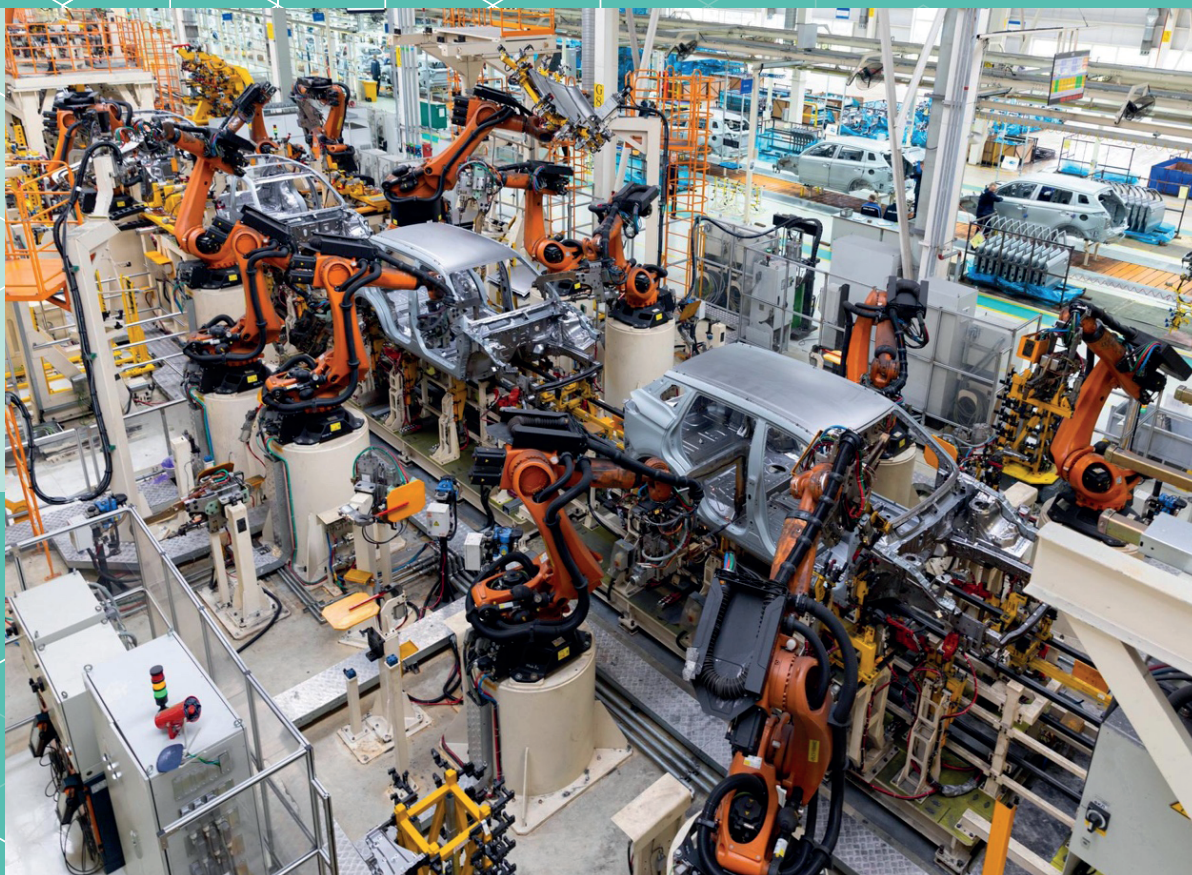




**EduSplit** Obrtna tehnička škola  
Regionalni centar kompetentnosti Split

# INDUSTRIJSKA ROBOTIKA

Priručnik za program obrazovanja Industrijska robotika



Dino Lozina



Dino Lozina

## INDUSTRIJSKA ROBOTIKA

Priručnik za program obrazovanja Industrijska robotika



Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Sadržaj publikacije isključiva je odgovornost Obrtne tehničke škole Split





**EduSplit** Obrtna tehnička škola  
Regionalni centar kompetentnosti Split

Dino Lozina

# INDUSTRIJSKA ROBOTIKA

Priručnik za program obrazovanja Industrijska robotika

Split, 2023.

Autor: Dino Lozina, mag. ing. comm. et inf. techn. (za Akritech d.o.o.)

Urednik: Igor Koletić, mag. ing. mech. (za ALGEBRA d.o.o.)

Naslov: **Industrijska robotika**

Fotografija naslovnice: Freepik

Recenzent: Jurica Kolić

Lektorica: Mira Barberić, prof

Grafičko oblikovanje: ALGEBRA d.o.o.

Nakladnik: Obrtna tehnička škola Split

Odgovorna osoba: ravnatelj Milivoj Kalebić

Za nakladnika: ALGEBRA d.o.o.

Više informacija:

Obrtna tehnička škola Split

Plančićeva 21

21000 Split

e-pošta: ured@ss-obrtna-tehnicka-st.skole.hr

mrežna adresa: edusplit.eu

ISBN: 978-953-50488-5-5

---

**Regionalni centar kompetentnosti Obrtne tehničke škole**

**Split, 2023.**

Obrtnička tehnička škola, Plančićeva 21, 21000 Split, OIB: 43651407703, nositelj je isključivog prava iskorištavanja ovog autorskog djela, prostorno, vremenski i sadržajno neograničeno, a koje pravo obuhvaća imovinska prava autora i to osobito, ali ne isključivo, pravo reproduciranja (pravo umnožavanja), pravo distribuiranja (pravo stavljanja u promet), pravo priopćavanja autorskog djela javnosti te pravo prerade. Pojedina imovinska autorska prava treća osoba može steći isključivo na temelju pisane suglasnosti Obrtničke tehničke škole.

# Sadržaj

<b>1. Uvod u industrijsku robotiku</b> .....	9
1.1. Kratka povijest industrijskih robota .....	9
1.2. Opis industrijskog lanca u proizvodnji .....	10
1.3. Uloga robota u industrijskom lancu .....	10
<b>2. Struktura i funkcionalnost industrijskih robota</b> .....	12
2.1. Komponente industrijskog robota: Pregled .....	12
2.2. Detaljna analiza FANUC industrijskog robota .....	12
<b>3. Koordinatni sustavi u industrijskoj robotici</b> .....	14
3.1. Opis tipova koordinatnog sustava .....	14
3.2. Koordinatni sustavi FANUC robota .....	15
3.2.1. World (apsolutni) koordinatni sustav .....	15
3.2.2. Joint (zglobni) koordinatni sustav .....	16
<b>4. Upotreba robotske ručne upravljačke jedinice: Vodič</b> .....	17
4.1. Opis grafičkog sučelja .....	17
4.1.1. Setup izbornik .....	20
4.1.2. Teach izbornik .....	20
4.1.3. Run izbornik .....	21
4.1.4. Utility izbornik .....	22
4.2. Opis upravljačkih elemenata .....	23
4.2.1. Ručno upravljanje .....	24
4.2.2. Numeričke tipke .....	24
4.2.3. Osnovne tipke - kratice .....	25
<b>5. Pokretanje robota u ručnom radu</b> .....	30
5.1. Pokretanje robota u „JOG“ načinu rada .....	30
5.2. Limiti u odnosu na automatski rad .....	35
<b>6. Središnja točka alata: Automatska detekcija i unos</b> .....	36
6.1. Vrste alata robota .....	37
6.2. Upisivanje TCP-a .....	38
6.3. Koordinatni sustav alata (TOOL) .....	39
6.4. Upisivanje TCP-a .....	40
6.5. Automatsko određivanje TCP-a .....	42

<b>7. Korisnički koordinatni sustav radnog objekta (engl. User Frame)</b> .....	44
7.1. Svrha korisničkog koordinatnog sustava .....	44
7.2. Definiranje korisničkog koordinatnog sustava radnog objekta .....	45
<b>8. Programiranje industrijskih robota</b> .....	48
8.1. Načini programiranja FANUC robota .....	48
8.2. Pisanje osnovnih programa .....	49
8.3. Pisanje programa za slaganje radnih objekata .....	53
8.4. Program za prebacivanje objekata (Podigni i spusti) .....	54
8.5. Naredbe za odabir alata i koordinatnog sustava .....	57
8.6. Definicija načina gibanja .....	58
8.7. Pozivanje drugih programa u programu koji je izvodi .....	63
<b>9. Pokretanje robota u automatskom radu</b> .....	65
9.1. Načini pokretanja robota u automatskom radu .....	65
9.2. Razlike u odnosu na pokretanje u ručnom radu.....	68
<b>10. Sigurnosno kopiranje i aktiviranje sigurnosne kopije</b> .....	69
10.1. Svrha sigurnosnog kopiranja .....	69
10.2. Izrada i aktiviranje sigurnosne kopije .....	69
<b>11. Integracija industrijskih robota u proizvodne lance</b> .....	73
11.1. Mogućnosti povezivanja robota s ostalim elementima u industrijskom lancu ...	73
11.2. Interakcija robota s PLC uređajima: Pregled i praksa .....	76
<b>12. Paket za simulaciju FANUC industrijskih robota: Uvod i uporaba</b> .....	79
12.1. Općenito o paketima za simulaciju .....	79
12.2. FANUC paket za simulaciju.....	80
12.3. Dizajniranje nove robotske ćelije .....	80
Popis literature .....	86



## 1. Uvod u industrijsku robotiku

### 1.1. Kratka povijest industrijskih robota

Industrijski roboti izumljeni su davne 1954. godine kada je američki izumitelj George Devol prijavio patent za „programirani prijenos dijelova“ (engl. *programmed article transfer*). Tako je rođena kompanija Unimation, prva koja se bavila industrijskom robotikom. Ta ista kompanija isporučila je General Motorsu 1961. godine prve robote koji su izvlačili proizvodne komade iz stroja za tlačni lijev. Roboti nisu nalikovali današnjim industrijskim robotima jer su se pogonili na hidraulički pogon, a tek kasnije su se mogli koristiti za ostale aplikacije u kojima je bilo potrebno prebacivati proizvodne komade i za elektrootporno točkasto zavarivanje karoserije auta.

Prvi veliki napredak dogodio se 1969. godine kada je dizajnirana Stanfordova ruka (engl. *Stanford Arm*). Prototip je razvio Victor Scheinman. Specifičnost toga robota je činjenica da ima šest stupnjeva slobode, pogonjen je električnom energijom i sami pokreti robota su upravljani najmoćnijim računalom toga vremena. Svi daljnji roboti su inspirirani dizajnom Stanfordove ruke.

Robot koji slični onima koje danas možemo naći u pogonima diljem svijeta, konstruiran je 1973. godine. Proizvela ga je kompanija ASEA, današnji ABB. Prvi put je robot upravljani mikrokontrolerom čime je omogućena kontinuirana putanja robota, što je posebno bitno za postupke zavarivanja. Kompanija ABB je i danas jedan od vodećih proizvođača industrijskih robota.

Sljedeća revolucija industrije omogućena je izumom robota od skupa usklađenih dijelova (engl. *selective compliance assembly*) - SCARA robota 1978. godine od strane Makina s Yamanashi sveučilišta u Japanu. Robot je imao četiri stupnja slobode gibanja i bilo ga je relativno lako proizvesti u dijelovima koji se kasnije sklapaju u jednu cjelinu. Dizajn samog robota je omogućavao montažu dijelova na jedno postolje male površine, koje zbog kinematike robota dozvoljava veliku brzinu i preciznost. SCARA robot je uvelike pridonio razvoju elektronike zbog i proizvodnje i potražnje za robotskim dijelovima, koji se velikim dijelom sastoje od elektroničkih komponenti i ostalih potrošnih materijala.

Već do 1984. godine postavljeni su principi industrijske robotike, povećao se broj proizvođača robota. Zahtjevi industrije za robotima bili su sve veći izazov u smislu brzine, točnosti i male težine. Od 1980. promijenio se pristup kako se izrađuju roboti, određeni proizvođači su krenuli u smjeru laganih i čvrstih struktura koje su omogućile dizajniranje paralelnih kinematičkih robota (engl. *parallel kinematic machines*) - PKM. Kod PKM robota alat je spojen s bazom robota pomoću tri ili šest podupirača. Ovi takozvani paralelni tipovi robota u mogućnosti su postići kratko vrijeme ciklusa, preciznost i velika opterećenja.

Roboti su postali neizostavan dio industrijskog procesa. Prosječna cijena robotskog sustava 2014. godine iznosila je 46.800 \$, a to je trećina cijene sustava jednakog kapaciteta 1990. godine. U isto vrijeme roboti su precizniji, brži te mogu biti pod većim opterećenjem.

## 1.2. Opis industrijskog lanca u proizvodnji

Industrijski lanac promatramo kao put od ideje do proizvodnje potrošačkih dobara u velikom opsegu. Najbolji primjer toga može biti automobilska industrija. Na početku procesa proizvodnje automobila proizvodimo dijelove karoserije od limenih ploča te je dijelove karoserije potrebno spojiti elektrootpornim točkastim zavarivanjem, zatim je potrebno cjelokupnu karoseriju obojati te montirati ostale dijelove poput vrata, vjetrobrana, sjedala i guma. Cijeli taj proces se ponavlja dokle god postoji potražnja za pojedinom serijom automobila. Industrijski roboti su izvrsni u poslovima koje je potrebno ponavljati. Dijelovi karoserije proizvode se određenom vrstom CNC strojeva (engl. *Computer Numerical Control*). Ovisno o vrsti proizvodnog lanca, industrijski robot može opsluživati CNC stroj limenim pločama: zadatak robota je da postavi ulaznu sirovinu u stroj te nakon proizvodnje preuzme gotov proizvod i proslijedi ga dalje u proizvodnu liniju. Na proizvodnoj liniji možemo imati niz robota koji se bave elektrootpornim točkastim zavarivanjem. Iz razloga što je način montaže pojedinih elemenata karoserije uvijek isti, roboti su idealni za taj dio proizvodne linije. Montirana karoserija dalje ide na bojanje koje također obavljaju roboti. Finalna montaža dijelova poput vjetrobrana, sjedala i guma, manje je automatiziran dio proizvodne linije te roboti mogu raditi u kolaboraciji s čovjekom da bismo na kraju dobili finalni proizvod, tj. automobil određenog proizvođača.

Kada gradimo industrijski lanac, moramo voditi računa o dva parametra, a to su brzina i kvaliteta proizvoda. Cilj nam je proizvesti čim je moguće brže što kvalitetniji proizvod. Da bismo ostvarili cilj, potrebno je posvetiti posebnu pozornost svakom dijelu u lancu proizvodnog procesa.

## 1.3. Uloga robota u industrijskom lancu

Poslove u industrijskom lancu može odrađivati čovjek, stroj ili robot. CNC strojevi preciznije i bolje od čovjeka mogu obraditi ulaznu sirovinu i mogu taj proces ponoviti s iznimnom preciznošću. Roboti nemaju preciznost ni ponovljivost CNC strojeva.

Zajednički rad čovjeka i robota omogućuje kombinaciju tipičnih snaga robota s mnogim snagama čovjeka. Tipične snage robota su izdržljivost, visok kapacitet tereta, preciznost i ponovljivost. Čovjekove snage trenutno je nemoguće replicirati bilo kojim robotom, a to su fleksibilnost prema novim zadacima, kreativno rješavanje problema i reakcija na nepredvidive situacije.

Kod primjera automobilskog industrijskog lanca spomenutog ranije, početak lanca je visoko automatiziran CNC strojevima i robotima. CNC stroj iznimno precizno proizvodi dijelove karoserije. Ti dijelovi mogu biti nezgodnog oblika i težine da bi ih čovjek mogao brzo i efikasno zavariti, što predstavlja idealan posao za robota. Roboti mogu biti korišteni kod bojanja karoserije, međutim kod delikatnih montiranja dijelova poput upravljačke ploče, vjetrobrana sjedala i guma, potreban je čovjek. Završni dio lanca manje je automatiziran jer je tu potrebna fleksibilnosti prema novim zadacima i reakcija na nepredvidive situacije - što su snage čovjeka.

Uloga robota u industrijskom lancu je u poslovima:

- rukovanja
- zavarivanja
- montaže
- bojanja.

Pod poslovima rukovanja podrazumijevamo procese koji uključuju hvatanje, transport, pakiranje i paletiziranje proizvoda. Kod svih procesa se koristi snaga robota da jako brzo izvršava radnje nad teškim teretom. Izazov kod procesa rukovanja je dizajn hvataljke robota.

Zavarivanje je proces koji spaja materijale pomoću topline, a nekada i pritiska. Tipične robotske aplikacije su za elektrootporno točkasto zavarivanje. Ručno zavarivanje zahtijeva stručne radnike jer čak i najmanje greške u zavarivanju mogu dovesti do teških posljedica. Nadalje, radnici su izloženi opasnim radnim uvjetima u smislu topline i para koje su štetne po zdravlje. Robot je u mogućnosti izvesti podjednaku kvalitetu zavarivanja jer ima visok stupanj ponovljivosti.

Procesi montaže formiraju 80% cijene proizvoda u proizvodnji, a povećanjem brzine montaže smanjujemo cijenu proizvoda. Određeni roboti imaju veliku brzinu što povećava kapacitet procesa montaže.

Slično kao kod zavarivanja, i kod bojanja su prisutni opasni uvjeti rada zbog isparavanja štetnih po zdravlje i eksplozivne atmosfere. Alat robota su pištolji za prskanje boje koji su u mogućnosti nanijeti ravnomjeran sloj boje na površinu uz što manju potrošnju boje.

## 2. Struktura i funkcionalnost industrijskih robota

### 2.1. Komponente industrijskog robota: Pregled

Industrijski roboti, iako raznoliki u dizajnu i mogućnostima, uglavnom dijele nekoliko ključnih komponenti koje određuju njihove funkcionalne sposobnosti. Dok specifične značajke robota mogu biti prilagođene određenoj aplikaciji, osnovne komponente obično uključuju:

- Mehaničku strukturu  
Ova struktura, često dizajnirana slično ljudskoj ruci, predstavlja osnovu robota. Uključuje motore, obično servomotore koji omogućavaju pokret. Broj motora, tipično četiri ili šest, određuje stupnjeve slobode gibanja robota.
- Elektroniku  
Elektronika upravlja motorima, omogućava precizno kontroliranje pozicije, brzine i akceleracije. Ključna komponenta ove sekcije je servokontroler smješten u glavnoj upravljačkoj jedinici robota koji prima signale od raznih senzora integriranih u robot.
- Računalnu jedinicu  
Ova komponenta služi kao mozak robota, interpretirajući ljudski jezik programiranja u upute koje robot može izvršiti. Kroz robotski kontroler, programeri mogu definirati koordinate i pokrete, dok računalna jedinica prevodi ove upute u niz podataka koji upravljaju motorima. Ova jedinica također je smještena u glavnoj upravljačkoj jedinici robota.

U tipičnom industrijskom okruženju, robotska ruka (mehanička struktura) je povezana s glavnom jedinicom koja pruža energiju i upravljačke signale. Glavna jedinica čini srce sustava, objedinjujući elektroničke i računalne komponente robota.

### 2.2. Detaljna analiza FANUC industrijskog robota

U svrhu dubljeg razumijevanja karakteristika i funkcija industrijskih robota, razmotrit ćemo model robota FANUC ER-4iA kao referentni model. Ovaj robot opremljen je s centralnom upravljačkom jedinicom R30-iB. Naša ciljana aplikacija usmjerena je na rukovanje: prenosimo cilindrične objekte s početne na određenu točku. U kontekstu ovakvih operacija, bitno je biti svjestan težine objekata kojima robot rukuje. U ovom scenariju svaki objekt teži manje od 1 kg. Važna je i procjena dosega robota kako bi se osiguralo da može lako pristupiti početnoj i određenoj točki.

Specifikacije FANUC ER-4iA uključuju:

- mehanička konfiguracija: robot ima 6 osi
- doseg: 550 mm
- maksimalna nosivost: 4 kg
- ponovljivost:  $\pm 0.01$  mm, što nam ukazuje na visoku preciznost kojom robot može ponavljati zadatke
- brzina zglobova:  $J1= 460^\circ/s$ ,  $J2= 360^\circ/s$ ,  $J3= 520^\circ/s$ ,  $J4=560^\circ/s$ ,  $J5=560^\circ/s$ ,  $J6=900^\circ/s$ .

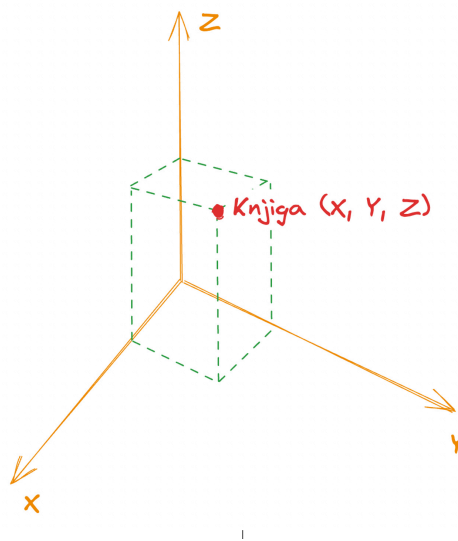
U proizvodnim procesima gdje je efikasnost presudna, brzina robota postaje ključni faktor. S druge strane, ponovljivost ukazuje na dosljednost i preciznost kojom robot izvršava zadatke, što je od vitalnog značaja u preciznim operacijama poput montaže.

## 3. Koordinatni sustavi u industrijskoj robotici

### 3.1. Opis tipova koordinatnog sustava

Kako se robot giba? Na koji način možemo robota naučiti da radi određene funkcije?

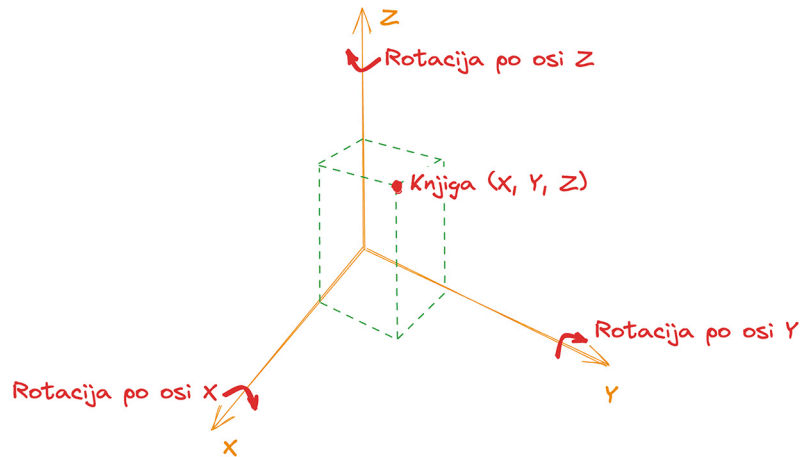
Da bi funkcionalno gibanje robota bilo moguće, potrebno je shvatiti koncept prostora. Dok čitate ovu knjigu, nalazite se u prostoru, knjiga koju čitate nalazi se na određenoj lokaciji u prostoru. Svaki predmet u prostoru ima svoju lokaciju. U slučaju da podignete knjigu, promijenili smo lokaciju knjige. Recimo da želite spremiti knjigu u ormar. Potrebno je znati u kojem smjeru trebate ići i koliko koraka do ormara trebate napraviti. Vaše gibanje od stola do ormara može se matematički izraziti koordinatnim sustavom. Ako pružite ruku ispred sebe, krećete se naprijed-nazad i tu os nazivamo **X os**. U slučaju da se krećete lijevo-desno, to je **os Y**. Ako pak podižete i spuštate ruku to je **os Z**. Prostor opisan na ovaj način pridodaje vrijednost X, Y i Z svakoj lokaciji. Sada možemo točno znati koliko milimetara se moramo pomaknuti u smjeru X, Y i Z da bismo spremili knjigu u ormar. Sustav u kojem lokacije, odnosno točke u prostoru opisujemo X, Y i Z koordinatama, nazivaju se **kartezijev koordinatni sustav**.



Slika 1. Kartezijev koordinatni sustav

Da bi industrijski roboti znali kamo se treba gibati, koriste kartezijev koordinatni sustav. Robotu je moguće zadati naredbu da dođe u određenu točku koja je opisana X, Y i Z koordinatama. Dakle ako želite programirati robota, potrebno mu je reći gdje se nalazi, kamo treba ići i kako se treba kretati i to na jeziku koji on razumije, a to je jezik koji kao svoju osnovu ima koordinatni sustav.

