



Programiranje robota

FANUC – osnovne podloge

Ver. 21.10.2021.

Sadržaj

Uvodni dio vezan uz sigurnost rada sa robotima	3
Konvencija označavanja.....	3
1. Oprema.....	4
1.1. Karakteristike opreme	4
1.2. Upravljačka konzola	6
1.3. Upoznavanje sa opremom	8
2. Osnove rada sa robotom	8
2.1. Zaslon.....	9
2.2. Ručno pokretanje robota (jog):	10
2.3. Osnove gibanja.....	11
2.3.1. Interpolacija gibanja zglobovima.....	11
2.3.2. Linearna interpolacija gibanja	11
2.3.3. Način izvršavanja zadanog gibanja	12
3. Koordinatni sustavi.....	13
3.1. Koordinatni sustav robota.....	13
3.2. Koordinatni sustav alata	13
3.3. Izbor aktivnog koordinatnog sustava	15
3.4. Offset i tool offset	15
4. Ulazno izlazni signali (I/O)	16
5. Napredne upravljačke strukture i programski elementi	17
5.1. Uvjetne IF naredbe	17
5.2. Petlje.....	17
5.2.1. Beskonačna petlja	17
5.2.2. FOR petlja	18
5.3. Poziv drugog programa	19
5.4. Numerički Registri	19
5.5. Izrada brojača	19
5.6. Čekanje - Wait.....	19
5.7. Pozicijski registri	20
6. Konfiguracije robotske ruke i izrada koordinatnih sustava	21
6.1. Izrada korisničkog koordinatnog sustava alata(UTOOL)	21
6.2. Korisnički koordinatni sustavi	22
6.2.1. Metoda tri točke	22
6.2.2. Metoda četiri točke	23
6.3. Izbor aktivnog koordinatnog sustava	23
6.4. Offset i tool offset	24
7. Razno.....	24
7.1. Brojač vremena (Timer).....	25
7.2. Override	25
7.3. Komentari (Remark)	25

8. Skip condition	25
Zadatak 1	27
Zadatak 2	29
Zadatak 3	30
Zadatak 4	31

Uvodni dio vezan uz sigurnost rada sa robotima.

- Rad pri malim (kontroliranim) brzinama
- Održavati siguran razmak od robota
- Nikada vizualno blokirati prostor između robota i osobe koja upravlja robotom (operator)
- Testiranje programa u koračnom (STEP) načinu rada pri kontroliranim brzinama – izvršavanje liniju po liniju koda

Konvencija označavanja

- PODVUČENO VELIKA SLOVA – oznake tipki upravljačke konzole
- *Kurziv* – engleski termini
- DRUGI_font – naredbe programa upravljačke konzole

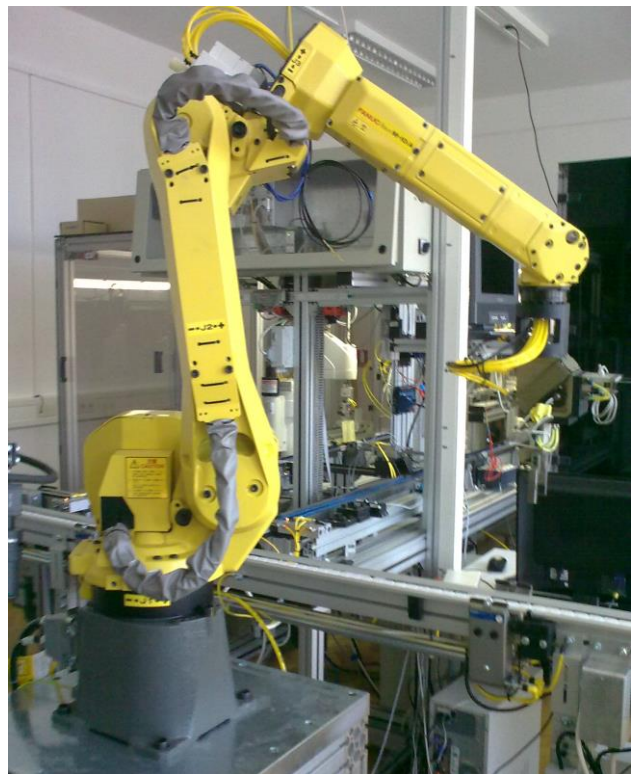
1. Oprema

Oprema: FANUC višeosni upravljački sklopov - FANUC LR Mate 200 iC5L i FANUC M10iA, kamera SONY XC56.

