[3]', FZ, STATUS)
DELAY(50)
R_X=ABS(FX-FX_0)
R_Y=ABS(FY-FY_0)
R_Z=ABS(FZ-FZ_0)

IF R_X<0.3 THEN
    R_X=0
ENDIF

IF R_Y<0.3 THEN
    R_Y=0
ENDIF

IF R_Z<0.3 THEN
    R_Z=0
ENDIF

IF R_X>3 THEN
    R_X=3
ENDIF

IF R_Y>3 THEN
    R_Y=3
ENDIF

IF R_Z>3 THEN
    R_Z=3
ENDIF

IF FX-FX_0>M_MAX THEN
    p1.X= p1.X+KP*(R_X)
ENDIF
```

```

IF FX-FX_0<M_MIN THEN
    p1.X= p1.X-KP*(R_X)
ENDIF

IF FY_0-FY>M_MAX THEN
    p1.Y= p1.Y+KP*(R_Y)
ENDIF

IF FY_0-FY<M_MIN THEN
    p1.Y= p1.Y-KP*(R_Y)
ENDIF

IF FZ-FZ_0>M_MAX THEN
    p1.Z= p1.Z-KP*R_Z
ENDIF

IF FZ-FZ_0<M_MIN THEN
    p1.Z= p1.Z+KP*R_Z
ENDIF

MOVE TO p1 NOWAIT
FX = 0
FY = 0
FZ = 0
R_X = 0
R_Y = 0
R_Z = 0

--Dio koda za spremanje pozicije željene točke
IF TPIN[135] = ON THEN    --pritisak tipke NEXT
    pon::

    WRITE TPDISPLAY(CHR(128), CHR(137))

    DELAY(500)

    WRITE('JE LI OVO ZELJENA TOCKA?', CR)
    WRITE('F1 - DA',CR)
    WRITE('F2 - NE',CR)
    WAIT FOR TPIN[129] OR TPIN[131]

    IF TPIN[129] = ON THEN

        WRITE TPDISPLAY(CHR(128), CHR(137))
        WRITE('TRAZENJE RUPE', CR)
        DELAY(500)
        CALL_PROG('SI1',prog_index)

        WRITE('JE LI CILINDAR U VALJKU?', CR)
        WRITE('F1 - DA',CR)
        WRITE('F2 - NE',CR)
        WAIT FOR TPIN[129] OR TPIN[131]

        IF TPIN[129] = ON THEN
            broj=broj+1
            $GROUP[1].$UFRAME = $MNUFRAME[1,2]
            $GROUP[1].$UTOOL = $MNUTOOL[1,9]

```



```

$GROUP[1].$MOTYPE = JOINT
-----$GROUP[1].$TERMTYPE = FINE
$GROUP[1].$SPEED = 400
pozicija=CURPOS(0,0)
SET_POS_REG(80+broj, pozicija, STATUS)
WRITE('TOCKA zapamcena',CR)
IF STATUS = 1 THEN
    WRITE('Greška pri citanju PR',CR)
    GO TO vod
ENDIF

$GROUP[1].$MOTYPE = LINEAR

$GROUP[1].$SPEED = 100
pozicija.z=pozicija.z+80
MOVE TO pozicija
DELAY(300)
GO TO vod
ENDIF

IF TPIN[131] = ON THEN
    WRITE('Ponovno vodenje',CR)
    broj=broj-1
    pozicija= CURPOS(0,0)
    SET_POS_REG(49, pozicija, STATUS)
    IF STATUS = 1 THEN
        WRITE('Greska citanja PR',CR)
        GO TO vod
    ENDIF

    pozicija.z=pozicija.z+80
    $GROUP[1].$MOTYPE = LINEAR
    $GROUP[1].$SPEED = 100
    MOVE TO pozicija
    DELAY(500)
    GO TO vod
ENDIF

ENDIF

ENDIF

IF TPIN[131] = ON THEN
    WRITE('Ponovno vodenje', CR)
    DELAY(1000)
    GO TO vod
ENDIF

--Kada su tri tocke zapamcene program obavjestava korisnika da je
sve spremno za izradu novog korisniskog sustava
IF broj=3 THEN
    broj=0
    WRITE TPDISPLAY(CHR(128), CHR(137))
    WRITE('Imate 3 tocke', CR)
    WRITE('Pritisnite tipku 5 za izradu',CR )
    WRITE('user frame-a', CR)
    DELAY(500)

```

```
ENDIF

IF TPIN[53] = ON THEN --pritisak tipke 5
    CALL_PROG('odlaganjecilindra',prog_index)
    WRITE TPDISPLAY(CHR(128), CHR(137))
    DELAY(2000)
    CALL_PROG('koord2',prog_index)
    WRITE TPDISPLAY(CHR(128), CHR(137))

    zad::
    WRITE('Koji program zelite pozvati',CR)
    WRITE('F1- Hole search',CR)
    WRITE('F2-Square insert',CR)
    WRITE('F3-Phase search',CR)
    WRITE('F4-Montaza ekstrudera',CR)
    DELAY(1000)
    WAIT FOR TPIN[129] OR TPIN[131] OR TPIN[132] OR TPIN[133]
    IF TPIN[129] = ON THEN

        WRITE TPDISPLAY(CHR(128), CHR(137))
        CALL_PROG('hssicuk',prog_index)

    ENDIF

    IF TPIN[131] = ON THEN

        WRITE TPDISPLAY(CHR(128), CHR(137))
        CALL_PROG('squarecuk',prog_index)

    ENDIF

    IF TPIN[132] = ON THEN

        WRITE TPDISPLAY(CHR(128), CHR(137))
        CALL_PROG('phasesearchcuk',prog_index)

    ENDIF

    IF TPIN[133] = ON THEN

        WRITE TPDISPLAY(CHR(128), CHR(137))
        CALL_PROG('ekstruder',prog_index)

    ENDIF

    WRITE('Zelite li pozvati drugi program?',CR)
    WAIT FOR TPIN[129] OR TPIN[131]
    WRITE('F1- DA',CR)
    WRITE('F2- NE',CR)
    IF TPIN[129] = ON THEN

        WRITE TPDISPLAY(CHR(128), CHR(137))
        GO TO zad

    ENDIF

    IF TPIN[131] = ON THEN
```

```
        WRITE ('Program izvršen')
        WRITE TPDISPLAY (CHR(128) , CHR(137))
        GO TO kraj

    ENDIF

ENDIF

ENDWHILE
kraj ::
END vodenjecuk3
```

- [1] <https://www.youtube.com/watch?v=S7BKzzrTcYk>
- [2] <https://www.youtube.com/watch?v=XmUqr7vynU0>
- [3] <https://www.youtube.com/watch?v=83kuJZS0etY>
- [4] <https://www.youtube.com/watch?v=7P6dfEToI1k&t=10s>
- [5] <https://www.youtube.com/watch?v=fINLA15zTQs>