Osim kretanja naprijed-nazad, lijevo-desno i gore-dolje, svoju ruku možemo i rotirati. Zamislite da se vrtite oko svoje osi dok stojite na jednom mjestu – to je rotacija. Roboti se također mogu gibati oko svoje osi, što im omogućuje precizno pozicioniranje.



Slika 2. Rotacija u kartezijevom koordinatnom sustavu

## 3.2. Koordinatni sustavi FANUC robota

Industrijski roboti imaju više koordinatnih sustava koji nam kao programerima omogućuju da sa robotom dođemo u bilo koju poziciju u prostoru s velikom preciznošću.

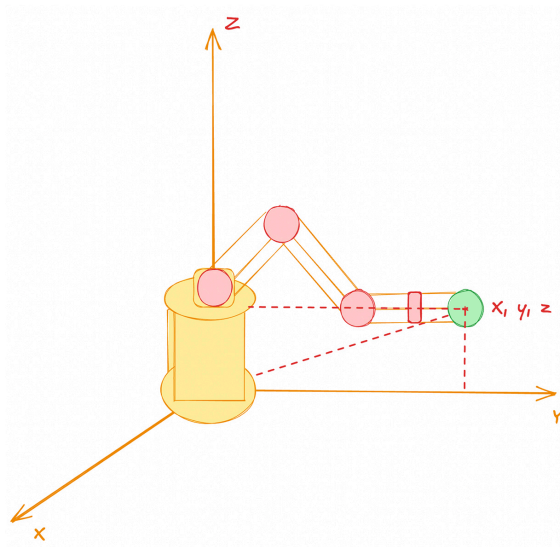
Fanuc robot posjeduje:

- WORLD apsolutni koordinatni sustav
- JOINT zglobni koordinatni sustav
- TOOL FRAME alatu orijentiran koordinatni sustav
- USER FRAME korisnički koordinatni sustav.

U narednim poglavljima koncentrirat ćemo se na WORLD i JOINT koordinatne sustave. Nakon što uvedemo pojam alata robota i krenemo s programiranjem, jako će nam koristiti TOOL i USER FRAME.

### 3.2.1. World (apsolutni) koordinatni sustav

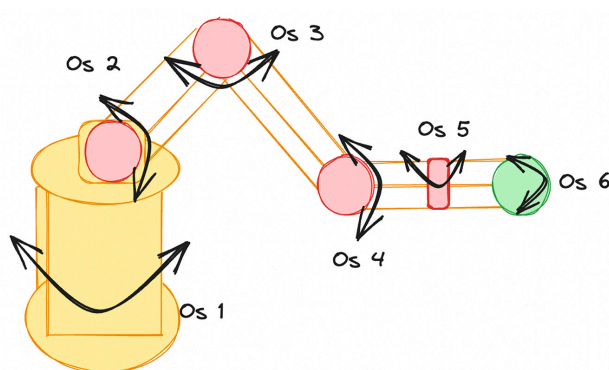
Ovaj sustav predstavlja globalni referentni koordinatni sustav, obično povezan s okolinom u kojoj se robot nalazi. World koordinatni sustav koristi kartezijev sustav kao referencu gibanja kroz prostor (X, Y i Z koordinate). Robot u svojoj bazi, prvoj osi, ima koordinate  $x=0$ ,  $y=0$ ,  $z=0$ . Sve ostale lokacije oko robota imaju različite koordinate u odnosu na tu osnovnu referentnu os. Recimo da se krećete po sobi, morate biti svjesni gdje je stolica, stol i vrata da biste izbjegli prepreke. Na isti način robotu možemo reći na kojim lokacijama u World koordinatnom sustavu se nalaze prepreke ili lokacije na koje robot treba doći.



Slika 3. Prikaz apsolutnog (engl. World) koordinatnog sustava

### 3.2.2. Joint (zglobni) koordinatni sustav

U ovom sustavu položaj robota definira se kutom svakog pojedinačnog zgloba na robotu. Zglobni koordinatni sustav izravno je povezan s fizičkom konstrukcijom robota i odnosi se na kutove svakog zgloba ili motora koji omogućavaju robotu kretanje. Ako zamislite vašu ruku i želite je ispružiti ravno ispred sebe, ne razmišljate o točnoj poziciji vaših prstiju u prostoru. Umjesto toga, razmišljate o tome kako ispružiti lakat i zglobove na ruci. Slično tome, u zglobnom koordinatnom sustavu, robot „razmišlja“ o tome koliko svaki od njegovih zglobova treba biti savijen ili rotiran da bi postigao željeni položaj.



Slika 4. Prikaz JOINT (zglobnog koordinatnog sustava)



## 4. Upotreba robotske ručne upravljačke jedinice: Vodič

Za upravljanje robotom i programiranje koristi se ručna upravljačka jedinica koja je spojena s robotom preko robotskog kontrolera. Na slici je prikazan privjesak FANUC robota.



Slika 5. Ručna upravljačka jedinica FANUC robota

Ručna upravljačka jedinica se sastoji od ekrana koji je osjetljiv na dodir te niza tipkala koji će biti opisani u sljedećim poglavljima.

### 4.1. Opis grafičkog sučelja

Na samom vrhu ekrana nalazi se statusna traka. Statusni indikatori igraju ključnu ulogu u pružanju brzog uvida u trenutno stanje i aktivnost robota.



Slika 6. Statusni prozor

Evo kratkog opisa statusnih indikatora:

- Busy - Ovaj indikator svijetli kada je robot zauzet izvođenjem neke operacije ili aktivnosti. To može biti pokretanje programa, izvođenje nekog zadatka ili čak interno procesiranje.
- Step - Svijetli kada je pritisnuta tipka STEP - u tom načinu rada robot izvodi program korak po korak. To je korisno za pronalaženje grešaka u programu ili provjeru programa jer omogućava korisnicima da pažljivo prate svaku operaciju koju robot izvodi.

- Hold - Kada je ovaj indikator aktivan, to znači da je rad robota privremeno obustavljen, ali nije potpuno zaustavljen. To može biti zbog nekog vanjskog signala, korisničkog unosa ili nekog drugog razloga.
- Fault - Ovaj indikator signalizira da postoji neki problem ili greška u radu robota. To može biti hardverska greška, softverska greška ili problem s programom koji se izvodi.
- Run - Kada je robot aktivan i izvodi program, ovaj indikator će svijetliti.
- I/O - Ovaj indikator svijetli kada se odvijaju ulazno/izlazne (I/O) operacije, što znači da robot komunicira s vanjskim uređajem ili sistemom.
- Prod - Ukazuje na „production mod“ ili proizvodni način rada, što znači da robot izvođa stvarne proizvodne zadatke.
- TCyc - je oznaka za „Tool Cycle“ ili ciklus alata. Ovaj indikator može svijetliti kada robot izvodi određeni ciklus ili operaciju s alatom koji koristi.

Drugi dio statusnog prozora odnosi se na brzinu robota, statusne poruke i koordinatni sustav u kojem robot radi.



Slika 7. Desni dio statusnog prozora

Kod Fanuc robota postoje dva načina rada (T1 i T2), koji se odnose na brzinu i sigurnost rada. T1 ili „Teach 1“ se koristi prilikom programiranja ili učenja robota. U T1 modu robot se kreće sporijom brzinom kako bi se osigurala veća sigurnost za programera koji je u blizini ili koji ručno vodi robota. Zbog smanjene brzine, programer može precizno postaviti robota na željene pozicije i stvoriti putanje pokreta bez rizika od ozljeda.

T2 ili „Teach 2“ također je način rada za programiranje, ali robot se u ovom modu kreće maksimalnom brzinom. Ovaj način rada se koristi kada smo već napravili testove na manjim brzinama i želimo postupno testirati kako se robot ponaša pri većim brzinama, sve do maksimalnih.

JOINT ili WORD predstavlja odabrani koordinatni sustav u kojem se robot kreće. Koordinatne sustave smo prošli u trećem poglavlju. Robot mora znati u kojem se koordinatnom sustavu nalazi da bi se mogao gibati u prostoru prema zadanim koordinatama.

Oznaka upozorenja se javlja u slučaju da se dogodi određena akcija ili greška na robotu za koju mi kao programeri moramo znati. Možemo pritisnuti oznaku upozorenja prikazanu na slici ispod da bismo dobili detaljnije informacije o statusu.



Slika 8. Detaljan opis greške (Nismo postavili parametar koji robotu govori s kojim opterećenjem radi u proizvodnom procesu - vrijednost u kg)

Zadnja stavka u statusnom prozoru je brzina kojom se robot giba. U slučaju da je postavljena vrijednost 100%, robot se giba maksimalnom brzinom ovisno o načinu rada. U slučaju da radimo u T1 ili T2 načinu rada, brzina je manja od stvarne brzine kojom se robot može gibati.

Ispod statusne trake nalazi se glavni izbornik. U glavnom izborniku mogu se podesiti minimalni konfiguracijski parametri da bi se robotom moglo upravljati i napisati osnovni program.



Slika 9. Prikaz glavnog izbornika

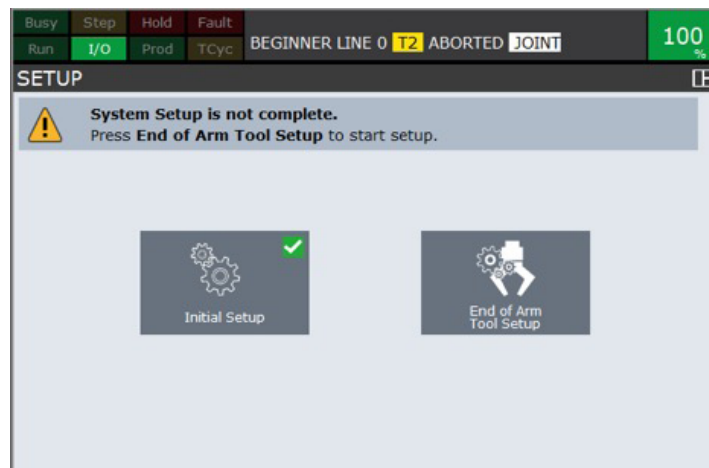
Glavni izbornik se sastoji od četiri sekcije:

- SETUP - osnovna podešavanja
- TEACH - mogućnost pisanja novih programa
- RUN - praćenje rada robota u proizvodnom procesu
- UTILITY - sigurnosno kopiranje i pomoć pri ručnom upravljanju robota.

### 4.1.1. Setup izbornik

U prvom dijelu izbornika „Initial Setup“ podešavamo regiju u kojoj se robot nalazi, vremensku zonu, vrijeme i podešavanje mreže. (Robot može biti spojen na lokalnu LAN mrežu ili internet).

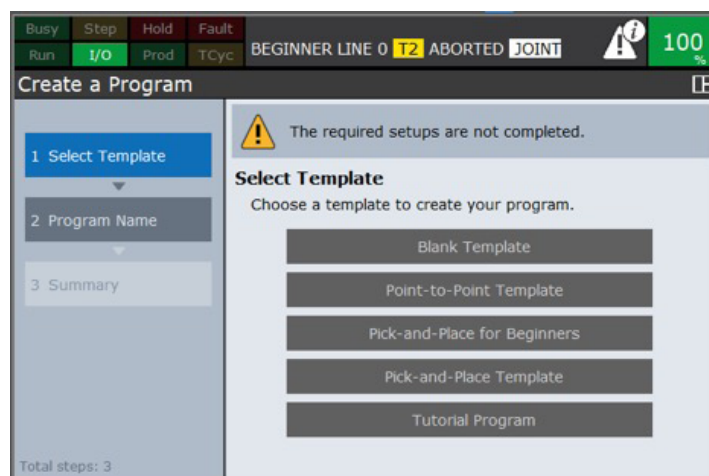
U drugom dijelu izbornika „End of Arm Tool Setup“ podešavamo alate robota koji su neophodni za rad robota. Više o alatu i podešavanju alata bit će riječi u idućim poglavljima.



Slika 10. Setup izbornik

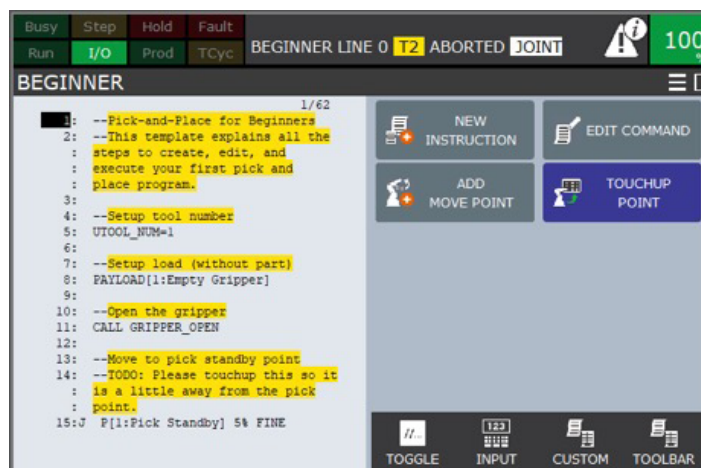
### 4.1.2. Teach izbornik

U ovom izborniku možemo odabrati izradu novog programa prema odabranom predlošku. Predlošci su već izrađeni programi za tipične aplikacije koje se koriste u industriji. Roboti su najčešće programirani da se gibaju od točke do točke (engl. *Point to Point*) na kojoj obavljaju određeni rad ili prebacuju radne komade s jednog mjesta na drugo (engl. *Pick and Place*). Prema potrebama aplikacije programer može izmijeniti predložak.



Slika 11. Inicijalizacija novog programa

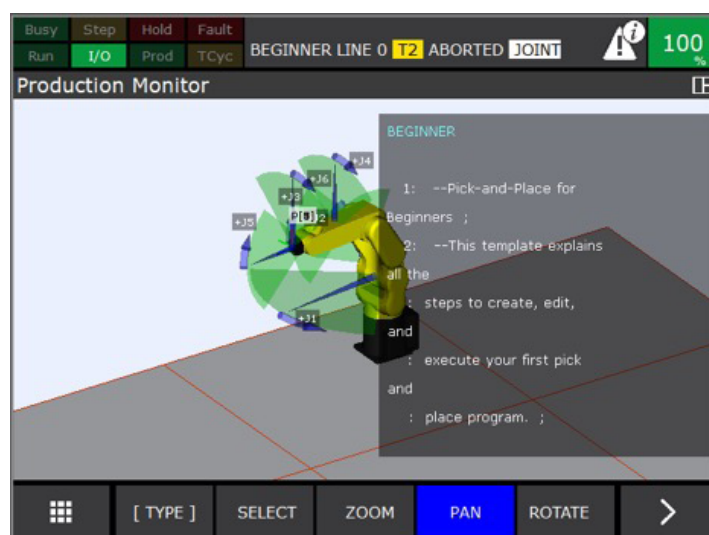
Nakon odabira predloška, dovoljno je dati programu ime koje se razlikuje od ostalih programa u robotu. Izbornik nam neće dopustiti da inicijaliziramo program s imenom koje već postoji. Ako u robotu već imamo inicijaliziran program, možemo mu pristupiti u izborniku (TEACH).



Slika 12. Prikaz programa koji je već inicijaliziran u robotu

### 4.1.3. Run izbornik

Kada se radi u proizvodnom procesu, jako je korisno da programer ili operator može pogledati koja se akcija ili linija programskog koda trenutno izvodi. Isto tako se može vidjeti koji je dio programa problematičan ako robot stane.



Slika 13. Prikaz robota u izborniku za praćenje rada

Na samom ekranu se vidi ime programa koji robot trenutno izvodi, komentari programa i slika koja prikazuje gibanje robota u prostoru.

#### 4.1.4. Utility izbornik

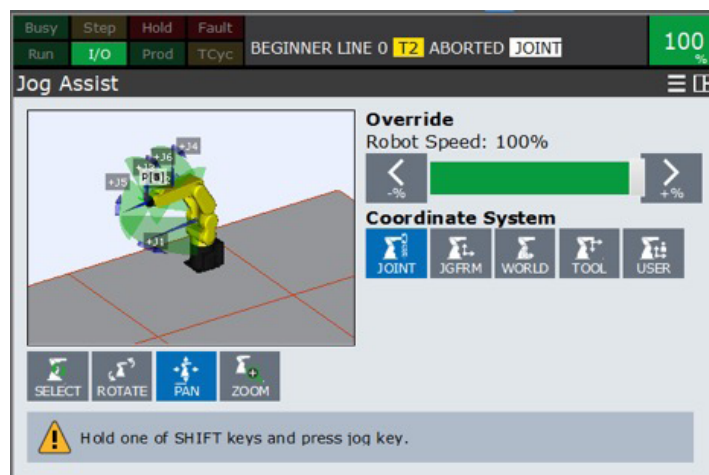
U ovom izborniku možemo napraviti sigurnosnu kopiju robota na USB uređaj. Korisno je imati sigurnosnu kopiju u slučaju da upravljačka ploča pregori ili se dogodi bilo kakva druga nepogoda koja nam može izbrisati program. Ako posjedujemo sigurnosnu kopiju, dovoljno je zamijeniti element robota koji je u kvaru i robot može nastaviti s radom.



Slika 14. Stvaranje sigurnosne kopije

Sigurnosna kopija se može spremiti na USB ili memorijsku karticu.

U UTILITY izborniku možemo odabrati JOG ASSIST izbornik. Ovaj izbornik pruža mogućnost početnicima da upravljaju robotom uz upute.

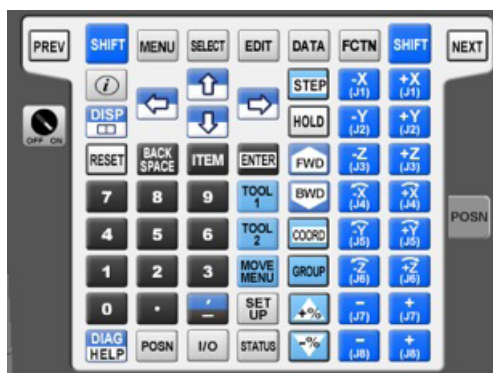


Slika 15. Upravljanje robota uz upute

Na ekranu se jasno vidi brzina robota, koordinatni sustav koji je odabran, 3D prikaz gibanja robota u prostoru te kratka uputa. Za gibanje robota potrebno je držati SHIFT tipku na ručnoj upravljačkoj jedinici. (Postoje dvije tipke SHIFT na lijevoj i desnoj strani upravljačke jedinice.)

## 4.2. Opis upravljačkih elemenata

Iako upravljačka jedinica ima zaslon osjetljiv na dodir, postoji opcija upravljanja robotom i pisanja programa pomoću tipkala. Roboti imaju puno postavki i mogu se prilagoditi raznim poslovima. Zbog toga upravljačka jedinica ima puno tipkala i opcija. U ovom poglavlju bit će opisane osnovne funkcije koje su potrebne za ručno upravljanje robotom. Nadalje ćemo objasniti kako dobiti osnovne informacije o poziciji robota te pregledati koji se programi nalaze u robotu.

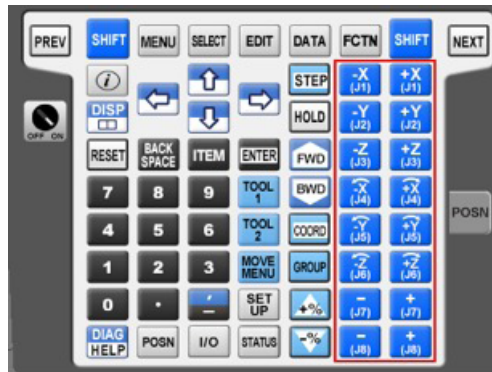


Slika 16. Upravljačka tipkala

Upravljački dio podijeljen je u tri dijela:

- ručno upravljanje
- numeričke tipke
- osnovne tipke - kratice.

### 4.2.1. Ručno upravljanje

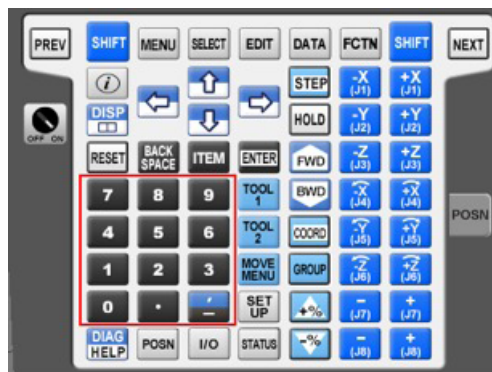


Slika 17. Ručno upravljanje

Ovaj set tipki se koristi za ručno upravljanje robotom. Svaka pojedina tipka odgovara jednoj koordinati robota. U slučaju da se nalazimo u WORLD koordinatnom sustavu, obraćamo pozornost na X, Y i Z oznake, što čini prvi set od šest tipki. Plus i minus određuju smjer u kojem će se robot gibati po jednoj od osi. Drugi set od šest tipki ima također oznake X, Y i Z, ali s dodatnom strelicom poviše oznaka osi. Robot u WORLD koordinatnom sustavu ima mogućnost da se rotira oko pojedine osi. Tipkalo sa strelicom poviše osi označava gibanje robota oko pojedine osi. Rotacije se kod FANUC robota označavaju sa w, p i r. Oznaka pozicije w označava rotaciju oko osi X, p označava rotaciju oko osi Y i r označava rotaciju oko osi Z. (Trenutna pozicija robota i rotacije prikazane su u svakom trenutku na zaslonu.)

Ako odaberemo JOINT koordinatni sustav, onda obraćamo pozornost na oznake J, od J1 do J6. Većina industrijskih robota ima šest osi. U JOINT koordinatnom sustavu, kako smo opisali u poglavlju ranije, pomičemo svaku os robota posebno. Na alat robota može se spojiti nekoliko dodatnih osi gibanja, a tipkama s oznakom J7 i J8 se može podešavati upravljanje tim dodatnim osima.

### 4.2.2. Numeričke tipke



Slika 18. Numeričke tipke



Nešto više o upotrebi numeričkih tipkala bit će riječi u idućim poglavljima kada se više upoznamo s načinom na koji se robot programira. Fanuc roboti spremaju podatke u niz registara čija je lokacija u memoriji označena brojkama. Snimljene pozicije na kojima se robot nalazi imaju niz numeričkih vrijednosti. Sve navedene parametre možemo izmijeniti numeričkim tipkalima.

### 4.2.3. Osnovne tipke - kratice

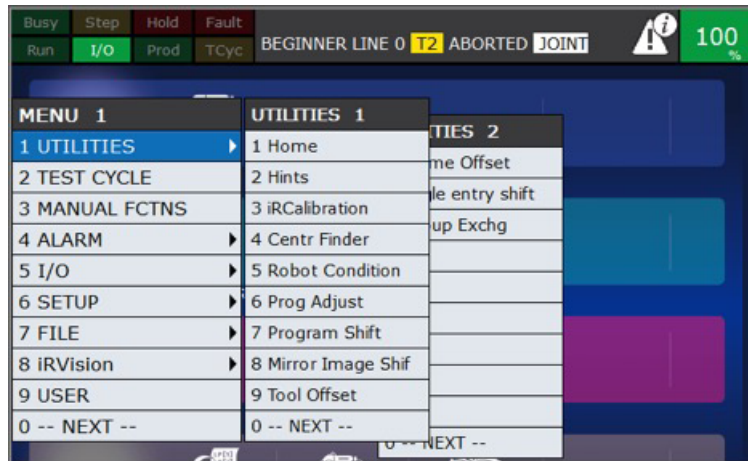


Slika 19. Osnovne tipke - kratice

Ovaj set tipkala otvara na ekranu najčešće korištene funkcije robota:

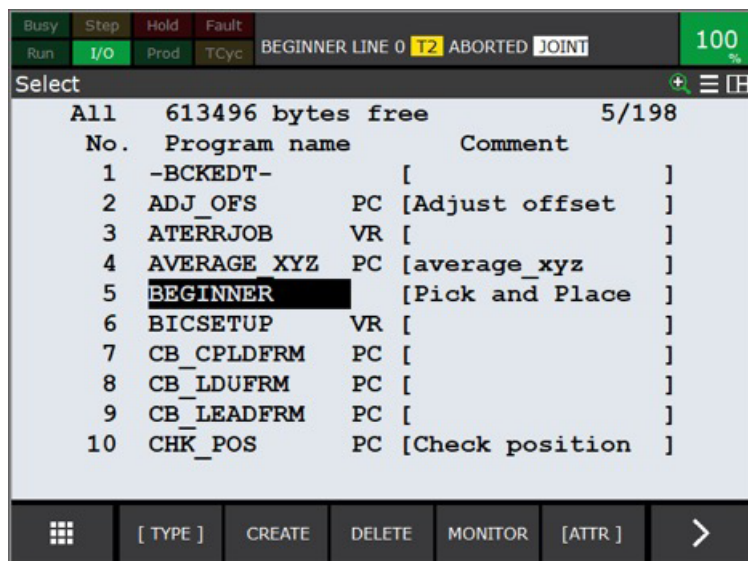
- MENU - prikaz svih opcija robota
- SELECT - odabir programa
- EDIT - izmjena programa
- DATA - memorijski registri
- FCTN - prikaz svih funkcija robota.