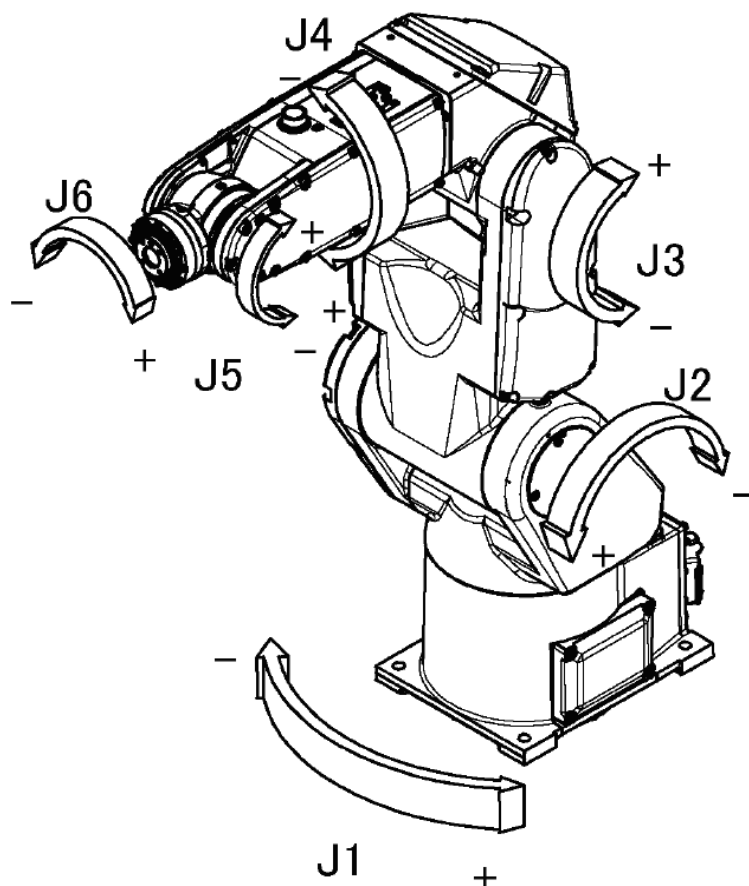
1.1. Karakteristike opreme



FANUC LR Mate 200iC



FANUC M-10iA



Slika 1. Shematski prikaz šest-osnog upravljačkog sklopa (robota) – LR Mate 200iC5L

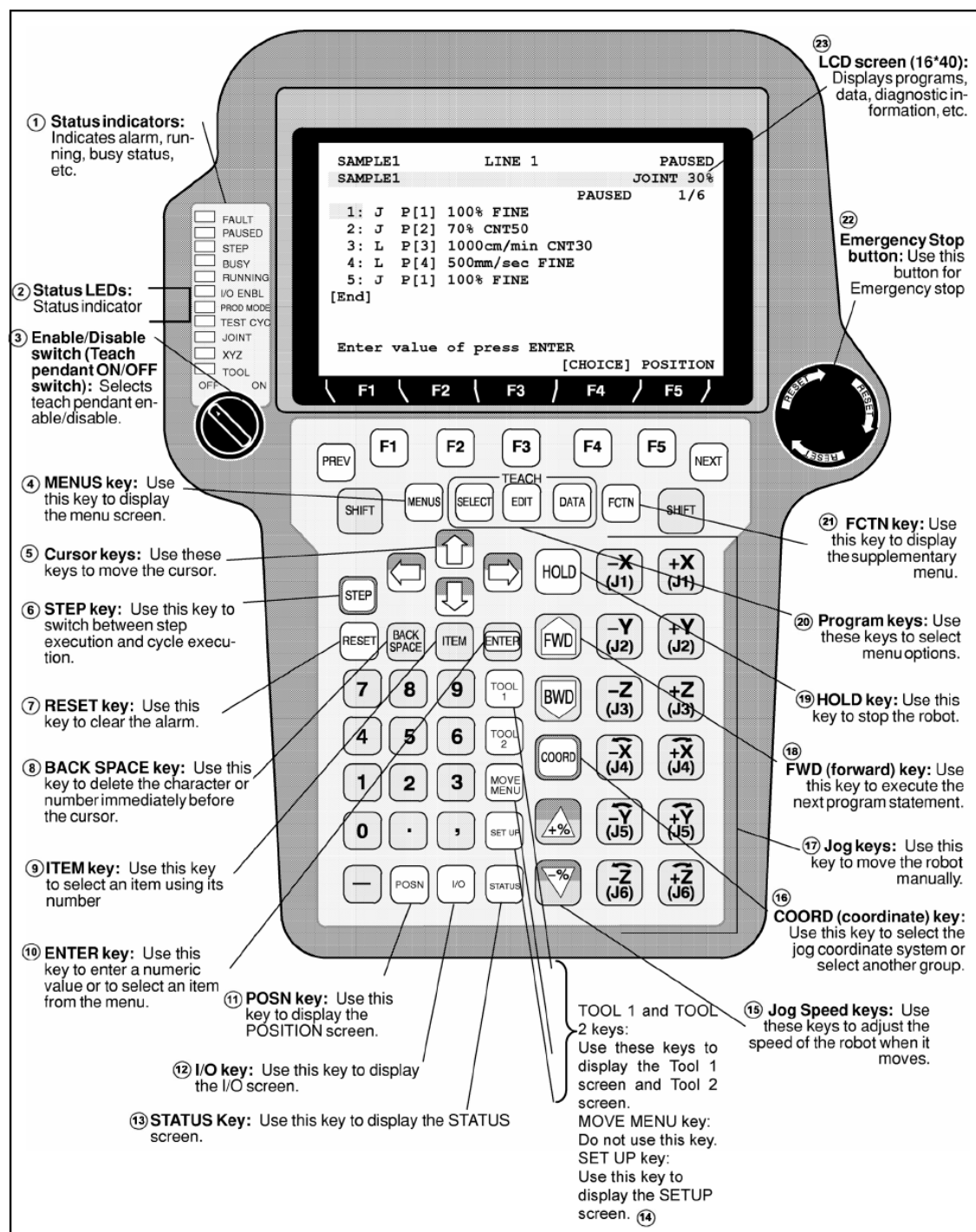
**Osnovne tehničke karakteristike robota
FANUC LR Mate 200iC 5L**