Tipkalo MENU otvara napredni izbornik robota. Od ove točke možemo doći do bilo koje funkcije ili parametra robota. Primjer može biti podešavanje vizijskog sustava (engl. *iRVision*) ili podešavanje ulazno-izlaznih jedinica (engl. *I/O*). Kada se jednom podese, te funkcije često nije potrebno više imati na glavnim ekranima.



Slika 20. Pritisak na tipku MENU

Pritiskom na SELECT ulazimo u izbornik programa koji se nalaze na robotu. Na listi se nalaze korisnički USER programi koje smo mi, kao programeri, inicijalizirali i niz sistemskih programa koji su već u robotu te su neophodni za rad robota.



Slika 21. Pritisak na tipku SELECT

S lijeve strane nalazi se ime programa, a s desne komentari koje pišemo mi, programeri, da bismo na prvi pogled znali čemu program služi. Iz slike poviše vidimo da je odabran program BEGINNER s komentarom „Pick and Place“ (Pokupi i ostavi - najčešća aplikacija industrijskih robota).

EDIT nam omogućava prikaz programa i njegovu izmjenu. Program koji je moguće izmijeniti je onaj isti koji je odabran u prethodnom koraku tipkom SELECT.



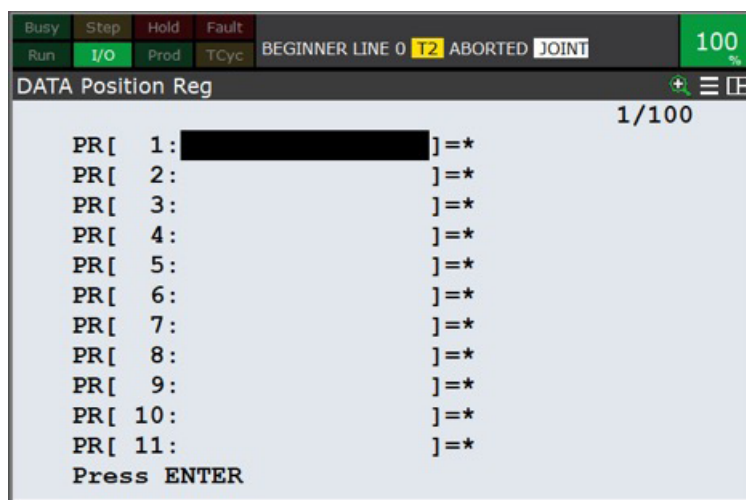
Slika 22. Pritisak na tipku EDIT

Određene pozicije robota i varijable spremamo u registre.

Postoje dvije vrste registara:

- brojevni registri (engl. *Numerical registers*)
- pozicijski registri (engl. *Position registers*)

Pozicijski registri se razlikuju od brojevnih po tome što mogu spremati poziciju robota, a to znači kada naučimo poziciju (engl. *Teach*), u registru se nalazi podatak o svakoj pojedinoj osi robota s pripadajućom rotacijom. U brojevnim registrima moguće je samo spremati brojevni tip podatka.

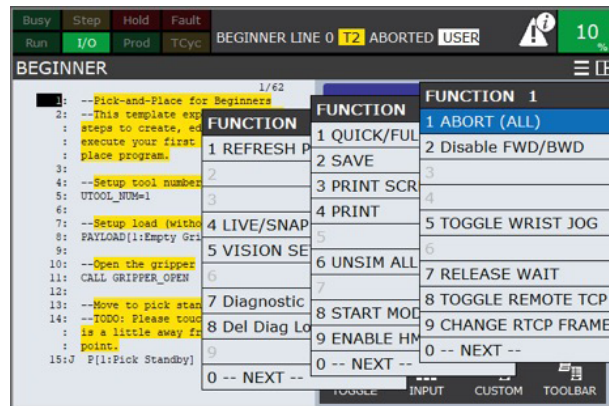


Slika 23. Prikaz pozicijskih registara pritiskom na tipku DATA (na slici nisu spremljeni podaci u registre)

FCTN tipka otvara izbornik funkcija robota.

Neke od važnih funkcija su:

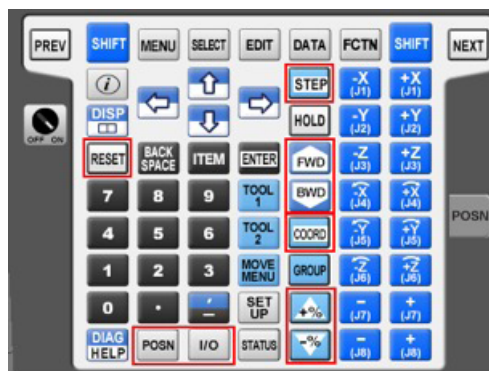
- ABORT - poništavanje svih radnji robota (engl. Reset)
- START MODE - podešavanje na koji način robot starta kada se uključi
- VISION SETUP - u slučaju da je robot spojen na kameru, prikazuju se vizijski registri.



Slika 24. Pritisak na tipku FCTN

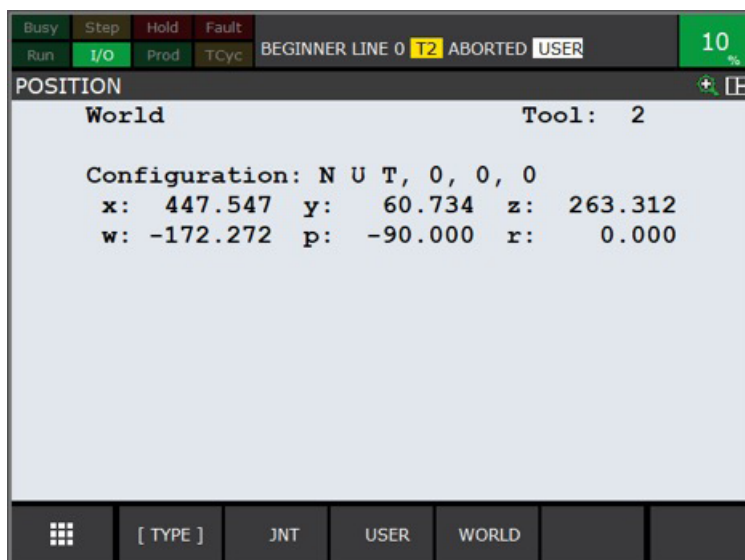
Tipkala koja su iznimno bitna su:

- RESET - resetira sva stanja greške na robotu
- POSN - prikaz trenutne pozicije robota
- I/O - prikaz svih ulazno/izlaznih jedinica
- STEP - izvođenje robotskog programa korak po korak
- FWD - prelazak u idući korak robotskog programa
- BWD - vraćanje na prethodni korak robotskog programa
- COORD - promjena koordinatnog sustava
- + % - povećavanje brzine robota (maksimalno 100 %)
- - % - smanjivanje brzine robota (minimalno 0 % - robot se ne giba).



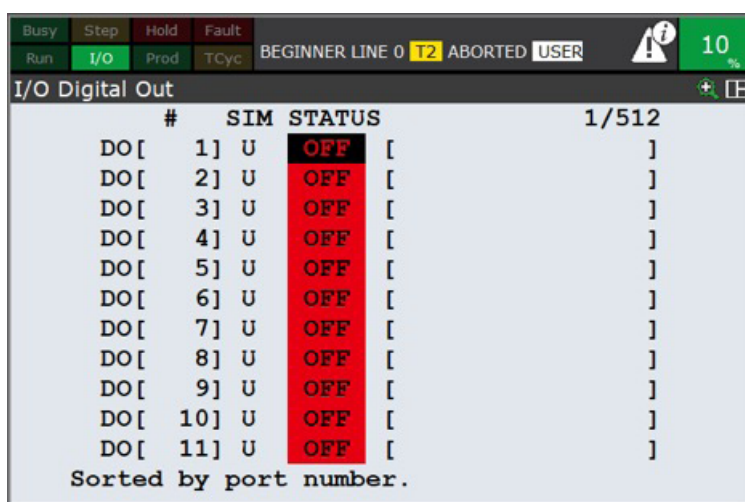
Slika 25. Ostala tipkala

Pritiskom na tipkalo POS prikazuje se trenutna pozicija robota u WORLD koordinatnom sustavu bez obzira koji smo koordinatni sustav odabrali (npr. JOINT).



Slika 26. Prikaz trenutne pozicije robota

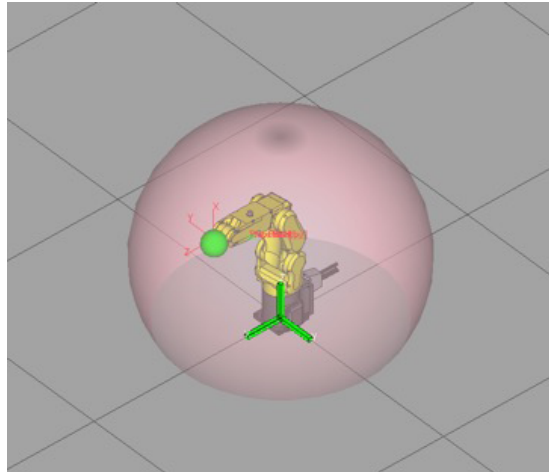
Robot ima ulazno/izlazne jedinice (I/O). Robot može slati digitalne signale drugim uređajima, npr. strojevima u procesu (engl. *Digital output*) i može primati signale s drugih uređaja (Digital input). Pritiskom na tipku I/O može se vidjeti trenutno stanje signala, jesu li signali uključeni ili isključeni.



Slika 27. Na slici vidimo da su svi digitalni izlazi u isključenom stanju (OFF)

## 5. Pokretanje robota u ručnom radu

Robot možemo pokrenuti ručnom upravljačkom jedinicom i kretati se u prostoru. U našem slučaju FANUC ER-4iA robot ima šest osi i granica kretanja je njegov doseg od 550 mm. Kada se približimo granici, robot stane s kretanjem i signalizira stanje alarmom.

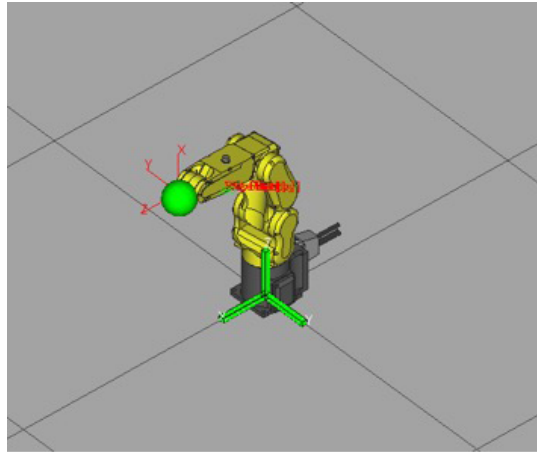


Slika 28. Prikaz limita kretanja robota

Kada robot pokrećemo ručnom upravljačkom jedinicom, on nema svoju maksimalnu brzinu iz razloga što bi to bilo opasno za programera ili operatera. Pokretanje u ručnom radu nam zapravo služi da robot dovedemo u određenu poziciju koja nam je potrebna za aplikaciju, npr. pozicija poviše radnog objekta koji moramo prebaciti na drugu stranu procesa. Kada je robot u poziciji od interesa, koordinate te točke možemo snimiti tako da robot u automatskom radu može doći ponovno u tu istu poziciju.

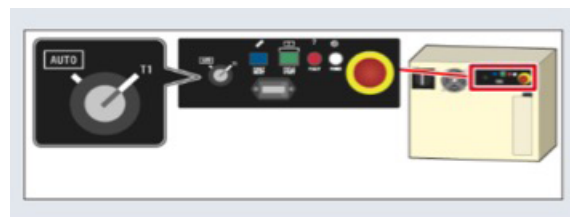
### 5.1. Pokretanje robota u „JOG“ načinu rada

U ovom poglavlju opisat ćemo kako ručno pokretati robot ručnom upravljačkom jedinicom. Robot ćemo pokretati u WORLD i JOINT koordinatnom sustavu. JOINT će pomicati pojedine osi robota, dok ćemo se u WORLD koordinatnom sustavu pomicati u kartezijevom koordinatnom sustavu u odnosu na bazu ili alat robota kao što je prikazano na slici (zeleni koordinatni sustav u bazi robota, crveni koordinatni sustav se nalazi u alatu robota).



Slika 29. Prikaz kartezijevog (X, Y, Z) koordinatnog sustava u bazi i alatu robota

U prvom koraku potrebno je na glavnoj jedinici robota preklopku za odabir načina rada prebaciti u položaj T1.



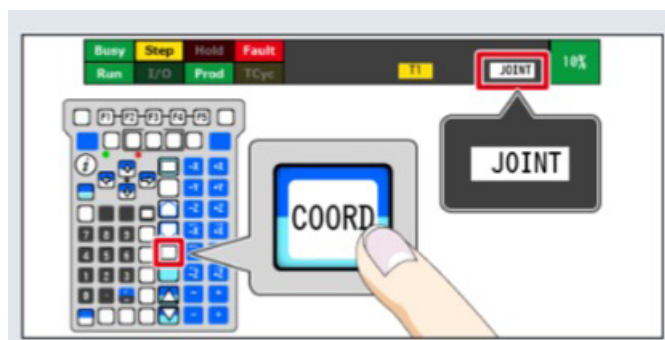
Slika 30. Preklopka za odabir načina rada je u poziciji T1

Zatim je potrebno uključiti ručnu upravljačku jedinicu, postaviti preklopku na vrhu jedinice u poziciju ON.



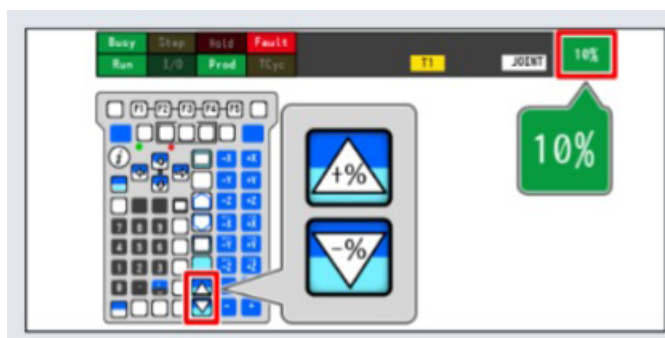
Slika 31. Prikaz preklopke za uključivanje ručne upravljačke jedinice

S tipkalom CORD odaberemo koordinatni sustav robota. U prvom slučaju odabiremo JOINT, što možemo provjeriti na statusnoj traci.



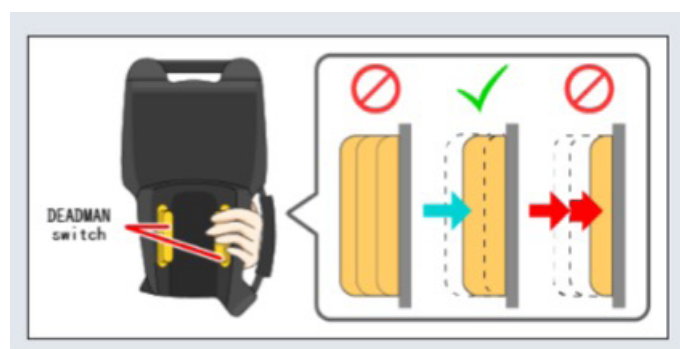
Slika 32. Prikaz odabira koordinatnog sustava tipkom CORD

Kada prvi put pokrećemo robot, preporuka je da krenemo s jako malom brzinom. Tipkama + % i - % podesimo brzinu na 10 % maksimalne brzine u ručnom radu.



Slika 33. Podešavanje brzine robota

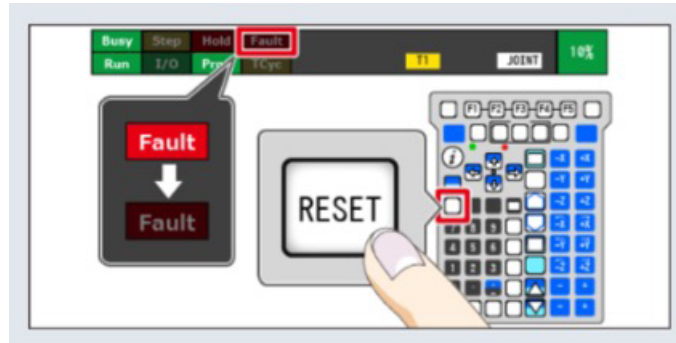
Na poleđini ručne upravljačke jedinice imamo sigurnosni prekidač (engl. *Deadman switch*) koji ne smije biti do kraja pritisnut niti do kraja pušten, već treba biti točno u sredini. U slučaju da trebamo naglo zaustaviti robot, dovoljno je do kraja stisnuti sigurnosni prekidač ili ga pustiti. Određene osobe u stanju stresa puste ili jače stisnu sigurnosni prekidač te su ovakvom konstrukcijom sigurnosnog prekidača pokrivena obje skupine.



Slika 34. Pravilni način upotrebe sigurnosnog prekidača

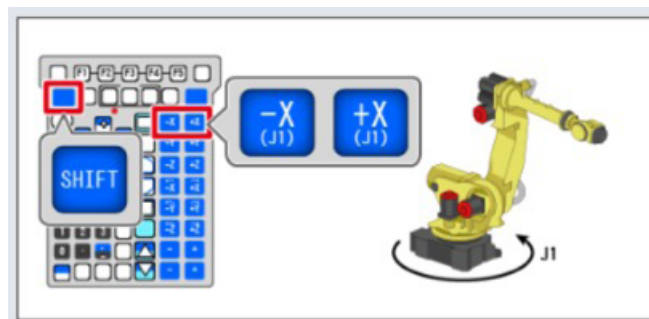


Potrebno je resetirati stanje robota i sve greške u slučaju da one postoje. Dok držimo sigurnosni prekidač u pravilnoj poziciji, pritisnemo tipku RESET. U slučaju da stanje robota ne možemo resetirati, potrebno je pogledati u dokumentaciji robota o čemu se zapravo radi te koje je rješenje.



Slika 35. Pritiskom tipke RESET greška na statusnoj traci bi trebala biti isključena

Dok držimo sigurnosni prekidač, potrebno je držati i tipku SHIFT. Pritiskom tipkala iz skupine ručnog upravljanja (pogledati prethodna poglavlja), robot će se gibati u smjeru koje je označeno na tipkalu. Pošto je odabran JOINT način rada, potrebno je obratiti pozornost na oznake J1 - J6.



Slika 36. Držanjem tipke SHIFT i jedne od tipki iz sekcije ručnog upravljanja, robot će se pokrenuti.

U JOINT modu možemo tipkalicama iz sekcije ručnog upravljanja pomicati svaku os zasebno od J1 do J6.

Robot možemo pomicati i u WORD koordinatnom sustavu. Pritiskom tipke CORD mijenjamo koordinatni sustav dok na statusnoj traci ne vidimo da se robot nalazi u WORD koordinatnom sustavu.

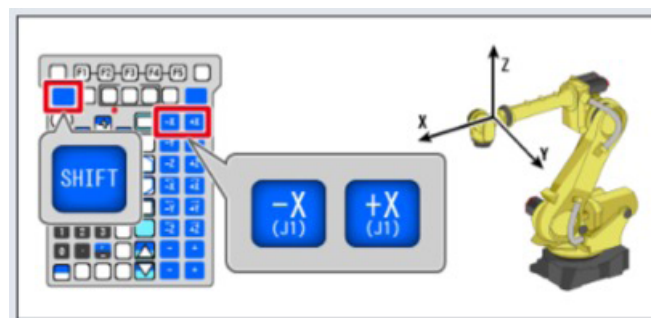


Slika 37. Promjena koordinatnog sustava na robotu

Moramo ponoviti par istih koraka kao i u JOINT koordinatnom sustavu:

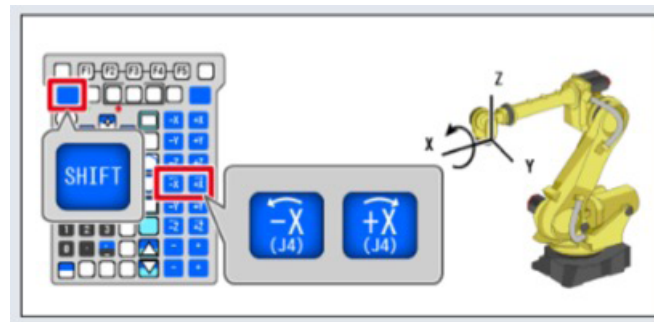
1. Preklopka za odabir rada na glavnoj jedinici mora biti u poziciji T1.
2. Potrebno je uključiti ručnu upravljačku jedinicu.
3. Potrebno je smanjiti brzinu robota.
4. Potrebno je osigurati da sigurnosni prekidač držimo u srednjoj poziciji.
5. Potrebno je resetirati stanje robota i greške.

Kada su svi uvjeti zadovoljeni, pritiskom tipkala iz sekcije ručnog upravljanja robot će se pokrenuti, ali ovog puta se giba u kartezijevom koordinatnom sustavu. Pošto je odabran WORLD način rada, potrebno je obratiti pozornost na oznake +X, -X, +Y, -Y, +Z, -Z te varijacije koje su označene strelicom nagore.



Slika 38. Pomicanje robota u WORLD koordinatnom sustavu

Pošto se u WORLD koordinatnom sustavu gibamo po osima X, Y i Z, možemo se rotirati oko istih (pogledati prethodna poglavlja). Tipkalo sa strelicom poviše osi označava gibanje robota oko pojedine osi. Na slici je prikazano kako se možemo rotirati oko osi X.



Slika 39. Gibanje robota oko X osi

## 5.2. Limiti u odnosu na automatski rad

Na glavnoj jedinici robota možemo odabrati dva načina rada AUTO i T1. Određeni modeli robota imaju dodatni T2 način rada.

U slučaju da je robot u automatskom radu, giba se maksimalnom brzinom. Da bi se pokrenuo, svi sigurnosni signali moraju biti zadovoljeni. Jedan od sigurnosnih signala mogu biti vrata iza kojih robot radi. Ta ista vrata moraju biti zatvorena da bi se robot pokrenuo u automatskom radu. Nismo u mogućnosti pokretati robot ručnom upravljačkom jedinicom. Ovaj način rada se koristi kada je program robota inicijaliziran i testiran te robot može bez pomoći operatera obavljati programirane akcije.

Kod T1 načina rada, brzina robota ograničena je na 250 mm/s. Robot se može pokrenuti iako svi sigurnosni signali nisu zadovoljeni. U mogućnosti smo pomicati robot ručnom upravljačkom jedinicom. Ovaj način rada se koristi pri programiranju robota i testiranju napisanog programa.

T2 način rada je ekvivalentan T1 načinu rada. Jedina je razlika u tome što nemamo limit brzine, već se robot giba maksimalnom brzinom. Iz razloga što svi sigurnosni signali ne moraju biti zadovoljeni kada radimo u T2 načinu rada, potreban je poseban oprez. Ovaj način rada koristi se za finalno testiranje programa robota pri najvećim brzinama.

## 6. Središnja točka alata: Automatska detekcija i unos

Središnja točka alata (engl. Tool Center Point – TCP) je ključni koncept kod industrijskih robota. U prijašnjim poglavljima naučili smo kako se gibati robotom po prostoru, naučili smo koji koordinatni sustavi postoje i spomenuli WORLD koordinatni sustav. Nedostaje nam odgovor na pitanje koja se točka robota giba u prostoru u slučaju da smo odabrali WORLD koordinatni sustav.

Središnja točka alata (TCP) je precizno definirana točka na alatu koji je pričvršćen za robot. U osnovi, to je točka koju robot prati ili koristi kao reference prilikom izvođenja zadataka. Na primjer, ako robot koristi olovku za crtanje, TCP bi bio vrh olovke, tj. točka koja dodiruje papir prilikom crtanja. Realni primjer bio bi slučaj u kojem je na robot montirana hvataljka koja se koristi za prenošenje objekata. TCP bi bila točka između njegovih prstiju kojima hvata objekt.

Središnja točka alata (engl. Tool Center Point – TCP) je važna iz više razloga i nužna je za ispravan rad s robotom.

Glavni razlozi zbog kojih moramo definirati TCP:

- preciznost
- programiranje
- prilagodba.

### Preciznost

U industrijskim operacijama često je potrebna velika preciznost. Pravilno definiranje TCP-a omogućuje robotu da u svakom trenutku točno zna gdje se njegov alat nalazi, čime mu omogućava precizno pozicioniranje i manipulaciju.

### Programiranje

Kada programiramo robot za određeni zadatak, koristimo TCP kako bismo robot naveli gdje treba ići i što treba raditi. U slučaju da želimo pomaknuti olovku montiranu na robotu od točke A do točke B na papiru, koristimo TCP koji je definiran na vrhu olovke kako bismo definirali točke A i B.

### Prilagodba

Ako na robotu zamijenimo alat, moramo ponovno definirati TCP. Postoji mogućnost da robotu dodamo novi alat jer roboti imaju mogućnost da rade s više alata istodobno. U oba

navedena slučaja potrebno je ponovno definirati TCP kako bi robot znao gdje je središnja točka novih alata. Na primjer, ukoliko zamijenimo olovku škalicama, TCP će se morati prilagoditi da odražava novu funkciju alata.

Središnju točku alata (TCP) možemo podesiti na dva načina:

1. upisivanjem
2. automatskim određivanjem.

## 6.1. Vrste alata robota

Robotski alati su specijalizirane komponente koje se montiraju na robotsku ruku kako bi se omogućila interakcija s objektima ili obavljanje specifičnih zadataka. Vrste alata koji se mogu pričvrstiti na robote su raznovrsni i određeni su primarno funkcijom koju robot treba obavljati. Prilikom dizajniranja robotske aplikacije najvažniji parametri o kojima trebamo voditi računa su:

- doseg robota
- izgled objekata s kojima robot mora raditi.

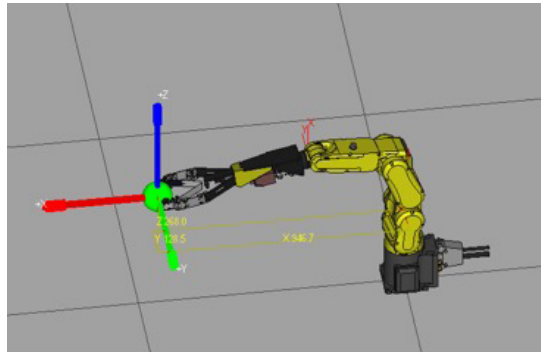
Ova dva parametra direktno će utjecati na izgled samog alata.

Nekoliko primjera vrsta robotskih alata koji se koriste u industriji:

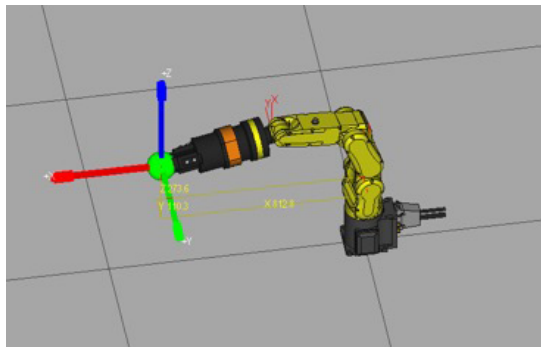
- Hvataljke (*Gripperi*) - koriste se za hvatanje i premještanje objekata. Postoje različite vrste hvataljki, uključujući mehaničke, vakumske, magnetske i adaptivne.
- Zavarivački alati - koriste se u postupcima zavarivanja, poput MIG/MAG ili TIG zavarivanja.
- Alati za bojenje i prskanje - koriste se u industrijskom bojenju ili prskanju premazom.
- Alati za bušenje i rezbarenje omogućuju obradu materijala bušenjem i rezbarenjem.
- Kamere i senzori omogućuju robotima detekciju objekata, prepoznavanje oblika ili obavljanje inspekcijskih zadataka.

U određenim slučajevima potrebno je pričvrstiti servomotor kao alat. Obično se to koristi u situacijama kada je potrebna dodatna rotacija, precizno pozicioniranje ili specifičan pokret koji robot ne može sam po sebi postići.

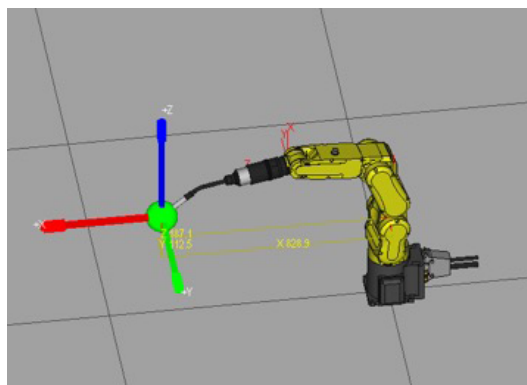
## 6.2. Upisivanje TCP-a



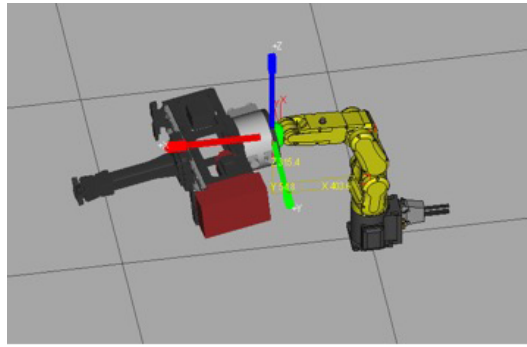
Slika 40. Prikaz hvataljke montirane kao alat robota s prikazom koordinatnog sustava alata



Slika 41. Prikaz hvataljke s tri prsta montirane kao alat robota s prikazom koordinatnog sustava alata



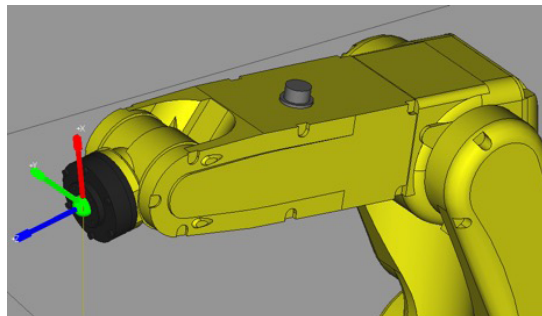
Slika 42. Prikaz pištolja za zavarivanje montiranog kao alat robota s prikazom koordinatnog sustava alata



Slika 43. Prikaz servomotora montiranog kao alat robota s prikazom koordinatnog sustava alata

### 6.3. Koordinatni sustav alata (TOOL)

Spomenuli smo da robot ima JOINT i WORLD koordinatni sustav. Robot posjeduje još jedan koordinatni sustav koji nismo spominjali, a to je koordinatni sustav alata ili (engl. TOOL) koordinatni sustav. Taj koordinatni sustav nalazi se u prirubnici (flanši) robota.

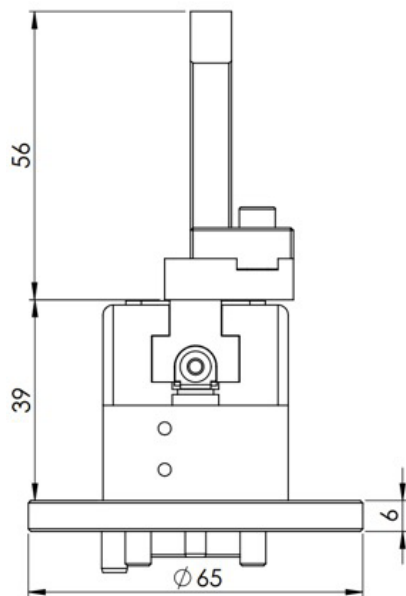


Slika 44. Prikaz ishodišta TOOL koordinatnog sustava s osima X, Y i Z

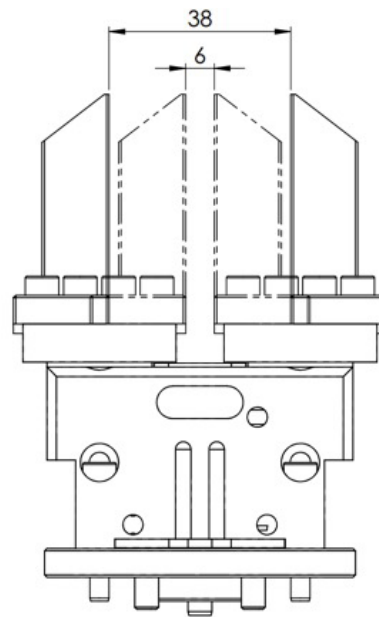
Iz gornje slike možemo vidjeti da Z os TOOL koordinatnog sustava „izlazi“ iz prirubnice robota. To je ujedno i referentni TOOL koordinatni sustav koji se nalazi u početnim postavkama robota. TOOL koordinatni sustav definira odnos između kraja robotove ruke (prirubnice) i funkcionalnog dijela alata koji će biti pričvršćen na robot. Položaj i orijentacija TOOL referentnog koordinatnog sustava ovisi o specifičnom alatu koji je pričvršćen za robota i kako je alat postavljen u odnosu na robotovu ruku. Na primjer, zamislite da vaša ruka predstavlja robotovu ruku, a u ruci držite olovku koja predstavlja alat. Ako želite crtati ili pisati nešto, fokusirate se na vrh olovke, a ne na svoju ruku. Na sličan način se robot fokusira na kraj alata kako bi precizno obavljao željeni zadatak. Da bi se robot mogao fokusirati na alat u prethodnom primjeru, potrebno je podesiti ishodište referentnog TOOL koordinatnog sustava u vrh olovke (točka  $X=0$ ,  $Y=0$ ,  $Z=0$ ). Definiranjem ishodišta robot može precizno manipulirati alatom (u ovom slučaju olovkom) jer zna točnu poziciju i orijentaciju alata u svakom trenutku.

## 6.4. Upisivanje TCP-a

Središnju točku alata (TCP) možemo upisati kao parametar robota i to je ujedno i najbolji način definiranja TCP-a. Prilikom konstrukcije robotske aplikacije prvi je korak sama konstrukcija alata robota. Inženjeri strojarstva koji rade na aplikaciji odabiru tip i konstruiraju alat robota. Programerima robotske aplikacije potrebno je dostaviti nacrt alata s mjerama iz kojih se može definirati središnja točka alata. Kao primjer uzet ćemo aplikaciju za premeštanje objekata koja kao alat koristi hvataljku.



Slika 45. Bočni prikaz hvataljke s mjerama



Slika 46. Prednji prikaz hvataljke s mjerama

Iz bočnog prikaza vidimo da je visina hvataljke 101 mm. Donji dio hvataljke montiran je na prirubnicu odnosno 6. os robota. U poglavlju ranije objasnili smo pojam TOOL koordinatnog sustava. Robotu na određeni način moramo reći da je montiran alat na prirubnicu dužine 101 mm. Na slici 44 možemo vidjeti da u TOOL koordinatnom sustavu Z os izlazi iz prirubnice. Da bismo mogli precizno manipulirati predmetima koristeći alat, potrebno je ishodište TOOL koordinatnog sustava pomaknuti za 101 mm.

Postoji više načina kako podesiti pomak od 101 mm po osi Z, a najlakši je upis kroz parametre robota vezanih za alat.

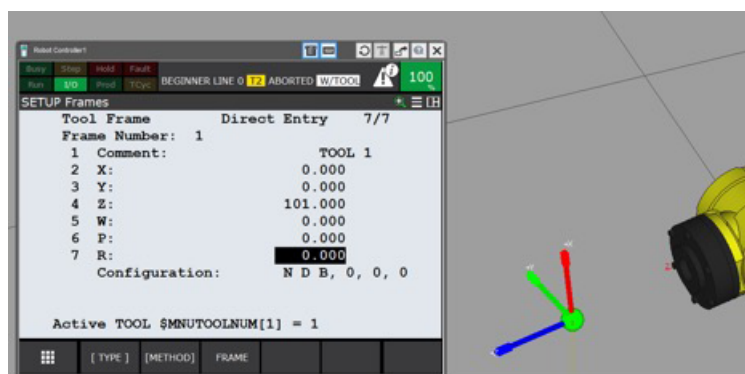


Na ručnoj upravljačkoj jedinici pritisnemo tipku MENU, zatim iz izbornika odaberemo SETUP pa FRAMES (Okviri). Otvorit će nam se izbornik s okvirima alata (TOOL FRAME). To su zapravo koordinatni sustavi alata robota i robot ih u osnovnom paketu može imati 10.

Tool Frame	X	Y	Z	Comment	Direct Entry	7/10
1	0.0	0.0	101.0	[TOOL 1		]
2	0.0	0.0	0.0	[Eoat2		]
3	0.0	0.0	415.0	[3jaw_Gripper		]
4	-84.5	0.0	439.9	[350GC_30L_Ga>		
5	0.0	0.0	0.0	[ultrasonic_k>		
6	0.0	0.0	0.0	[Seal_Gun01		]
7	0.0	0.0	0.0	[STUD_GUN_BRK>		
8	8.0	0.0	550.0	[Bin_Picking_>		
9	0.0	0.0	0.0	[Servo_Hand_2>		
10	0.0	0.0	0.0	[Eoat10		]

Slika 47. Prikaz izbornika s okvirima alata

Odaberemo okvir alata broj 1 i pritisnemo DETAILS izbornik na dodirnom zaslonu (engl. *touch screen*). Otvorit će nam se detalji prvog alata koje možemo podesiti. Naš je alat jednostavan i montiran je direktno na pribudnicu robota. Dovoljno je pomaknuti koordinatni sustav po osi Z TOOL koordinatnog sustava. Pritiskom tipke ENTER na ručnoj upravljačkoj jedinici, podesili smo hvataljku kao alat broj 1 u robotu.



Slika 48. Pomicanje referentnog TOOL koordinatnog sustava

Na slici 48 možemo vidjeti da smo pomakli referentni TOOL koordinatni sustav za 101 mm od pribudnice robota i definirali novo ishodište. Kada montiramo hvataljku na robot, moći ćemo precizno manipulirati hvataljkom te krenuti s programiranjem aplikacije.

## 6.5. Automatsko određivanje TCP-a

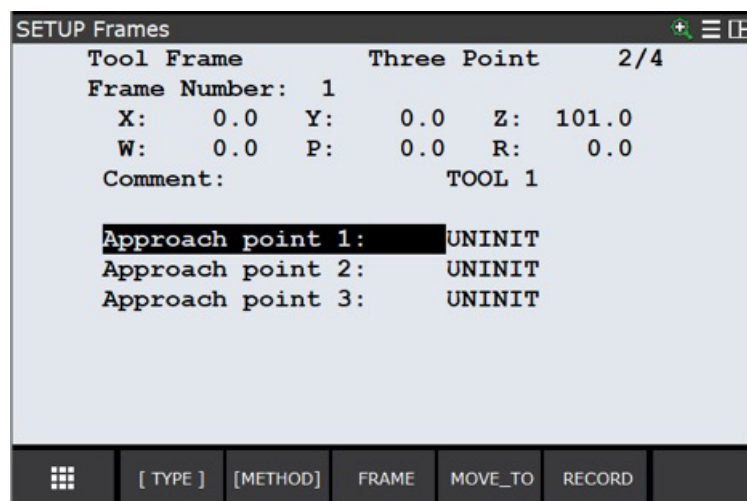
Najbolja opcija za određivanje središnje točke alata (TCP) je direktnim unosom, međutim nekada to nije moguće. Hvataljka ili bilo koji drugi alat montiran na robotu ne odgovara nacrtu ili se jednostavno alat pomaknuo s prirubnice robota u određenom smjeru. Industrijski roboti imaju nekoliko metoda s kojima se TCP može odrediti automatski.

Kod FANUC robota te metode su:

- metoda tri točke
- metoda šest točaka (XZ)
- metoda šest točaka (XY)
- dvije točke i Z
- četiri točke.

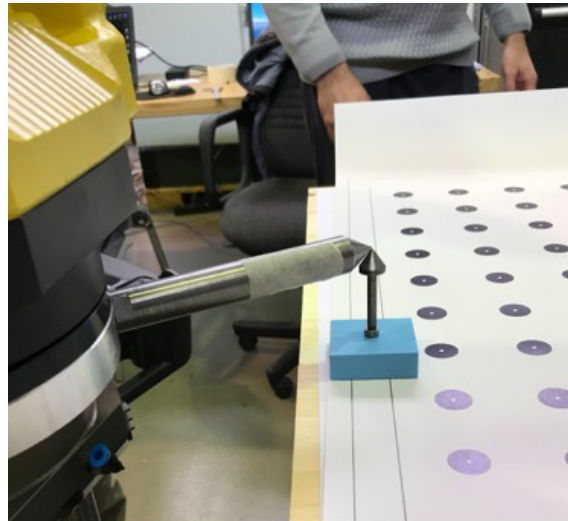
Najčešće korištena metoda je metoda tri točke. Ostale metode mogu imati nešto bolju preciznost u ovisnosti alata koji je montiran na robotu. U većini slučajeva zadovoljavajuću preciznost možemo dobiti metodom tri točke.

Da bismo pristupili automatskom određivanju alata, potrebno je na ručnoj upravljačkoj jedinici pritisnuti tipku MENU, zatim odabrati SETUP i nakon toga izbornik FRAMES. Kao osnovni prozor trebao bi se otvoriti TOOL okvir. Ako to nije slučaj, potrebno je odabrati opciju OTHER na dodirnom zaslonu te odabrati TOOL okvir. Dobit ćemo prikaz TOOL okvira (engl. *frameova*) na ekranu (10 lokacija). Potrebno je izabrati okvir koji želimo automatski podesiti pritiskom DETAILS izbornika na zaslonu. U izborniku DETAILS vidimo trenutna podešavanja koja su vezana za odabrani alat. Odabirom izbornika METHOD na zaslonu, otvara se novi izbornik s listom metoda za određivanje TCP-a.



Slika 49. Izbornik za automatsko određivanje TCP-a pomoću metode tri točke

Za automatsko podešavanje potreban nam je šiljak u prostoru. Šiljak predstavlja kalibracijski objekt, odnosno referentnu točku kalibracije. Cilj automatskog podešavanja TCP-a je ručno upravljati robot (JOG mode) na način da sa željenom središnjom točkom alata (TCP-om) dođemo na vrh šiljka s tri različite orijentacije (prikazano na slici 50). Nakon što smo snimili tri točke na šiljku (pritiskom tipke RECORD), robot će algoritmom odrediti automatski središnju točku alata i spremi je u podešavanje.



Slika 50. Automatsko određivanje središnje točke alata (TCP)

Automatski određenu središnju točku alata potrebno je provjeriti, a najbolji način za to je vizualna provjera. Potrebno je ručno pokrenuti robot u TOOL koordinatnom sustavu. Ovaj sustav ima različite koordinate od WORLD koordinatnog sustava. U nekim slučajevima mogu biti iste, ali to ponajviše ovisi o alatu koji je montiran na robot i kako se podesio TOOL koordinatni sustav. Pokušamo s alatom robota doći do referentne točke kalibracije, u našem primjeru vrha šiljka. Nadalje, kada smo došli u vrh šiljka, pokušavamo se rotirati robotom oko osi X, Y i Z. U slučaju da je kalibracija odrađena s visokim stupnjem točnosti, alat robota će se rotirati oko kalibracijske točke bez da mijenja poziciju alata u odnosu na referentnu točku.

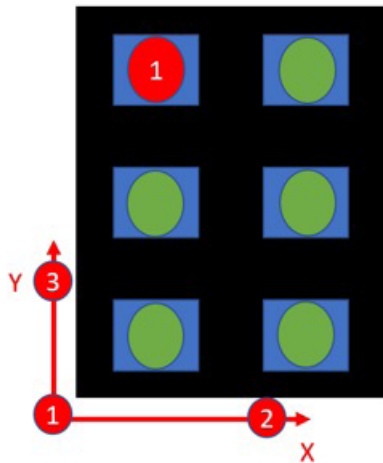
## 7. Korisnički koordinatni sustav radnog objekta (engl. *User Frame*)

### 7.1. Svrha korisničkog koordinatnog sustava

U prethodnim poglavljima prošli smo kroz osnovne koordinatne sustave robota WORLD i JOINT. Industrijski roboti imaju mogućnost definiranja korisničkih koordinatnih sustava. Kao što smo vidjeli u 6. poglavlju na slici 29, centar koordinatnog sustava ( $X=0$ ,  $Y=0$ ,  $Z=0$ ) nalazi se u bazi robota. Sve ostale točke u kartezijevom koordinatnom sustavu imaju vrijednost u odnosu na to ishodište. Kod većine aplikacija potrebno je podesiti korisnički sustav iz više razloga.

Prvi razlog je što baza robota može biti montirana u jednom dijelu prostora, dok je paleta s koje je zadatak preuzeti objekte u drugom dijelu prostora. Jako je mala vjerojatnost da su podloge jednake. U slučaju da definiramo koordinatni sustav na rubu palete (kod FANUC robota koristi se termin okvir, engl. FRAME), uzeli smo u obzir razliku u podlozi. Na taj način definiran koordinatni sustav omogućuje robotu da se giba okomito na paletu s velikom točnošću.

Drugi razlog može biti činjenica da na paleti često imamo više objekata koje je potrebno prebaciti. Ako je ishodište koordinatnog sustava na rubu palete, jednostavno je proračunati koliko se trebamo pomaknuti od ruba da bismo došli do prvog objekta i koliki je razmak do drugog objekta. Taj koordinatni sustav nazivamo korisnički koordinatni sustav radnog objekta.



Slika 51. Koordinatni sustav robota možemo definirati na rubu palete – korisnički koordinatni sustav radnog objekta

## 7.2. Definiranje korisničkog koordinatnog sustava radnog objekta

Kod FANUC robota možemo definirati devet korisničkih koordinatnih sustava. U svakom trenutku korisnički koordinatni sustav br. 0 ima vrijednost WORLD koordinatnog sustava.

Da bismo došli do izbornika korisničkih koordinatnih sustava, potrebno je na ručnoj upravljačkoj jedinici pritisnuti tipku MENU, zatim u izborniku odabrati SETUP i nakon toga FRAMES. Otvorit će se izbornik sustava koji može biti:

- koordinatni sustav alata (engl. *Tool Frame*)
- koordinatni sustav ručnog pokretanja (engl. *Jog Frame*)
- koordinatni sustav radnog objekta (engl. *User/RTCP*)
- koordinatni sustav ćelije (engl. *Cell Frame*)
- koordinatni sustav poda ćelije (engl. *Cell Floor*).

Koordinatni sustav alata obradili smo već u poglavlju (7.3.). Ostali sustavi koji se spominju nadilaze osnovna podešavanja robota i koriste se za kompleksnije operacije. Industrijski roboti mogu biti montirani za strop ili na zid, stoga je potrebno imati veliku fleksibilnost kod odabira koordinatnih sustava da bi se robot pomicao u prostoru s velikom točnošću.

User/RTCP	X	Y	Z	Comment
1	0.0	0.0	0.0	[UFrame1]
2	0.0	0.0	0.0	[UFrame2]
3	0.0	0.0	0.0	[UFrame3]
4	0.0	0.0	0.0	[UFrame4]
5	0.0	0.0	0.0	[UFrame5]
6	0.0	0.0	0.0	[UFrame6]
7	0.0	0.0	0.0	[UFrame7]
8	0.0	0.0	0.0	[UFrame8]
9	0.0	0.0	0.0	[UFrame9]

Active UFRAME/RTCP \$MNUFRAMENUM[1] = 0

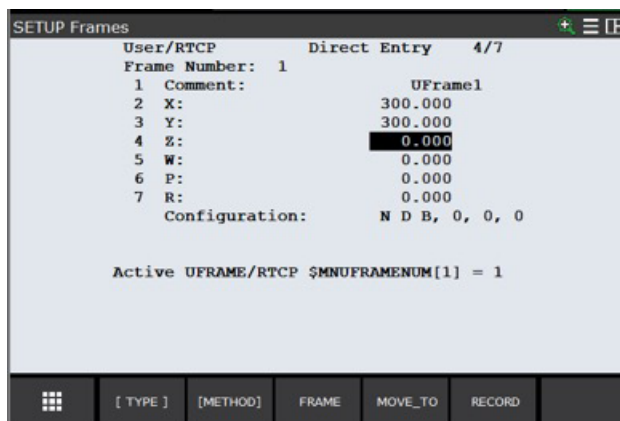
Slika 52. Prikaz različitih korisničkih koordinatnih sustava

Korisnički sustav se može podesiti na dva načina, kao i okvir alata:

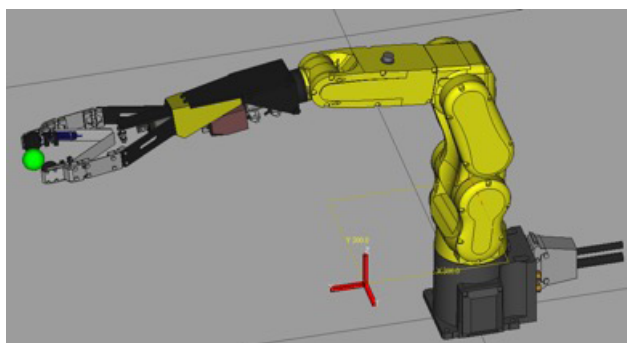
1. upisivanjem
2. automatskim određivanjem.

Kod sustava alata spomenuli smo da težimo upisu koordinata prema nacrtu, a automatski određujemo alat samo ako nemamo drugu opciju.

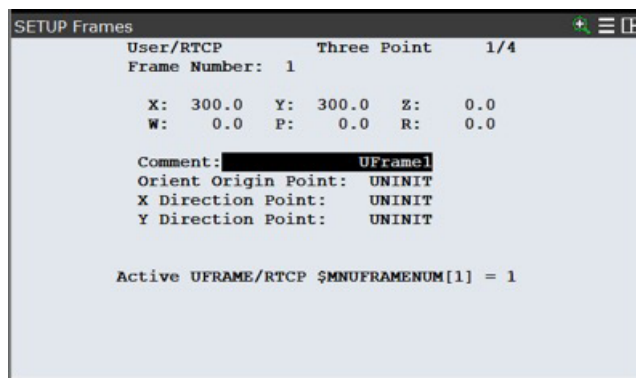
Kod korisničkog koordinatnog sustava je upravo suprotno. Iako imamo nacrt palete ili nekog drugog objekta s kojim robot radi, taj isti objekt je montiran u industrijskom okruženju. Podloga može biti viša nego što je predviđeno ili može biti pod određenim kutom. Stoga možemo, slično kao i kod okvira alata, koristiti oštri alat s kojim možemo definirati prostor pomoću tri točke.



Slika 53. Direktan unos korisničkog koordinatnog sustava radnog objekta



Slika 54. Direktnim upisom korisničkog sustava ( $X=300$ ,  $Y=300$ ,  $Z=0$ ) novi koordinatni sustav pomaknut je za navedene vrijednosti od WORLD koordinatnog sustava



Slika 55. Određivanje korisničkog koordinatnog sustava pomoću tri točke

Kod metode određivanja korisničkog koordinatnog sustava pomoću tri točke potrebno je s oštrim alatom doći u tri točke:

1. točka orijentacije (engl. *Orient Origin Point*)
2. točka u X smjeru (engl. *X Direction Point*)
3. točka u Y smjeru (engl. *Y Direction Point*)

To su točke definirane u prikazu na slici 54. Nakon toga će robot automatski odrediti X, Y, Z koordinate novog korisničkog koordinatnog sustava. Bitno je napomenuti da će se automatski odrediti i rotacija korisničkog sustava W, P i R koordinate. S tim koordinatama smo uzeli u obzir zakrivljenost prostora i ostale nepravilnosti. Rezultat ovog podešavanja je mogućnost robota da se giba duž cijele palete kao da je ona montirana na savršeno ravnoj podlozi.

## 8. Programiranje industrijskih robota

### 8.1. Načini programiranja FANUC robota

Naučili smo na koji način ručno pokretati robot u prostoru koristeći ručnu upravljačku jedinicu, međutim krajnji cilj je da robot samostalno obavlja određenu funkciju bez programa ili operatera. Konkretno, FANUC robote možemo programirati na dva načina:

1. koristeći TP
2. koristeći KAREL.

Za TP možemo reći da je podskup KAREL jezika, odnosno njegova pojednostavljena verzija prilagođena programerima koji za programiranje koriste ručnu upravljačku jedinicu. Na ručnoj upravljačkoj jedinici možemo inicijalizirati novi program, editirati postojeći i spremiti promjene.



Slika 56. Primjer TP programa koji možemo otvoriti na zaslonu ručne upravljačke jedinice

KAREL je programski jezik FANUC robota koji je baziran na višem programskom jeziku PASCAL. KAREL ima strukturu i konvencije kao većina viših programskih jezika, kao i funkcije koje su posebno razvijene za robotske aplikacije.

Funkcije KAREL programskog jezika:

- jednostavni i strukturirani tipovi podataka
- aritmetički, relacijski i Booleovi operatori
- kontrolne strukture za petlje i selekcije
- uvjetno izvršavanje
- procedure i funkcijske rutine
- ulazno/Izlazne (UI) operacije.



```
PROGRAM reference
VAR arg : INTEGER
ROUTINE test(param : INTEGER)
BEGIN
  param = param * 3
  WRITE ('value of param:', param, CR)
END test
BEGIN
  arg = 5
  test((arg))    --arg passed to param by value
  WRITE('value of arg:', arg, CR)
  test(arg)     --arg passed to param by reference
  WRITE('value of arg:', arg, CR)
END reference
```

Slika 57. Izgled KAREL programskog jezika

Svaka od navedenih opcija za programiranje ima svojih prednosti i mana. U slučaju da koristimo TP za programiranje, programiramo jednostavnim programskim jezikom koji odmah možemo testirati na robotu. Mana je što cijeli program robota u principu pišemo i testiramo s ručnom upravljačkom jedinicom, što uključuje dosta tipkanja i promjene izbornika. Povrh toga, ograničeni smo funkcionalnostima, KAREL ima puno više opcija. (Postoji opcija da se program piše u tekst editoru, ali svejedno moramo snimiti točke programa.)

KAREL je dosta kompleksniji programski jezik i njegova prednost je što ima puno više opcija u odnosu na TP. Mana KARELA je upravo kompleksnost samog jezika jer je potrebno dosta specifičnog znanja o FANUC robotima. Jako je teško pisati program direktno na ručnoj upravljačkoj jedinici pa se često razvija na računalu u bilo kojem tekst editoru te se kasnije prebaci na robot. Dosta je teže testirati sam program kada je prebačen na robot. U ovom priručniku usredotočit ćemo se na TP i pisanje programa koristeći ručnu upravljačku jedinicu iz razloga što je to dosta lakši način za programiranja za početnike.

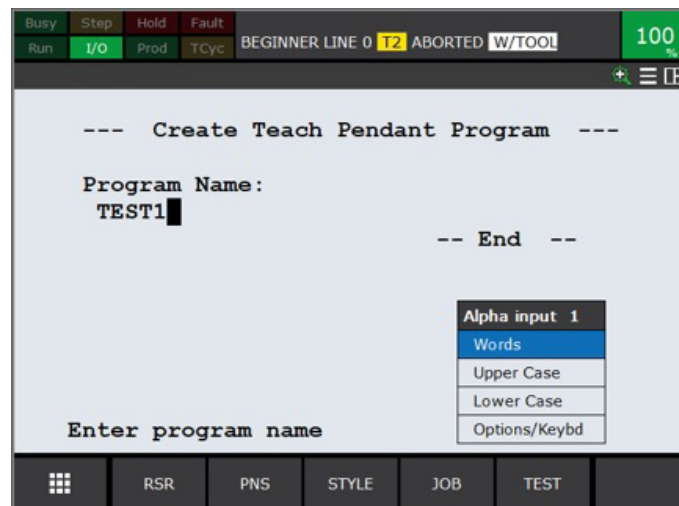
## 8.2. Pisanje osnovnih programa

Koristeći ručnu upravljačku jedinicu možemo krenuti s pisanjem programa na dva načina:

1. inicijalizirati novi program (nema dodanih linija koda)
2. inicijalizirati novi predložak (dodane su često korištene linije koda).

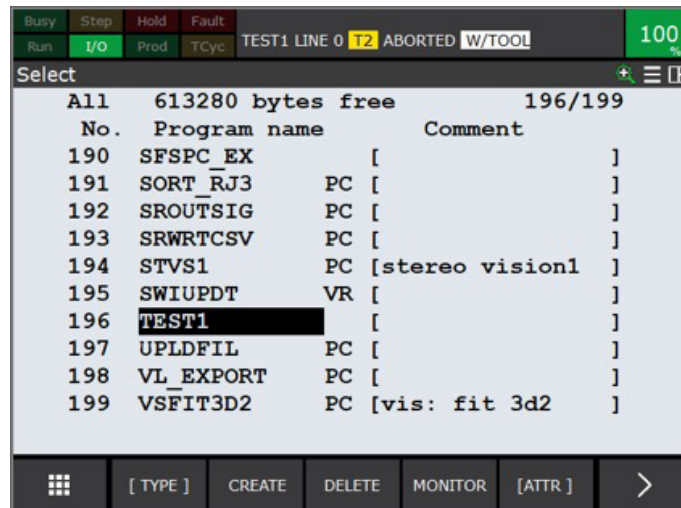
Inicijalizacija novog programa se radi na način da pritisnemo tipku SELECT te nakon toga na zaslonu odaberemo CREATE izbornik.

Otvorit će se izbornik za upis imena programa koji je prikazan na slici 58.



Slika 58. Prikaz izbornika za upis imena programa

Pritiskom na tipku ENTER te odabirom END na zaslonu, inicijalizacija programa je završena. Nakon što smo inicijalizirali program s imenom koje smo odabrali, taj isti program možemo naći u listi programa koji se nalaze na robotu. Listu programa možemo otvoriti pritiskom na tipku SELECT.



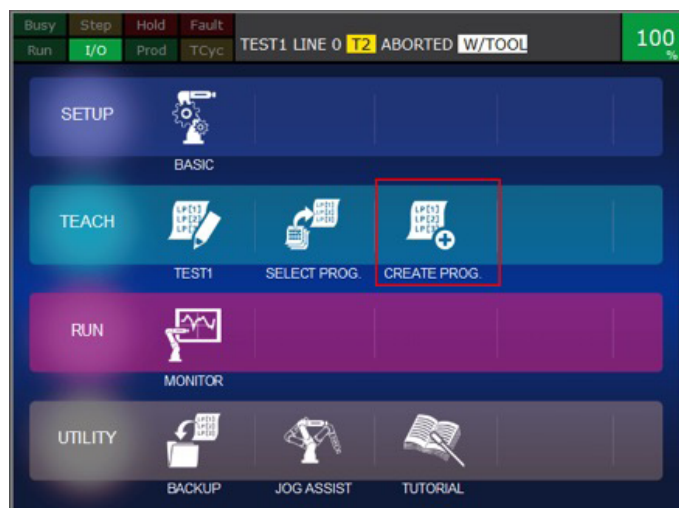
Slika 59. Prikaz programa nakon pritiska SELECT tipke na ručnoj upravljačkoj jedinici

Odabirom programa s liste i pritiskom tipke EDIT možemo vidjeti kako program izgleda. Pošto smo inicijalizirali novi program u izborniku, vidimo samo jednu komandu, a to je END komanda koja označava kraj programa. Imamo „prazan list“ te možemo dodati bilo koju komadu koja odgovara aplikaciji koju je potrebno programirati.



Slika 60. Inicijalizacija novog programa - na slici je prikazano da novi program nema niti jednu komandu osim END (što označava kraj programa)

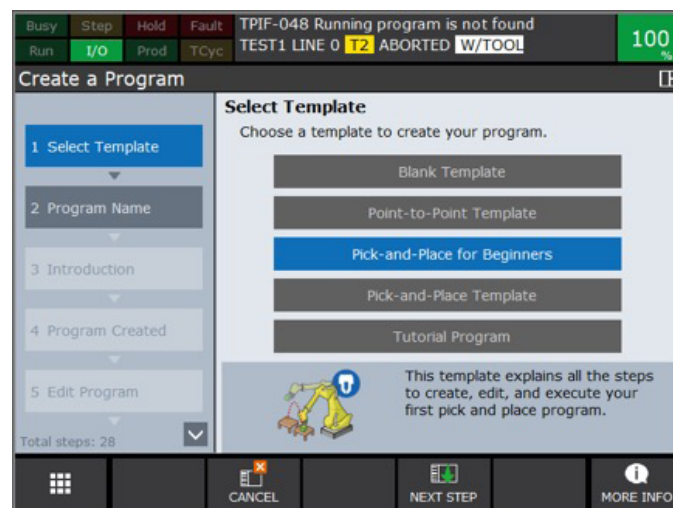
Ponekad je lakše krenuti s predloškom programa. Većina naredbi koje dajemo robotu su zajedničke za većinu aplikacija, npr. odabir alata, odabir koordinatnog sustava, potrebno je doći u neutralnu poziciju itd. Odabirom predloška određene naredbe robota već su dodane te ih je potrebno izmijeniti prema potrebama naše aplikacije. Predložak možemo inicijalizirati na način da se vratimo na glavni ekran ručne upravljačke jedinice te odaberemo CREATE PROGRAM u izborniku, kao što je prikazano na slici 61.



Slika 61. Prikaz glavnog izbornika s potrebnim odabirom za inicijalizaciju novog predloška

Imamo više predložaka za odabir:

1. Prazni predložak (engl. *Blank Template*) - ovaj predložak je ekvivalent inicijalizaciji novog programa bez dodanih linija koda
2. Od točke do točke predložak (engl. *Point to Point Template*) - ovaj predložak sadrži naredbe za pomicanje robota od točke do točke
3. Podigni i spusti predložak za početnike (engl. *Pick-and-Place for Beginners*) - najčešća aplikacija koju industrijski roboti rade je podizanje objekta i spuštanje/premještanje objekata; ovaj predložak sadrži početničke naredbe za tu standardnu aplikaciju
4. Podigni i spusti predložak (engl. *Pick-and-Place*) - ovaj predložak je naprednija verzija prethodnog predloška s dodatnim naprednim naredbama
5. Praktični vodič (engl. *Tutorial Program*) - predložak sadrži osnovne komande za pokretanje robota s komentarima za lakše razumijevanje.



Slika 62. Prikaz izbornika za odabir novog predloška

Kada se odabere Podigni i spusti predložak za početnike, dovoljno je upisati ime programa u sljedećim koracima i program je inicijaliziran. Ako idemo dalje s procedurom, u mogućnosti smo već u ovom koraku naučiti točke u koje robot treba doći u prostoru. Taj dio ćemo objasniti u sljedećim poglavljima.



Slika 63. Prikaz inicijaliziranog predloška Podigni i spusti s dodanim određenim uobičajenim komandama robota za aplikaciju

### 8.3. Pisanje programa za slaganje radnih objekata

Najčešće poslove koje roboti obavljaju u industrijskom okruženju su prebacivanje objekata s jednog mjesta na drugo (pokupi i ostavi - engl. *Pick and Place*).

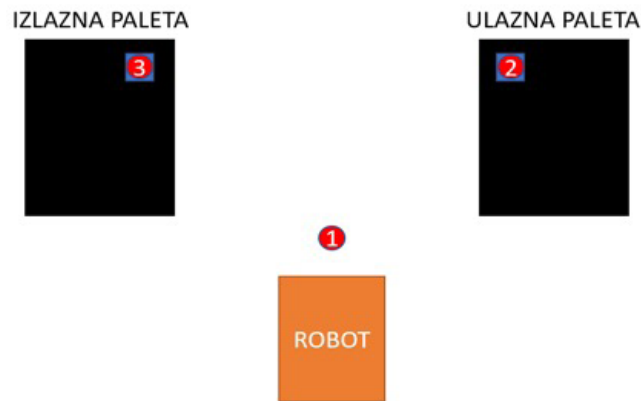
Tipični primjeri su:

- uzimanje objekta s palete i zatim prebacivanje u obradni stroj
- paletiziranje - slaganje objekata na standardnu paletu za potrebe isporuke proizvoda
- depaletiziranje - uzimanje objekata s palete i odlaganje na predviđeno mjesto za potrebe spremanja u skladište
- prebacivanje objekta na drugu obradnu stanicu.

Iako su aplikacije uvelike različite, iz perspektive programera sve se svodi na iste upravljačke naredbe koje moramo dati robotu.

### 8.4. Program za prebacivanje objekata (Podigni i spusti )

U ovom poglavlju objasniti ćemo kako napisati program za aplikaciju koja prebacuje objekte s jednog mjesta na drugo „Podigni i spusti“ (engl. *Pick and Place*). Posao koji bi robot trebao odraditi automatski prikazan je na slici 63. Zadatak robotu bi bio uzeti objekt s ulazne palete i prebaciti ga na izlaznu paletu.



Slika 64. Prikaz tipične aplikacije za prebacivanje objekata

Prije samog programiranja bilo bi dobro vizualizirati potrebne pokrete koje robot treba napraviti da bi uspješno odradio zadatak. Iz slike 64. vidimo da se robot nalazi inicijalno u poziciji br. 1. Iz pozicije br.1 potrebno je doći u poziciju br. 2 na kojoj se nalazi objekt. Potrebno je uzeti objekt s pozicije br. 2 te ga prebaciti na poziciju br. 3. Nakon što je zadatak obavljen, robot bi se trebao vratiti u poziciju broj br. 1 da bi bio spreman za prebacivanje idućeg objekta.

Do sada smo se s robotom gicali prostorom koristeći ručnu upravljačku jedinicu. Ovim programom naučit ćemo kako robotu dati naredbe za pomicanje u automatskom načinu rada.

Kao i kod ručnog upravljanja potrebno je robotu reći koji alat koristi i u kojem koordinatnom sustavu radi.

Da bismo podesili alat koji koristimo u programu, potrebno je koristiti naredbu:

#### **UTOOL\_NUM=1**

Naredba poviše znači da koristimo alat na poziciji broj jedan u cijelom programu.

Da bismo podesili koordinatni sustav koji koristimo u programu, potrebno je koristiti naredbu:

#### **UFRAME\_NUM=0**

Gore navedena naredba znači da koristimo WORLD koordinatni sustav prilikom gibanja robota.

Nakon definicije okruženja u kojem robot radi i s kojim alatom robot obavlja zadatke, krećemo s pisanjem naredbi za pokretanje robota. Naredba za gibanje robota:

### **J P[1: HOME] 100% FINE**

Navedenom naredbom moguće je pomaknuti robota, a sastoji se od četiriju dijelova:

- definicija načina gibanja
- točka do koje robot treba doći
- brzina kojom robot treba doći u točku
- točnost koja je potrebna da bi robot došao u točku.

Detaljno ćemo opisati sve parametre naredbe za gibanje robota u sljedećem poglavlju.

Iz početne pozicije potrebno je robot pomaknuti u poziciju poviše objekta koji moramo podignuti. To možemo učiniti naredbom:

### **J P[2: ABOVE\_PICK\_INPUT] 100% CNT100**

Nakon što smo se našli poviše objekta, potrebno je spustiti se prema objektu od interesa maksimalnom preciznošću te malo sporijom brzinom. To možemo učiniti naredbom:

### **L P[3: PICK\_OBJECT\_INPUT] 400mm/sec FINE**

Naš robot ima alat s kojim podiže objekt. Kada se namjestimo u poziciju za podizanje predmeta, potrebno je zatvoriti hvataljku. Robot može u programu koji trenutno izvodi pozvati drugi program koji zatvara hvataljku. To možemo učiniti naredbom CALL koja će pozvati program za zatvaranje hvataljke (engl. *Gripper*).

### **CALL GRIPPER\_CLOSE**

Ponovno se vratimo u poziciju poviše objekta, ali ovoga puta s objektom u hvataljci. Potrebno je paziti da ne koristimo zglobni (engl. *Joint*) način gibanja, već precizniji linearni.

### **L P[2: ABOVE\_PICK\_INPUT] 400mm/sec FINE**

Za prebacivanje objekta često nije potrebna preciznost i možemo se brže gibati robotom. U slučaju da nema prepreka na putanji, koristimo naredbu:

### **J P[4: ABOVE\_PLACE\_OUTPUT] 100% CNT100**

Da bismo odložili objekt na izlaznu paletu, potrebna nam je ponovno preciznost te koristimo linearnu naredbu s parametrom FINE.

### **L P[5: PLACE\_OBJECT\_OUTPUT] 400mm/sec FINE**

Nakon što smo došli u poziciju na koju trebamo odložiti objekt, potrebno je pozvati CALL naredbu, ali ovoga puta za otvaranje hvataljke jer želimo da objekt ostane na poziciji.

**CALL GRIPPER\_OPEN**

U slučaju da bismo iz ove pozicije koristili zgloбно (engl. *Joint*) gibanje, postoji opasnost da se sudarimo u objekt koji smo odložili. Podižemo se ponovno linearno u točku poviše objekta.

**L P[4: ABOVE\_PLACE\_OUTPUT] 400mm/sec FINE**

Nakon što smo na sigurnoj udaljenosti od objekta, možemo se vratiti u početnu poziciju naredbom:

**J P[1: HOME] 100% FINE**

Cijeli program za prebacivanje objekata:

**1: UTOOL\_NUM=1**

**2: UFRAME\_NUM=0**

**3:**

**4: J P[1: HOME] 100% FINE**

**5: J P[2: ABOVE\_PICK\_INPUT] 100% CNT100**

**6: L P[3: PICK\_OBJECT\_INPUT] 400mm/sec FINE**

**7: CALL GRIPPER\_CLOSE**

**8: L P[2: ABOVE\_PICK\_INPUT] 400mm/sec FINE**

**9:**

**10: J P[4: ABOVE\_PLACE\_OUTPUT] 100% CNT100**

**11: L P[5: PLACE\_OBJECT\_OUTPUT] 400mm/sec FINE**

**12: CALL GRIPPER\_OPEN**

**13: L P[4: ABOVE\_PLACE\_OUTPUT] 400mm/sec FINE**

**14: J P[1: HOME] 100% FINE**

**[End]**

Naredbe robota koje se koriste u programu:

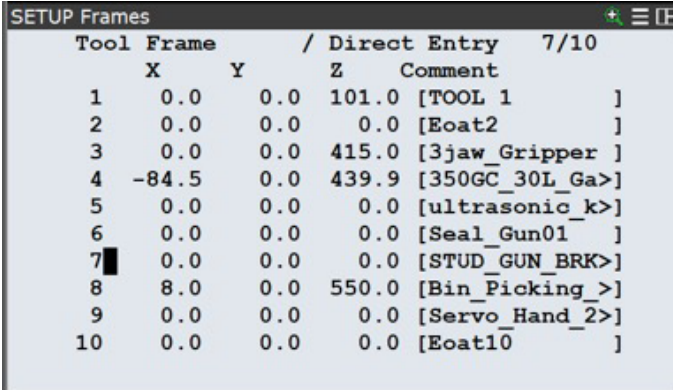
<b>UTOOL_NUM=...</b>	odabir alata koji se koristi u programu
<b>UFRAME_NUM=...</b>	odabir koordinatnog sustava koji se koristi u programu
<b>J P[... ] 100% FINE</b>	zglobno (engl. <i>Joint</i> ) gibanje robota
<b>L P[... ] 4000mm/sec FINE</b>	linearno (engl. <i>Linear</i> ) gibanje robota
<b>CALL &lt;ime_programa&gt;</b>	poziv drugog programa u program koji se trenutno izvodi
<b>[End]</b>	označava kraj izvođenja programa



U poglavljima koja slijede proći ćemo sve osnovne naredbe i detaljno ih objasniti.

## 8.5. Naredbe za odabir alata i koordinatnog sustava

Kao što smo pokazali u prethodnim poglavljima, robotu možemo definirati više alata. U automatskom načinu rada i prilikom učenja točaka potrebno je odabrati koji alat robot za pravo koristi za određenu akciju. U izborniku FRAMES možemo pogledati koji su definirani alati robota.



Tool	Frame	X	Y	Z	Direct Entry	Comment
1		0.0	0.0	101.0		[TOOL 1 ]
2		0.0	0.0	0.0		[Eoat2 ]
3		0.0	0.0	415.0		[3jaw_Gripper ]
4		-84.5	0.0	439.9		[350GC_30L_Ga>]
5		0.0	0.0	0.0		[ultrasonic_k>]
6		0.0	0.0	0.0		[Seal_Gun01 ]
7		0.0	0.0	0.0		[STUD_GUN_BRK>]
8		8.0	0.0	550.0		[Bin_Picking_>]
9		0.0	0.0	0.0		[Servo_Hand_2>]
10		0.0	0.0	0.0		[Eoat10 ]

Slika 65. Razni alati robota, maksimalan broj koji se može definirati je 10

Naredbom **UTOOL\_NUM=1** odabrali smo prvi alat s liste i robot ima definiranu središnju točku alata.

U istom izborniku možemo izabrati User/RTCP okvir. Osnovna postavka je da svi okviri odgovaraju WORLD koordinatnom sustavu, međutim nekada nam je potrebna mogućnost određivanja novog koordinatnog sustava koji se razlikuje od WORLD. Kao primjer možemo uzeti paletu u poglavlju ranije. Možemo u rub palete postaviti ishodište tako da nam je lakše matematički izračunati koliko se treba pomaknuti da bismo bili u mogućnosti podignuti drugi objekt na paleti.

User/RTCP	X	Y	Z	Comment
1	0.0	0.0	0.0	[UFrame1 ]
2	0.0	0.0	0.0	[UFrame2 ]
3	0.0	0.0	0.0	[UFrame3 ]
4	0.0	0.0	0.0	[UFrame4 ]
5	0.0	0.0	0.0	[UFrame5 ]
6	0.0	0.0	0.0	[UFrame6 ]
7	0.0	0.0	0.0	[UFrame7 ]
8	0.0	0.0	0.0	[UFrame8 ]
9	0.0	0.0	0.0	[UFrame9 ]

Active UFRAME/RTCP \$MNUFRAMENUM[1] = 0

Slika 66. razni korisnički sustavi, maksimalan broj koji se može definirati je 9

Naredbom **UFRAME\_NUM=1** odabrali smo prvi korisnički koordinatni sustav s liste i robot ima definirani koordinatni sustav.

## 8.6. Definicija načina gibanja

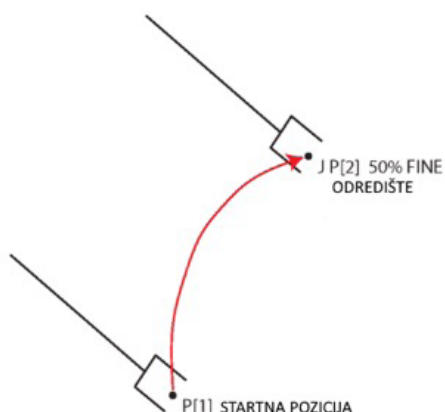
Naredba za gibanje:

**J P[1: HOME] 100% FINE - Označen dio koji definira način gibanja (J)**

Što se tiče definicije načina gibanja, postoje četiri načina:

- zglobni (engl. *Joint*) - oznaka **J**
- linearni (engl. *Linear*) - oznaka **L**
- kružni (engl. *Circular*) - oznaka **C**
- kružni luk (engl. *Circle Arc*) - oznaka **A**.

Kod zglobnog (engl. *Joint*) gibanja sve osi robota se gibaju istodobno, robot se giba maksimalnom brzinom, limit brzine robota ovisi o njegovoj najsporijoj osi. Brzina kojom se robot giba može se odrediti u postotku maksimalne brzine robota. Bitno je napomenuti da ovo gibanje robota nije gibanje po najkraćem putu. Zbog brzine kojom se gibamo i činjenice da se sve osi gibaju po optimalnoj putanji u smislu brzine, ovo gibanje uzrokuje najmanje opterećenje na mehaniku robota.

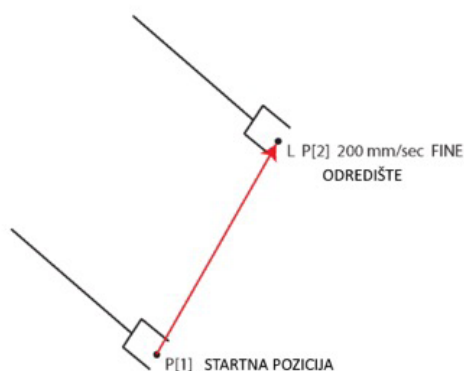


Slika 67. Prikaz zglobnog (engl. Joint) gibanja

Primjeri naredbi:

- **J P[1] 70% CNT100**
- **J P[1] 50% CNT20**
- **J P[1] 100% FINE**

Kod linearnih pokreta brzina robota je sporija. Robot se giba linearno od točke do točke s definiranom središnjom točkom alata. Iako je gibanje robota sporo, jako je precizno. Postizanje te preciznosti uzrokuje veće naprezanje mehaničkih zglobova robota. Iako je ovo često najkraći put od točke do točke, nije najbrži. Prilikom programiranja potrebno je što manje koristiti linearne pokrete, odnosno samo u dijelovima aplikacije gdje je potrebna velika preciznost. Brzina robota se zadaje u vrijednostima mm/sec.



Slika 68. Prikaz linearnog (engl. Linear) gibanja

Primjeri naredbi:

- **L P[1] 100mm/sec CNT100**
- **L P[1] 100mm/sec FINE**

Kružni način gibanja omogućuje robotu da se giba po kružnici. Robot se od svoje početne točke giba do međutočke, zatim do odredišta po kružnici. Limit korištenja ove naredbe je 180 stupnjeva. Uvijek moramo biti sigurni da je međutočka na kutu manjem od 180 stupnjeva u odnosu na odredište jer će se u protivnom javiti greška na robotu. Brzina robota se zadaje u vrijednostima mm/sec.



Slika 69. Prikaz kružnog (Circular) gibanja

Primjeri naredbi:

- **C**

**P[2] 4000 mm/sec FINE**

**P[1]**

Naredba **C** je specifična, ide u dva reda **P[2]** je međutočka, **P[3]** je točka odredišta.

Gibanje po kružnom luku je varijacija gibanja po krugu, ali krivulja gibanja robota je proračunata iz više točaka.

### Točka do koje robot treba doći

Naredba za gibanje:

**J P[1: HOME] 100% FINE - Označen dio predstavlja naučenu točku ( P[1: HOME] )**

Drugi dio naredbe za gibanje definira točku koja je naučena i snimljena u memoriji robota. Ta procedura kod FANUC robota se naziva „dotjerivanje“ (engl. *touchup*). Da bismo točku definirali, potrebno je s robotom doći u određenu poziciju te pritisnuti tipku na zaslonu TOUCHUP. Točka će biti spremljena u memoriji te će robot u automatskom radu znati doći u tu točku. Prilikom učenja potrebno je paziti u kojem smo koordinatnom sustavu i koji alat koristimo. Točka će biti spremljena s tim koordinatama.

```

PICKANDPLACE
P[1] UF:0 UT:8          CONF:NUT 000
X 931.848 mm W         .000 deg
Y 126.455 mm P        -84.009 deg
Z 267.440 mm R         7.728 deg
Position Detail
15:L @P[1:HOME] 4000mm/sec FINE
16:
17: --TODO: Please touchup this to
   : the pick point.
18:L P[2:Pick Point] 200mm/sec FINE
   :
19:
Enter value

```

Slika 70. Koordinate snimljene točke P[1: HOME] u WORLD koordinatnom sustavu

U slučaju da smo definirali točku u koordinatnom sustavu koji ne koristimo u programu, u najboljem slučaju robot će javiti grešku, dok će u najgorem slučaju robot krenuti prema pogrešnoj točki i pritom se možda sudariti s određenim objektom na putanji. Isto vrijedi i za učenje s pogrešno odabranim alatom.

```

Busy Step Hold Fault
Run I/O Prod TCyc PICKANDPLACE LINE 0 T2 ABORTED W/TOOL 100%
PICKANDPLACE 15/61
13: --Move to pick standby point
14: --TODO: Please touchup this so it
   : is a little away from the pick
   : point.
15:L @P[1:HOME] 4000mm/sec FINE
16:
17: --TODO: Please touchup this to
   : the pick point.
18:L P[2:Pick Point] 200mm/sec FINE
   :
19:

```

Slika 71. Način definiranja točke - pritiskom na TOUCHUP izbornik točka P[1: HOME] je snimljena.

Oznaka broja u zagradama, P[1], označava vrijednost registra u kojem je snimljena pozicija. U slučaju da koristimo P[1] bilo gdje u istom programu, ta točka ima istu pozicijsku vrijednost. Uz broj možemo pridodati komentar, npr. P[1: HOME]. Komentarom HOME opisali smo što zapravo predstavlja definirana točka u prostoru.

### Brzina kojom robot treba doći u točku

Naredba za gibanje:

**J P[1: HOME] 100% FINE** - Označen dio predstavlja brzinu kojom se robot giba do točke.

**L P[1: HOME] 4000 mm/sec FINE** - Označen dio predstavlja brzinu kojom se robot giba do točke.

Linearna gibanja su prikazana u vrijednosti mm/sec, dok su zglobni pokreti prikazani u postotku maksimalne brzine koju robot može postići.

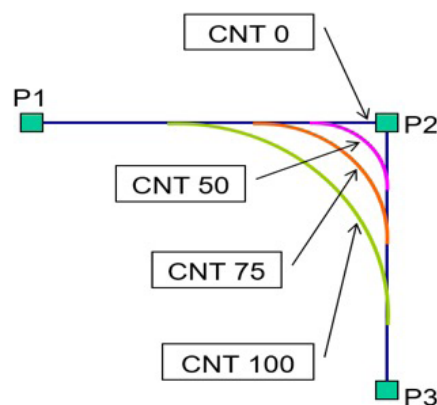
### Točnost koja je potrebna da bi robot došao u točku

Naredba za gibanje:

**J P[1: HOME] 100% FINE** - Označen dio predstavlja točnost.

**L P[1: HOME] 4000 mm/sec CNT100** - Označen dio predstavlja točnost.

Jako bitan parametar je način na koji dolazimo u točku, odnosno faktor točnosti. U slučaju da taj parametar ima vrijednost FINE, robot će doći točno u zadanu točku i u toj istoj točki će nakratko stati. Faktor ponovljivosti robota definira koliko smo u mogućnosti točno doći u točku (kod robota u primjeru taj faktor je +/- 0.01 mm). Ovaj način gibanja nije efikasan jer robot ima pauzu prije nego što krene u drugu točku. Da bismo postigli brže gibanje, potrebno je žrtvovati točnost u korist brzine i za to koristimo CNT faktor. U slučaju da CNT faktor u naredbi postavimo na 0 (CNT0), dobili smo ekvivalent parametra FINE. Međutim, ako postavimo veću vrijednost (maksimalno do 100), robot neće stati u zadanu točku, već će se truditi doći što je moguće bliže točki u ovisnosti o podešavanju. Na taj način smo dobili veću brzinu robota, a žrtvovali preciznost. U slučajevima u kojima je potrebna preciznost (npr. prilikom podizanja objekta), koristimo FINE parametar. U slučajevima gdje je potrebna brzina (npr. gibanje od osnovne pozicije do palete), koristimo CNT parametar.



Slika 72. Prikaz različitih CNT parametara. CNT0 odgovara parametru FINE, robot točno dolazi u točku P2

## 8.7. Pozivanje drugih programa u programu koji je izvodi

+Naredbom **CALL <ime\_programa>** možemo pozvati drugi program unutar programa koji se izvodi. CALL naredbu koristimo kad je potrebno refaktorirati programski kod. Dobra praksa je da programi budu sa što je moguće manje linija koda i podijeljeni u sekcije. U poglavlju (9.4. Program za prebacivanje objekata) koristili smo naredbu **CALL GRIPPER\_CLOSE** i **CALL GRIPPER\_OPEN**. Na ovaj način odvojili smo dio programa koji sadrži naredbe za kretanje robota od dijela programa koji otvara i zatvara hvataljku. Kompletan niz naredbi za otvaranje i zatvaranje hvataljke prebačen je u novi program.

Program za otvaranje hvataljke:

```
+1: DO[1: GRIPPER_OPEN]=ON
2: WAIT      .40(sec)
3: DO[1: GRIPPER_OPEN]=OFF
[End]
```

U primjeru programa poviše aktiviramo digitalni izlaz robota **DO[1: GRIPPER\_OPEN]=ON**. Na tom digitalnom izlazu fizički je žicom spojen signal za otvaranje hvataljke. Pošto se hvataljka ne zatvara trenutno (ovisno o tipu alata), potrebno je dodati određeno vrijeme čekanja. Naredbom **WAIT .40(SEC)** dodajemo vrijeme čekanja prije nego nastavimo s izvođenjem programa (prijelaz s linije 2 na liniju 3 programa). Vrijeme je prikazano u sekundama, u našem slučaju čekamo 0.4 sec odnosno 400 ms, što bi trebalo biti dovoljno vremena da se hvataljka zatvori. Nakon što se hvataljka zatvorila, potrebno je deaktivirati digitalni izlaz robota **DO[1: GRIPPER\_OPEN]=OFF**.

Program za zatvaranje hvataljke:

```
1: DO[2: GRIPPER_CLOSE]=ON
2: WAIT      .40(sec)
3: DO[2: GRIPPER_CLOSE]=OFF
[End]
```

Kod zatvaranje hvataljke imamo istu situaciju kao kod otvaranja, samo je razlika u tome što je potrebno aktivirati drugi digitalni izlaz **DO[2: GRIPPER\_CLOSE]=ON** da bi se hvataljka zatvorila. Trebamo imati na umu da se često hvataljka ne može trenutno zatvoriti pa je isto kao i kod otvaranja potrebno dodati vrijeme čekanja.

Svaki program koji pišemo automatski ima na kraju dodanu naredbu **[End]**. Ta naredba označava kraj programa. U slučaju da smo program pozvali iz nekog drugog programa koji se trenutno izvodi, nakon izvođenja naredbe **[End]** u programu koji je pozvan vraćamo se nazad u program koji se prethodno izvodio. Kao primjer možemo uzeti program u devetom poglavlju (9.4). Pozivamo program **CALL GRIPPER\_CLOSE** u liniji br. 7. Nakon što se izvede

cijeli **GRIPPER\_CLOSE** program koji smo opisali u ovom poglavlju i dođemo do **[End]** naredbe, vraćamo se u program koji je pozvao **GRIPPER\_CLOSE** na iduću liniju (br.8) te nastavljamo s izvođenjem programa.

Lista naredbi koje smo prošli do sada:

<b>UTOOL_NUM=...</b>	odabir alata koji se koristi u programu
<b>UFRAME_NUM=...</b>	odabir koordinatnog sustava koji se koristi u programu
<b>J P[...] 100% FINE</b>	zglobno (engl. <i>Joint</i> ) gibanje robota
<b>L P[...] 4000mm/sec FINE</b>	linearno (engl. <i>Linear</i> ) gibanje robota
<b>CALL &lt;ime_programa&gt;</b>	poziv drugog programa u program koji se trenutno izvodi
<b>DO[...]=ON</b>	aktiviranje digitalnog izlaza robota
<b>DO[...]=OFF</b>	deaktiviranje digitalnog izlaza robota
<b>WAIT ...(sec)</b>	vrijeme čekanja prije izvođenja iduće naredbe
<b>[End]</b>	označava kraj izvođenja programa



## 9. Pokretanje robota u automatskom radu

Svrha svih podešavanja i programiranja koje smo objasnili u prethodnim poglavljima je da omogućimo industrijskom robotu automatsko izvršavanje zadataka. Glavna prednost robota je što ponavljajuće zadatke rade s izvrsnom točnošću uz minimalni ljudski nadzor. Ponavljajuće radnje mogu obavljati koliko god je potrebno, često bez pauze, što utječe na poboljšanje proizvodnog procesa.

Kada smo detaljno testirali program robota pri malim brzinama (T1 način rada), u slučaju da je sve u redu možemo pristupiti pokretanju robota u automatskom radu.

### 9.1. Načini pokretanja robota u automatskom radu

Postoje dva načina pokretanja robota u automatskom radu:

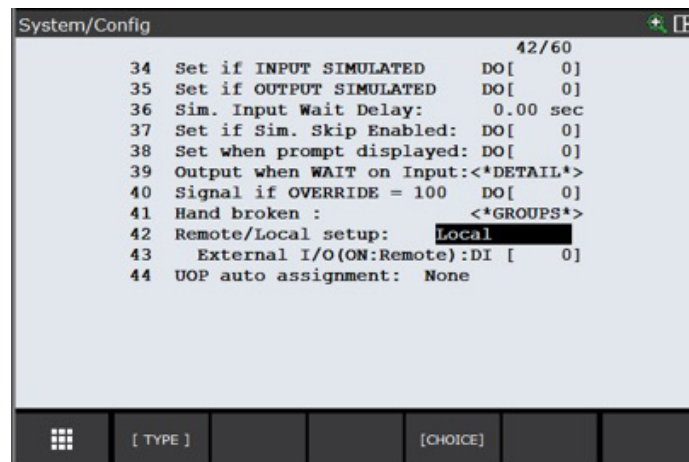
1. Lokalno (engl. *Local*)
2. Daljinski (engl. *Remote*).

Lokalno pokrećemo robota s centralne upravljačke jedinice R30-iB. Kada prebacimo preklopu u poziciju AUTO, više ne možemo koristiti ručnu upravljačku jedinicu za pokretanje programa. U slučaju da je ručna upravljačka jedinica uključena, robot će izbaciti alarm.



Slika 73. Centralna upravljačka jedinica

Nakon toga potrebno je u konfiguraciji robota odabrati lokalni način pokretanja. To možemo učiniti ako na ručnoj upravljačkoj jedinici pritisnemo tipku MENU, zatim u izborniku odaberemo NEXT, SYSTEM i CONFIG. U listi konfiguracije potrebno je odabrati parametar Remote/Local setup i postaviti ga na Local.



Slika 74. Prikaz konfiguracije robota i odabir lokalnog načina pokretanja robota

Kada smo odabrali lokalni način starta, moguće je robot pokrenuti pomoću standardnog panela za operatere koji se nalazi na centralnoj upravljačkoj jedinici (Standard Operator Panel - SOP). Spomenuli smo da u robotu može postojati više spremljenih programa, a pritiskom tipke START na standardnom panelu za operatere pokreće se onaj program koji smo odabrali tipkom SELECT na ručnoj upravljačkoj jedinici. Program koji je spreman za izvođenje i linija programa koja je iduća za izvršavanje može se provjeriti na statusnoj traci robota.



Slika 75. Prikaz odabranog programa na statusnoj traci: odabran je program BEGINNER i program je na samom početku (LINE 0)

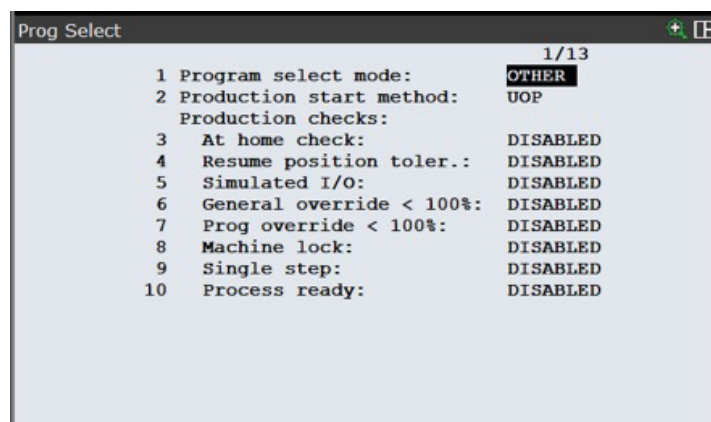
Program robota možemo pokrenuti daljinski. To znači da ne koristimo ručnu upravljačku jedinicu ili standardni panel za operatere. Signal za start može doći od strane nekog drugog uređaja, najčešće industrijskog računala. Industrijsko računalo, PLC (engl. *Programmable Logic Controller*), koristi se za upravljanje procesa u industriji. Industrijski robot je dio kompleksnog procesa koji se odvija da bi se ostvario određeni rad. Uspjeh ovisi o svakom pojedinom dijelu sustava. Da bi svaki pojedini dio bio sinkroniziran, koristimo industrijsko računalo kao sredstvo upravljanja cjelokupnog procesa. Umjesto pritiska tipkala START na SOP-u ili ručnoj upravljačkoj jedinici, industrijsko računalo šalje signal START robotu preko posebnih digitalnih signala koji su predviđeni za tu upotrebu. Poveznica između robota i industrijskog računala najčešće je Ethernet veza.

Kod FANUC robota digitalni signali predviđeni za spajanje na industrijska računala i druge uređaje nazivaju se UOP signali. Oni trebaju biti aktivirani u konfiguraciji robota. Nakon što smo poslali određene signale robotu koji su nužni za njegov START, potrebno je odabrati program koji će se izvoditi (u daljinskom načinu rada ne izvodi se program koji je odabran na ručnoj upravljačkoj jedinici).

Postoje četiri metode za odabir programa:

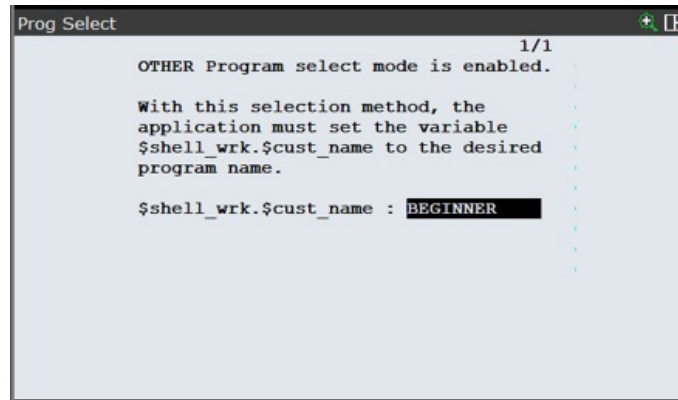
1. RSR
2. PNS
3. STYLE
4. OTHER.

Svaka od metoda ima svoj način odabira programa, prednosti i nedostatke. Najjednostavnija metoda je OTHER. Ako u konfiguraciji koristimo OTHER, robot će prilikom detektiranja START signala pokrenuti program koji smo naveli u samom podešavanju. Potrebno je pritisnuti tipku MENU na ručnoj upravljačkoj jedinici, zatim odabrati SETUP i nakon toga Prog Select. U izborniku koji se otvori potrebno je podesiti Program select mode na OTHER odabirom [CHOICE] izbornika.



Slika 76. Podešavanje metode za odabir programa, podesili smo parametar na OTHER

Program koji želimo pokrenuti podesimo odabirom izbornika DETAIL, u novom izborniku varijablu **\$shell\_wrk.\$cust\_name** podesimo na program koji želimo pokrenuti.



Slika 77. Slika prikazuje odabir BEGINNER programa koji će biti pokrenut prilikom aktivacije start signala

## 9.2. Razlike u odnosu na pokretanje u ručnom radu

Osnovne razlike u odnosu na ručni rad su:

1. isključena ručna upravljačka jedinica
2. način izvođenja programa
3. maksimalna brzina
4. sigurnosni protokoli.

U automatskom radu ručna upravljačka jedinica se ne može koristiti za upravljanje robotom te je uvjet za start automatskog rada da ona bude isključena. Unatoč tome što je isključena, ipak imamo limitirani set opcija koje možemo koristiti: npr. u mogućnosti smo pratiti izvođenje programa koji se izvodi na zaslonu, možemo smanjiti brzinu robota na tipkala označena + % i - % i pogledati status robota na statusnoj traci.

Program se u automatskom radu izvodi kontinuirano, od prve do zadnje linije. Kada dođemo do naredbe [engl. *End*], isti se program može ponoviti ili prekinuti.

Robot u automatskom radu radi s maksimalno mogućim brzinama, ne postoji određeni limit radi sigurnosti kao u ručnom radu.

Pošto industrijski robot predstavlja potencijalnu opasnost za čovjeka u industrijskom okruženju, roboti su često okruženi zaštitnom ogradom dok obavljaju akcije. Često su uključene dodatne mjere u smislu laserskih senzora ili kamera kako bi se osiguralo da nema čovjekove prisutnosti u radnom području robota. Da bi se robot pokrenuo u automatskom radu, svi sigurnosni uvjeti moraju biti zadovoljeni dok kod ručnog rada to nije slučaj. U ručnom radu strogi sigurnosni protokoli otežali bi podešavanje i programiranje robota.

## 10. Sigurnosno kopiranje i aktiviranje sigurnosne kopije

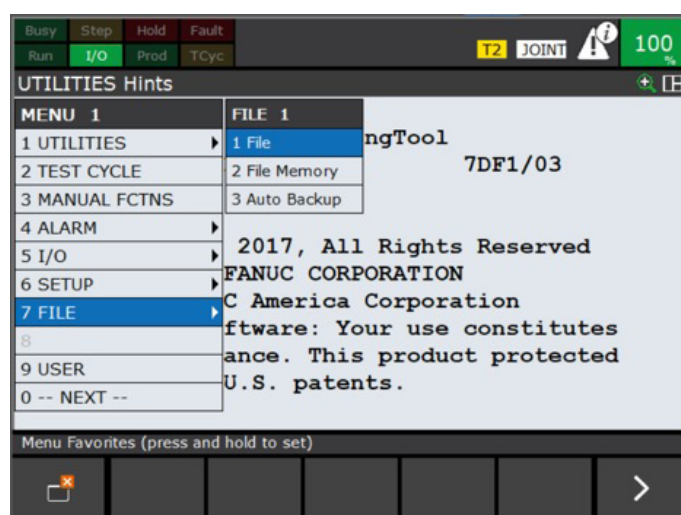
### 10.1. Svrha sigurnosnog kopiranja

Kroz ovaj priručnik stekli smo dojam što je potrebno i u koliko koraka napraviti sve da bismo podesili industrijskog robota. Što imamo kompleksniju aplikaciju, to će i samo podešavanje robota biti kompleksnije. Program robota i sva podešavanja su spremljena u robotu, međutim može se desiti više scenarija u kojima možemo izgubiti sve što je spremljeno. Iz tog razloga robot posjeduje opciju sigurnosne kopije. Generalni slučaj može biti kvar samog robota ili određenog dijela koji je zadužen za memoriju. U slučaju da nismo napravili sigurnosnu kopiju, cijeli posao bismo trebali ponoviti sa zamjenskim robotom. Ostali slučajevi mogu biti greška osobe koja programira robota, slučajno možemo izbrisati program koji je ključan za rad aplikacije. Dobra praksa bila bi snimanje sigurnosne kopije nakon svakog testiranja programa.

Sigurnosna kopija ne sadrži samo programe koje smo kao korisnici robota napisali i podešavanja korisničkih okvira, već i osnovne postavke koje su tvornički postavljene na robotu. Ako se iz nekog razloga izgube tvorničke postavke (čest slučaj može biti potrošnja baterije glavne upravljačke jedinice), možemo ih vratiti pomoću sigurnosne kopije.

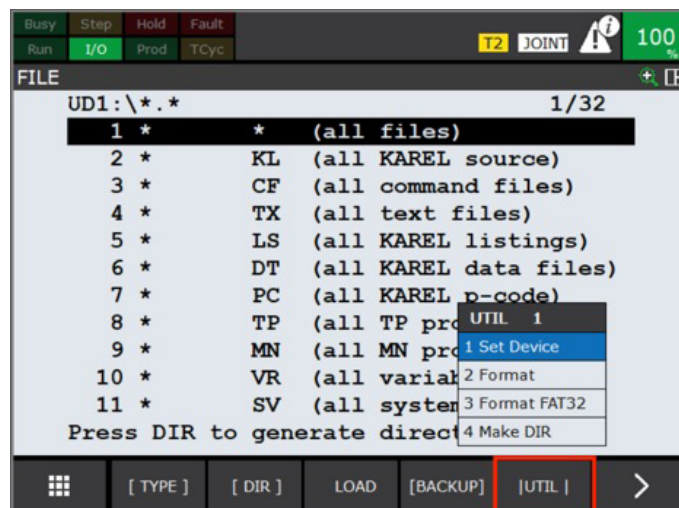
### 10.2. Izrada i aktiviranje sigurnosne kopije

Najlakši način za izradu sigurnosne kopije je koristeći ručnu upravljačku jedinicu. Potrebno je pritisnuti tipku MENU, zatim u prvom izborniku izabrati FILE i nakon toga ponovno File (prikaz se nalazi na slici ispod).



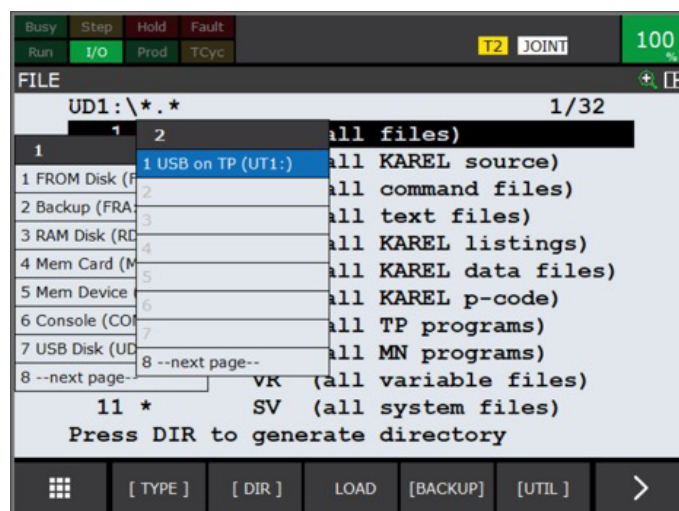
Slika 78. Odabir izbornika FILE prvi je korak koji moramo napraviti da bismo došli do sigurnosne kopije

Otvorit će se izbornik koji prikazuje sve datoteke koje se nalaze na robotu. Datoteke možemo podijeliti na sistemske, koje su potrebne za osnovni rad robota (tvorničke postavke) i korisničke, koje su izradili programeri. Sigurnosnu kopiju možemo napraviti na više načina i različitih uređaja. Najčešće i najlakše radimo sigurnosnu kopiju na USB memoriju. Imamo dva USB ulaza: jedan se nalazi na glavnoj upravljačkoj jedinici, a drugi na ručnoj upravljačkoj jedinici. Potrebno je odabrati koji USB ulaz koristimo prilikom izrade sigurnosne kopije. To radimo na način da na grafičkom sučelju odaberemo UTIL zatim odabir uređaja (engl. Set Device).



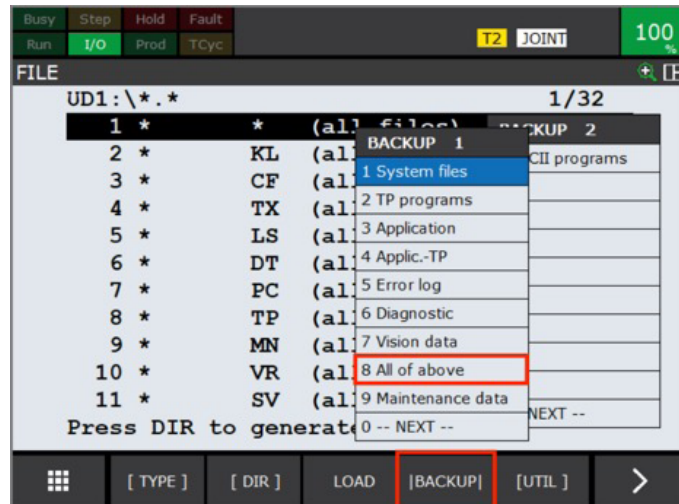
Slika 79. Odabir uređaja na koji želimo izraditi sigurnosnu kopiju

U sljedećem izborniku odaberemo USB on TP (UT:1), odnosno USB ulaz na ručnoj upravljačkoj jedinici.



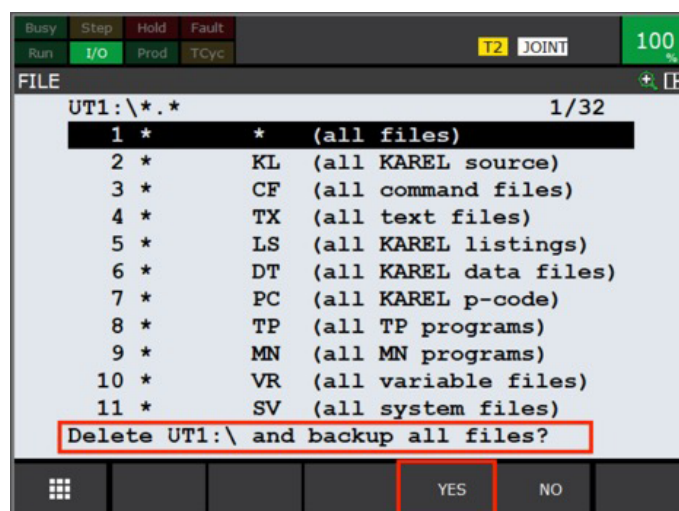
Slika 80. Prikaz odabira USB ulaza na ručnoj upravljačkoj jedinici kao odredišta spremanja sigurnosne kopije

Nakon odabira odredišta spremanja, možemo napraviti sigurnosnu kopiju odabirom [BACKUP] na grafičkom zaslonu. U izborniku možemo odabrati što želimo spremiti. Možemo spremiti samo sistemske datoteke, korisničke programe, određenu dijagnostiku i sl. Međutim, najlakše je spremiti kompletno stanje robota odabirom opcije za sve površine (engl. *All of above*).



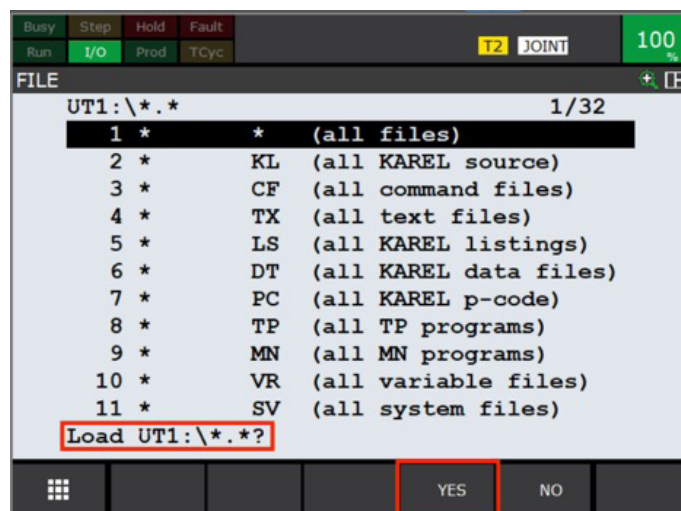
Slika 81. Odabir izrade sigurnosne kopije koja sprema kompletno stanje robota

Na grafičkom sučelju pojavit će se tekst koji nas upozorava da će već postojeće datoteke na USB memoriji biti izbrisane. Ako je to u redu, potvrdimo odabirom YES. Kreće proces snimanja novih datoteka na USB memoriju koji traje nekoliko sekundi. Kad je proces završen, sigurnosna kopija je izrađena.



Slika 82. Potrebno je potvrditi slažemo li se da se izbrišu prethodne datoteke s USB memorije

S obzirom na to da smo već odabrali uređaj na koji spremamo sigurnosnu kopiju, to isto podešavanje vrijedi za aktivaciju sigurnosne kopije. Sigurnosnu kopiju koja već postoji na USB memoriji možemo prebaciti na robot odabirom LOAD opcije na grafičkom sučelju.



Slika 83. Potrebno je potvrditi da želimo aktivirati sigurnosnu kopiju s odabranog uređaja

Odabirom YES potvrđujemo da želimo kopirati sigurnosnu kopiju s USB memorije na sami robot. Proces traje nekoliko sekundi i nakon toga sigurnosna kopija će biti aktivna na robotu.



## 11. Integracija industrijskih robota u proizvodne lance

Industrijski roboti automatiziraju određene procese u industriji, međutim oni su sami po sebi samo dio industrijskog automatizacijskog lanca. Potrebno je razlikovati industrijsku automatizaciju i robotsku automatizaciju.

Industrijska automatizacija je proces upravljanja sustavima bez neposredne ljudske intervencije. Primarni cilj je povećati učinkovitost, smanjiti troškove i poboljšati kvalitetu određenog industrijskog procesa. Da bi to bilo moguće, potrebno je baviti se širom slikom, odnosno sustavom u cjelini. Primjer automatizacije su proizvodne linije u tvornicama za proizvodnju kalupa za staklene boce ili proizvodne linije za proizvodnju automobila. U procesu industrijske automatizacije znamo koji je finalni proizvod (kalup ili automobil) te je potrebno razmišljati kako od sirovine i u koliko koraka doći do finalnog proizvoda sa što manje ljudske intervencije.

Robotska automatizacija se bavi automatiziranjem određenog procesa u procesu industrijske automatizacije. Robotsku automatizaciju možemo smatrati podskupom industrijske automatizacije. Ako se bavimo robotskom automatizacijom, često nije potrebno razmišljati o sustavu kao cjelini, već se bavimo dizajnom, konstrukcijom i primjenom robota za određeni zadatak. Primjer robotske automatizacije je robot koji obavlja funkciju zavarivanja ili montažu. U procesu industrijske automatizacije razmišljamo o odabiru adekvatnog robota, senzora i načina upravljanja koji će automatizirati određeni posao. Kod navedenog primjera zavarivanja dovoljno je znati kakav proizvod dolazi na obradu, kako izgleda proces zavarivanja i gdje je potrebno odložiti obrađeni proizvod.

### 11.1. Mogućnosti povezivanja robota s ostalim elementima u industrijskom lancu

Kroz poglavlja ovog priručnika pokazali smo kako se pokreće robot, kako podesiti alat na robotu i postaviti koordinatne sustave. Koncentrirali smo se na robot kao izolirani sustav, međutim nakon osnovnih podešavanja potrebno je naučiti kako da robot radi s drugim elementima u sustavu. Česta aplikacija robota je posluživanje CNC strojeva (engl. *Computer Numerical Control*).



Slika 84. Haas CNC stroj

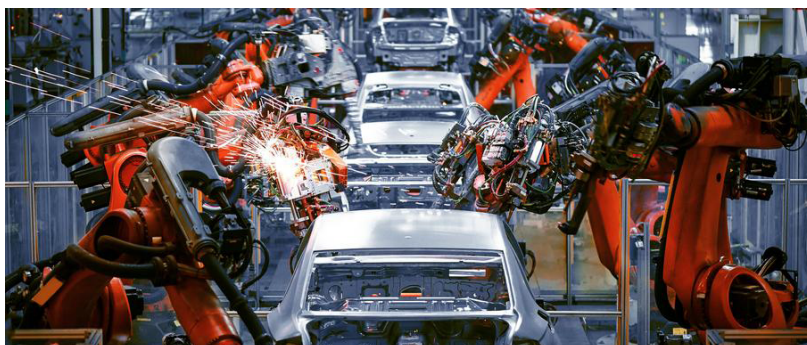
CNC strojeva ima raznih vrsta i najčešće se koriste za obradu sirovina. Postoje razne vrste obrade poput glodanja, bušenja, rezanja itd. Glavna prednost CNC strojeva je visoka preciznost u obradi sirovina. U industriji CNC strojevi proizvode metalne dijelove, sitne elektroničke dijelove, graviraju i režu. Neke od navedenih operacija bilo bi teško ručno izvesti ili ih čak ne bi bilo moguće izvoditi.

Da bi robot bio u mogućnosti poslužiti stroj, potrebno je na određeni način ostvariti komunikaciju sa strojem. Robot bi trebao znati par osnovnih informacija o stroju:

- vrata stroja su zatvorena ili otvorena
- stroj je završio s obradom ili je u obradi
- stroj u svojoj hvataljci sadrži sirovinu, gotov proizvod ili je hvataljka prazna.

U ovisnosti o navedenim signalima, programeri robota trebaju napisati upravljački program koji direktno ovisi i o drugim elementima u sustavu.

Na isti način možemo promatrati robotsku automatizaciju u automobilskoj industriji.



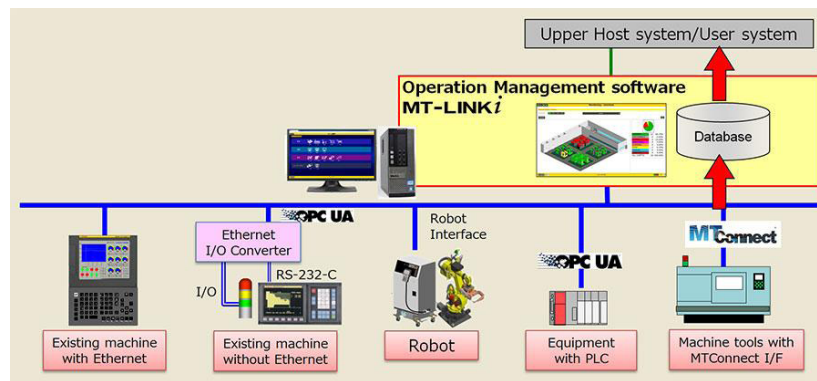
Slika 85. Prikaz proizvodne linije u automobilskoj industriji

Na gornjoj slici možemo vidjeti da više robota radi zajedno različite poslove na karoseriji automobila. Robot i u ovom slučaju treba imati dodatne informacije o procesu:

- potrebno je imati informaciju o lokaciji drugog robota
- potrebno je znati je li drugi robot završio s poslom.

U oba slučaja, tj. primjeru posluživanja stroja ili proizvodnje automobila, može doći do problema ako upravljački program ne uzme u obzir informacije drugih dijelova sustava. Ako ne znamo jesu li vrata stroja otvorena, može doći do sudara s njima. Isto tako kada roboti rade u paru, a ne znaju poziciju drugoga, može doći do sudara.

Industrijski roboti imaju mogućnost povezivanja s drugim robotima, CNC strojevima i bilo kojim drugim dijelovima industrijskog lanca.



Slika 86. Prikaz komunikacije robota s drugim dijelovima sustava (CNC strojevima)

Postoje različiti standardizirani mrežni protokoli koji se koriste za komunikaciju između uređaja u industrijskim okruženjima:

- EtherCAT
- EtherNet/IP
- Profinet
- CC Link IE
- Modbus TCP.

Svaki od navedenih protokola je varijanta industrijskog Etherneta i to je najčešći način kako se povezuju elementi u industrijskom lancu. Prednost ovog načina povezivanja je:

- visoka brzina prijenosa informacija
- spajanje preko klasičnog Ethernet kabela
- većina proizvođača opreme koristi bar jedan od navedenih protokola.

Drugi čest način povezivanja robota i ostalih uređaja je pomoću ulazno/izlaznih signala (I/O signala). U glavnu upravljačku jedinicu može se ugraditi dodatni modul na koji se žicama mogu spojiti signali s drugih uređaja ili senzora i to bi bio ulazni modul. Isto tako robot može imati dodatni izlazni modul koji može žičanom vezom slati signale drugim uređajima. U principu su to signali ON ili OFF (Boolean logika).

Kao što imamo ulazno/izlazne digitalne module, postoji i analogna verzija modula. Analogni ulazno/izlazni moduli koriste standardne strujne i naponske signale. U slučaju da koristimo signal struje, koristimo vrijednosti od 4 mA do 20 mA, a ako koristimo naponske signale, onda je vrijednost napona od 0 do 10 V. U oba slučaja donja granica predstavlja nulu, a gornja granica maksimalnu vrijednost veličine. Kao primjer možemo uzeti temperaturu. U slučaju da senzor mjeri temperaturu od 0 do 100 stupnjeva, donja vrijednost će predstavljati 4 mA (0 V), a gornju vrijednost od 100 stupnjeva će predstavljati 20 mA (10 V).

Komunikacija pomoću digitalnih signala najčešće se koristi kada je potrebno senzore spojiti direktno na robot.

## 11.2. Interakcija robota s PLC uređajima: Pregled i praksa

Najčešće se u praksi komunikacija industrijskog robota s ostalim dijelovima sustava uspostavlja pomoću industrijskog digitalnog računala (engl. PLC - *Programmable Logic Controller*). PLC je prilagođen za kontrolu industrijskih procesa kao što su npr. proizvodnja cementa, plastičnih dijelova u automobilskoj industriji i upravljanje robotskim ćelijama. Za razliku od računala koja koristimo svakodnevno, PLC ima visoku pouzdanost, jednostavan je za programiranje i ima toleranciju na teške radne uvjete. Bitno je spomenuti da su PLC računala dizajnirana na način da brzo obrađuju ulaze i izlaze, što je ključno za procese automatizacije i kontrole.

Najčešće u robotskoj automatizaciji PLC ima ključnu ulogu. Na PLC su spojeni signali sa sigurnosnih vrata, strojeva koje robot poslužuje i raznih ostalih elemenata koji sudjeluju u proizvodnji. PLC je spojen s robotom pomoću jednog od industrijskih Ethernet protokola. Pošto PLC ima sve signale potrebne da bi se ciljani proces odradio, često on preuzima kontrolu i daje naredbe za start robotu. Kao primjer možemo uzeti aplikaciju u kojoj robot poslužuje CNC stroj. PLC dobiva signal da CNC stroj nema obradni komad u hvataljci i da su vrata otvorena, zadovoljeni su svi uvjeti da robot uzme sirovi komad s ulaza u obradni proces te ga odloži u hvataljku stroja. PLC će robotu poslati signal za start. Isto tako PLC će dobiti signal od CNC stroja da je obrada sirovog komada završena te je potrebno otvoriti vrata stroja i poslati robotu signal za start preuzimanja komada iz stroja te odlaganje na izlaz obradnog procesa.

Integracija PLC računala i FANUC robota se radi pomoću standardnih UOP (engl. *User Operated Panel signals*). UOP signali definirani su za ulazne i izlazne signale robota.

Sa PLC-a moramo poslati određenu sekvencu UOP ulaznih signala robotu. Lista bitnih UOP ulaznih signala:

1. **UI[1:IMSTP]** - U slučaju da je signal u OFF stanju (isključen), robot odmah staje.
2. **UI[2:HOLD]** - U slučaju da je signal u OFF stanju (isključen), robot pauzira izvršavanje programa.
3. **UI[3:SFSPD]** - U slučaju da je signal u OFF stanju (isključen), program robota se pauzira i brzina robota se smanji na određenu sigurnosnu brzinu.
4. **UI[4:Cycle stop]** - U slučaju da je signal u ON stanju (uključenom), robot će se zaustaviti kada završi određenu programiranu sekvencu.
5. **UI[5:Fault reset]** - U slučaju da je signal u ON stanju (uključenom), resetiraju se sve greške na robotu.
6. **UI[6:Start]** - U slučaju da je signal u ON stanju (uključenom), robot starta s izvođenjem programa.
7. **UI[7:Home]** - U slučaju da je signal u ON stanju (uključenom), robot se vraća u HOME poziciju (program za povratak u HOME poziciju mora biti podešen).
8. **UI[8:Enable]** - U slučaju da je signal u ON stanju (uključenom), robot se može pomaknuti.

Iz navedenih signala proizlazi sekvenca za start robota koji PLC mora generirati. Za start robota sa PLC-om potrebno je:

1. uključiti **UI[1:IMSTP]**
2. uključiti **UI[2:HOLD]**
3. uključiti **UI[3:SFSPD]**
4. uključiti **UI[8:Enable]**
5. poslati puls na **UI[5:Fault reset]** (reset grešaka na robotu u slučaju da postoje)
6. poslati puls na **UI[6:Start]** (robot starta s izvršavanjem programa).

Dobra praksa prije početka slanja sekvence za start sa PLC-a na robot je provjeriti vrijednosti UOP izlaznih signala (signala koje robot šalje PLC-u). Na taj način prije samog slanja naredbe za start provjerimo u kojem je stanju robot i je li uopće spreman za start.

Lista bitnih UOP izlaznih signala:

1. **UO[1:Cmd enabled]** - U slučaju da je signal u ON stanju (uključenom), robot je u remote načinu rada i nije u grešci. Možemo poslati start naredbu robotu.
2. **UO[2:System ready]** - U slučaju da je signal u ON stanju (uključenom), servomotori robota su uključeni.
3. **UO[3:Prg running]** - U slučaju da je signal u ON stanju (uključenom), program robota se izvršava.
4. **UO[4:Prg paused]** - U slučaju da je signal u ON stanju (uključenom), program je pauziran.
5. **UO[5:Motion held]** - U slučaju da je signal u ON stanju (uključenom), robot se drži u HOLD stanju (UI[2:HOLD] signal je u stanju OFF).

6. **UO[6:Fault]** - U slučaju da je signal u ON stanju (uključenom), robot je u grešci.
7. **UO[7:At perch]** - U slučaju da je signal u ON stanju (uključenom), robot je u podešenom referentnom stanju.
8. **UO[8:TP Enabled]** - U slučaju da je signal u ON stanju (uključenom), ručna upravljačka jedinica je uključena.
9. **UO[8:Batt alarm]** - U slučaju da je signal u ON stanju (uključenom), CMMOS RAM baterija robota ima napon manji od 2.6 V.
10. **UO[10: Busy]** - U slučaju da je signal u ON stanju (uključenom), robot je zauzet određenom radnjom (npr. izvršavanjem programa).

Pomoću UOP izlaza možemo napisati program upravljanja koji će uzeti u obzir stanje robota te provjeriti jesu li zadovoljeni svi uvjeti za start. Napredni PLC program bi mogao kontrolirati sljedeće:

Potrebno je osigurati da je **UO[1:Cmd enabled]** u stanju ON prije slanja signala za start (**UI[6:Start]**).

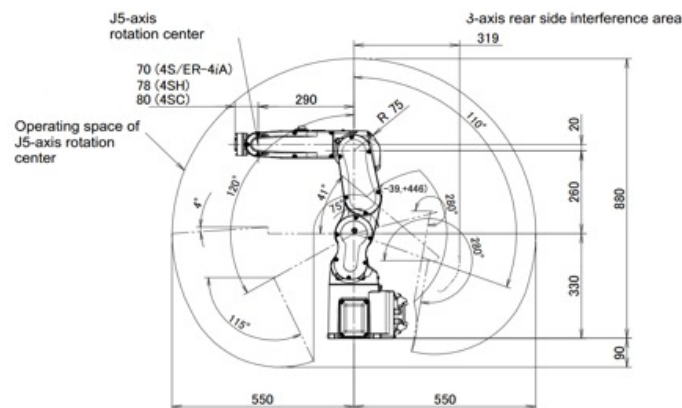
1. Nije potrebno slati start signal u slučaju da imamo **UO[3:Prg running]**, **UO[5:Motion held]** ili **UO[6:Fault]** u ON stanju.
2. U slučaju da je **UO[6:Fault]**, znamo da je na robotu greška te je potrebno uključiti **UI[5:Fault reset]** da bi se greška resetirala i robot mogao startati.
3. Ako želimo program startati iz određene pozicije, možemo provjeriti je li **UO[7:At perch]** u stanju ON te nakon toga krenuti sa startom.
4. U slučaju da je ručna upravljačka jedinica uključena, nije moguće pokrenuti robot u automatskom načinu rada. PLC tu informaciju može proslijediti dalje u sustav (npr. uređaju za operatore s grafičkim sučeljem HMI – engl. *Human Machine Interface*).
5. Na HMI možemo poslati informaciju da je baterija robota blizu kraja ako je signal **UO[8:Batt alarm]** u ON stanju.

## 12. Paket za simulaciju FANUC industrijskih robota: Uvod i uporaba

### 12.1. Općenito o paketima za simulaciju

Robotska automatizacija može biti kompleksna u smislu elemenata u samom procesu, npr. robot može posluživati više strojeva, više robota može raditi zajedno. Sve to moramo uzeti u obzir prilikom konstruiranja robotske ćelije. Potrebno je izmjeriti udaljenost između strojeva, izračunati poziciju gdje robot treba biti montiran jer je jedan od najvažnijih parametara doseg samog robota. Robotska ruka ima određene limite po svakoj osi te nismo u mogućnosti doći u svaku točku prostora. U alatima za simulaciju moguće je složiti prostor kakav bi trebao izgledati u stvarnosti s realnim mjerama i točno postavljenim lokacijama strojeva i robota koji sudjeluju u procesu.

Simulacijski paketi mogu učitati CAD (engl. *Computer Aided Design*) datoteke, odnosno modele strojeva i drugih robota. Kada je prostor za simulaciju postavljen, možemo se gibati s robotom u simulacijskom prostoru i na taj način provjeriti koji su limiti robota te možemo li odraditi sve zadatke s robotom. Roboti i strojevi su često teški (tipičan stroj teži 5 tona). U slučaju da smo napravili grešku u mjerenju ili dizajnu, promjena košta vremena i rada. Simulacijski alati nam pomažu da predvidimo moguće greške i ako postoje, da ih na vrijeme ispravimo.



Slika 87. Nacrt koji prikazuje doseg robota ER-4iA - vidimo da je maksimalni doseg 550 mm

Druga velika prednost je što paketi za simulaciju omogućavaju pisanje programa i pokretanje istog. Nije potrebno čekati da se robot i ostala oprema montiraju u tvornici, već je cijelu sekvencu moguće testirati u *offline* načinu rada.

Ako smo vjerno složili simulacijsku okolinu, možemo biti sigurni da nećemo imati problema s dosegom i već imamo pripremljen program. Dovoljno je samo popraviti pojedine točke u koje robot mora doći jer ipak nismo u mogućnosti predvidjeti eventualne izmjene koje mogu nastati u stvarnoj robotskoj aplikaciji.

## 12.2. FANUC paket za simulaciju

Svaki proizvođač ima svoj paket za simulaciju, no svi su jako slični i koriste iste principe. U ovom poglavlju upoznat ćemo Roboguide, FANUC-ov paket za simulaciju. Roboguide nam omogućava da napravimo 3D model robotske aplikacije. U tom 3D modelu možemo provjeriti jesu li pozicije elemenata u redu (doseg robota) i možemo li programirati robota. Posebna prednost alata je činjenica da prototip aplikacije možemo napraviti računalom, ne moramo imati fizički dostupan robot i strojeve da bismo provjerili mogućnost izvedbe. Realan 3D prikaz smanjuje mogućnost grešaka u realnoj aplikaciji.

Roboguide ima već gotove softverske alate koji nam olakšavaju simulaciju pojedinih tipičnih aplikacija:

- HandlingPRO - olakšava programiranje rukovanja objektima, sadrži 3D modele strojeva, pokretnih traka te je moguće iz modela izvući robotski program.
- PaintPRO - olakšava programiranje aplikacija bojanja, moguće je grafički označiti na modelu dio koji je potrebno obojati i iz te informacije se može generirati program.
- PalletPRO and Pallet Tool - olakšava programiranje aplikacija za paletiziranje i depaletiziranje. Sadrži standardne 3D modele elemenata koji se koriste u industriji.
- WeldPRO - olakšava programiranje aplikacija zavarivanja, moguće je simulirati putanje kojima varimo određeni materijal te generirati program koji kasnije možemo prebaciti u robot.

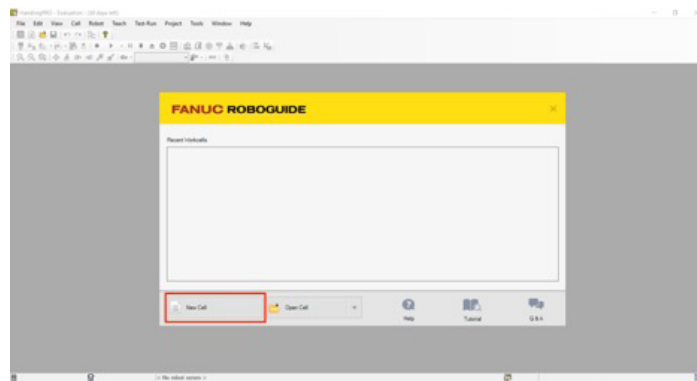
Korištenje simulacijskog paketa ubrzava samu montažu aplikacije i programiranje robota. Većinu posla možemo napraviti u izvan mrežnom (engl. *Offline*) načinu rada. Simulacijsko okruženje složeno u Roboguideu možemo spojiti s realnim robotom te direktno s računala prebacivati program i podešavanja na sami robot. Imamo mogućnost nadziranja kako se program robota izvodi.

## 12.3. Dizajniranje nove robotske ćelije

U sljedećih par koraka opisat ćemo kako se dizajnira nova robotska ćelija s Roboguide paketom za simulaciju. Preduvjet je da imamo Roboguide instaliran na računalu.

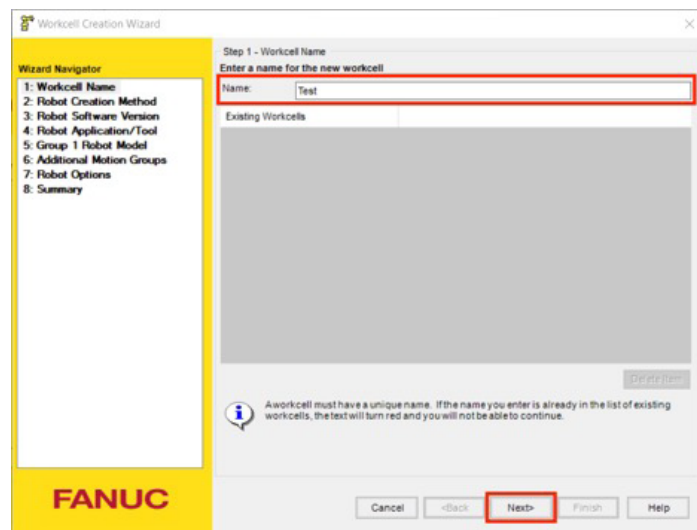
Prilikom pokretanja Roboguidea na računalu otvorit će se prozor prikazan na slici ispod.





Slika 88. Roboguide početni ekran

Na početnom ekranu odaberemo izbornik *New Cell*. U sljedećem koraku odaberemo ime projekta odnosno robotske ćelije. Postoji ograničenje u smislu da ne možemo imati više ćelija istog imena.

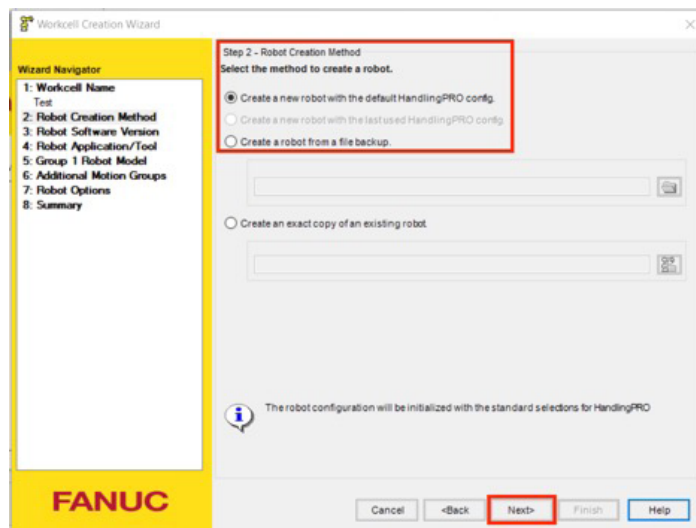


Slika 89. Odabir naziva robotske ćelije

Robotsku ćeliju moguće je napraviti na dva načina:

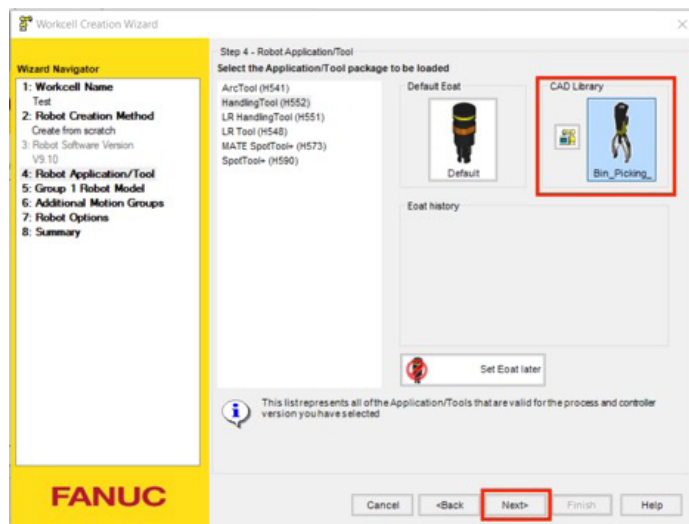
1. inicijaliziranjem nove ćelije
2. iz sigurnosne kopije postojećeg robota.

U ovim uputama koristit ćemo metodu inicijaliziranja nove ćelije. S odabirom nove ćelije potrebno je izabrati novi robot, dok je odabirom alternativne metode (sigurnosna kopija) robot već odabran i svi parametri robota prebacuju se u samu simulaciju.



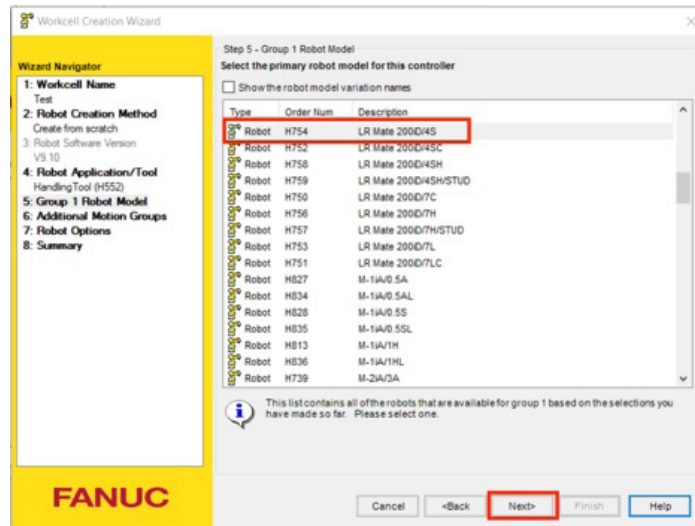
Slika 90. Možemo napraviti novi projekt na način da inicijaliziramo novi ili ga napravimo iz sigurnosne kopije (engl. Backup)

Sljedeći korak nam pruža odabir vrste hvataljke robota. Za odabranu aplikaciju (u našem slučaju HandlingPRO) u izborniku možemo odabrati neki od zadanih predložaka, zatim postoji mogućnost unosa CAD projekta iz knjižnice, a moguće je ovo podešavanje ostaviti za kasnije. Na slici 90 odabrana je hvataljka iz CAD knjižnice. U knjižnicu se može dodati bilo koji CAD projekt koji je napravljen prilikom samog dizajniranja robotske ćelije.



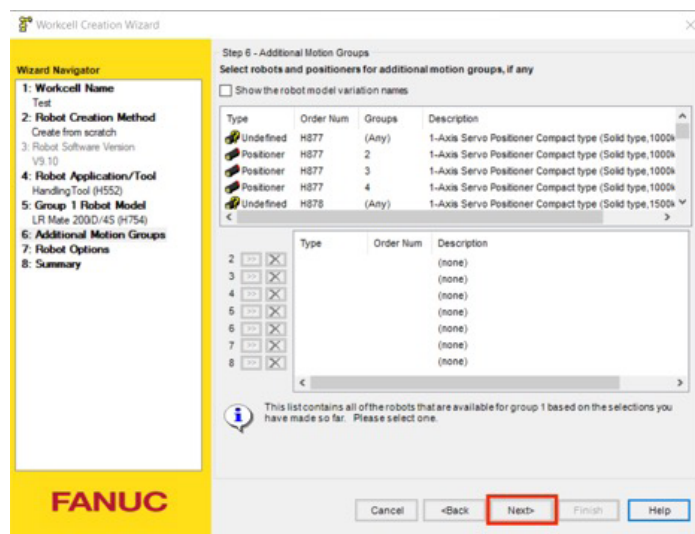
Slika 91. Odabir hvataljke robota

Postoji više tipova robota koji se razlikuju u mehaničkoj konstrukciji, dosegu i nosivosti (engl. Payload). U sljedećem koraku biramo tip robota koji koristimo u simulaciji.



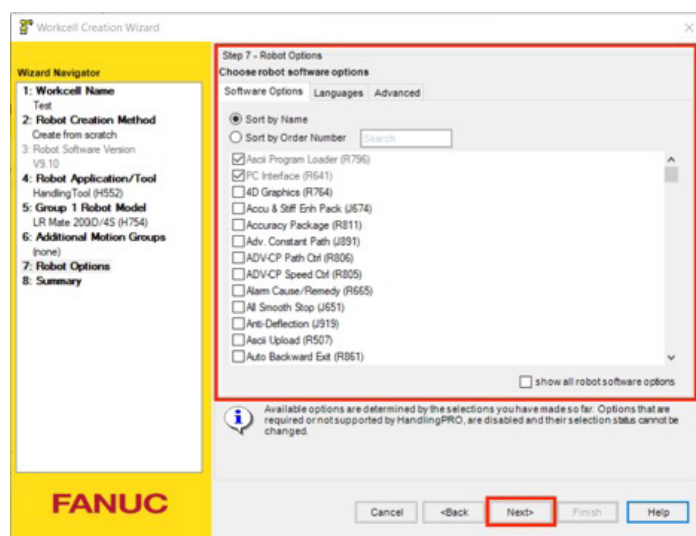
Slika 92. Odabir tipa robota

Robot koji smo odabrali u prethodnom koraku ima šest osi, međutim moguće je dodati na sami robot dodatne pokretne osi, npr. u slučaju da kao alat montiramo još jedan motor. Podešavanje robota za takav slučaj možemo dodati u sljedećem koraku. U našem slučaju ovaj izbor ostavljamo praznim i pritisnemo dalje (engl. *Next*) za završetak podešavanja.



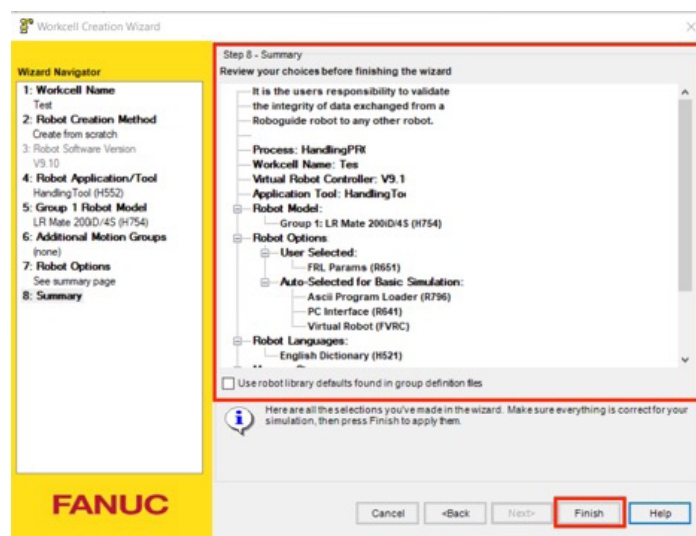
Slika 93. Odabir dodatnih pokretnih osi

Na sam robot možemo instalirati dodatne pakete. Neki od paketa su samo softverske prirode, a kod određenih paketa sama mehanička konstrukcija mora imati dodatnu funkciju. Primjeri dodatnih funkcija su automatski oporavak od kolizije (engl. *Auto Collision Recovery*) i dodatni komunikacijski modul (engl. *CC-Link interface*). Osnovni paketi koje sadrže svi roboti su već odabrani, npr. sučelje za spajanje s računalom (engl. *PC interface*).



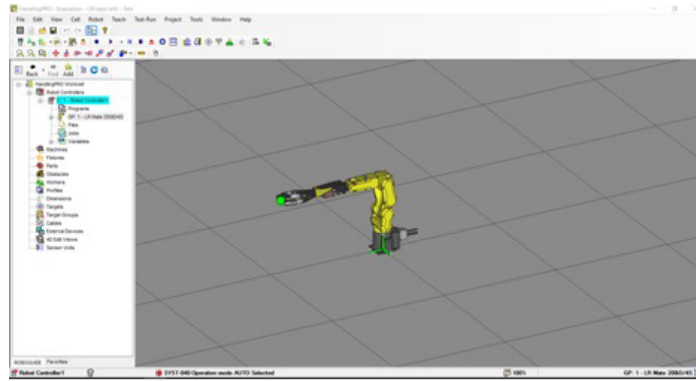
Slika 94. Odabir dodatnih funkcija robota

Pritiskom na dalje (engl. Next) dolazimo do zadnjeg izbora gdje možemo provjeriti jesu li sva podešavanja koja smo odabrali u prethodnim koracima u redu. Ako smo pronašli određenu grešku u podešavanju, postoji mogućnost vraćanja na prethodni korak izborom nazad (engl. Back). U slučaju da je sve u redu, odabirom završetak (engl. Finish) završavamo cijeli proces.



Slika 95. Sažetak svih odabranih podešavanja robotske ćelije

Nakon podešavanja, otvorit će se prostorni prikaz s robotom koji smo odabrali. Robot će imati sva podešavanja i imat će montiranu hvataljku.



Slika 96. Rezultat podešavanja robotske ćelije

U prethodnim koracima opisali smo osnovna podešavanja Roboguide paketa. S ovim osnovnim podešavanjima dobili smo robot u prostoru s kojim možemo raditi sve što i s fizičkim robotom. Možemo uključiti opciju ručne upravljačke jedinice te napraviti sve vježbe koje smo do sada objasnili u priručniku.

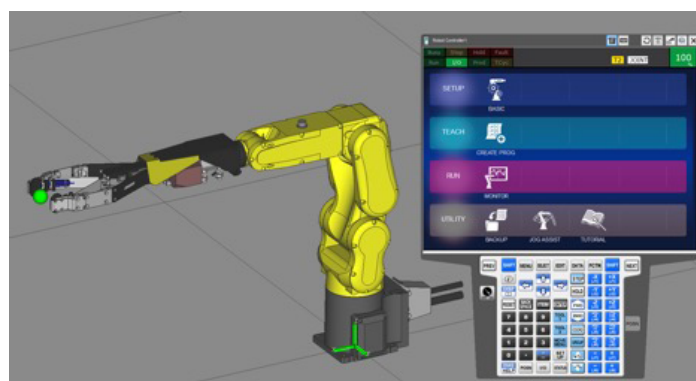
Imamo mogućnost:

- ručnog pokretanja robota
- određivanja TCP-a robota
- definiranja korisničkih koordinatnih sustava radnog objekta
- pisanja programa robota
- pokretanje robota u automatskom radu.

Ručna upravljačka jedinica može se uključiti pritiskom na prvu opciju druge alatne trake koja je prikazana na slici 96.



Slika 97. Otvaranje virtualne ručne upravljačke jedinice



Slika 98. Uključena je virtualna ručna upravljačka jedinica kojom možemo odraditi sve radnje objašnjene u priručniku

## Popis literature

- 1 Siciliano B., Khatib O. (2008), Springer Handbook of Robotics, izdanje 2008, Springer
- 2 Jon Stenerson (2002), Industrial Automation and Process Control, izdanje 1, Prentice Hall
- 3 Technical Documentation: ER-4iA Robot with R-30iB Mate Plus Controller, V5.3 (S/W Version 9.3), FANUC





**EduSplit** Obrtna tehnička škola  
Regionalni centar kompetentnosti Split



Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Za više informacija o EU fondovima molimo pogledajte web-stranicu  
Ministarstva regionalnoga razvoja i fondova Europske unije.  
[www.strukturfondovi.hr](http://www.strukturfondovi.hr)