Broj osi		6
Težina		29 kg
Doseg		892 mm
Točnost ponavljanja		± 0.03 mm
Maksimalna nosivost na zglobu		5 kg
Opseg gibanja	J1	340° (5,93 rad)
	J2	230° (4,01 rad)
	J3	373° (6,51 rad)
	J4	380° (6,63 rad)
	J5	240° (4,19 rad)
	J6	720° (12,57 rad)
Maksimalna brzina	J1	270°/s (4,71 rad/s)
	J2	270°/s (4,71 rad/s)
	J3	270°/s (4,71 rad/s)
	J4	450°/s (7,85 rad/s)
	J5	450°/s (7,85 rad/s)
	J6	720°/s (12,57 rad/s)

**Osnovne tehničke karakteristike FANUC
robota M-10iA**

Broj osi		6
Težina		130 kg
Doseg		1420 mm
Točnost ponavljanja		± 0.08 mm
Maksimalna nosivost na zglobu		10 kg
Opseg gibanja	J1	340° (5,93 rad)
	J2	250° (4,36 rad)
	J3	445° (7,76 rad)
	J4	380° (6,63 rad)
	J5	380° (6,63 rad)
	J6	720° (12,57 rad)
Maksimalna brzina	J1	210°/s (3,67 rad/s)
	J2	190°/s (3,32 rad/s)
	J3	210°/s (3,67 rad/s)
	J4	400°/s (6,98 rad/s)
	J5	400°/s (6,98 rad/s)
	J6	600°/s (10,47 rad/s)

1.2. Upravljačka konzola



Prikaz upravljačke konzole sa pojašnjenjem namjene pojedinih tipki

Opis upravljačkih tipki :

1. Indikator stanja (eng. *status indicator*) – označava alarm, u radu, zauzet itd...
2. LED diode stanja – na upravljačkoj konzoli upravljačke jedinice R 30iA Mate ove oznake stanja nalaze se na zaslonu upravljačke konzole
3. Omogućena / onemogućena (eng. *enable - disable*) upravljačka konzola – ON/OFF sklopka

4. Tipka *izbornik* (eng. *menu*) – služi za prikazivanje glavnog izbornika
5. Tipke pokazivača (eng. *cursor*) – služe za pomicanje pokazivača
6. Tipka *korak* (eng. *step*) – služi za promjenu načina rada između koračnog i kontinuiranog načina izvršavanja naredbi
7. Tipka za vraćanje na izvorne postavke (eng. *reset*)
8. Tipka pomak unatrag (eng. *backspace*) – koristi se za brisanje broja ili znaka koji se nalazi prije pokazivača
9. Tipka predmet (eng. *item*) – koristi se za označavanje predmeta koristeći njegov broj
10. Tipka unos (eng. *enter*) – koristi se za upis numeričke vrijednosti ili znaka, označavanje programa, potvrde predmeta sa izbornika, itd...
11. Tipka pozicije (POSN, eng. *position*) – koristi se za prikaz zaslona pozicija robota
12. Tipka ulaz/izlaz (I/O, eng. *input/output*) – koristi se za prikaz zaslona sa ulazno izlaznim signalima robota
13. Tipka stanja (eng. *status*) – Koristi se za prikaz zaslona sa stanjem robota
14. Tipke alata (eng. *tool*) – prikaz zaslona alata
15. Tipka brzine kretanja robota (eng. *jog speed*) – koriste se za određivanje brzine robota prilikom izvođenja naredbi gibanja
16. Tipka koordinatnih sustava (COORD, eng. *coordinate*) – koristi se za izbor koordinatnog sustava robota
17. Tipka za ručno pomicanje robota (eng. *Jog*) – ručno pomicanje robota
18. Tipka naprijed (FWD, eng. *forward*) –koristi se za izvođenje slijedeće naredbe u programu
19. Tipka čekanja (eng. *hold*) – koristi se za zaustavljanje robota
20. Tipke programa (eng. *program keys*) – koriste se za izbor opcija izbornika
21. Tipka funkcije (FCTN) – prikaz dodatnog izbornika
22. Tipka hitnog zaustavljanja (eng. *emergency stop button*) – koristi se za trenutno zaustavljanje rada robota
23. LCD ekran upravljačke konzole

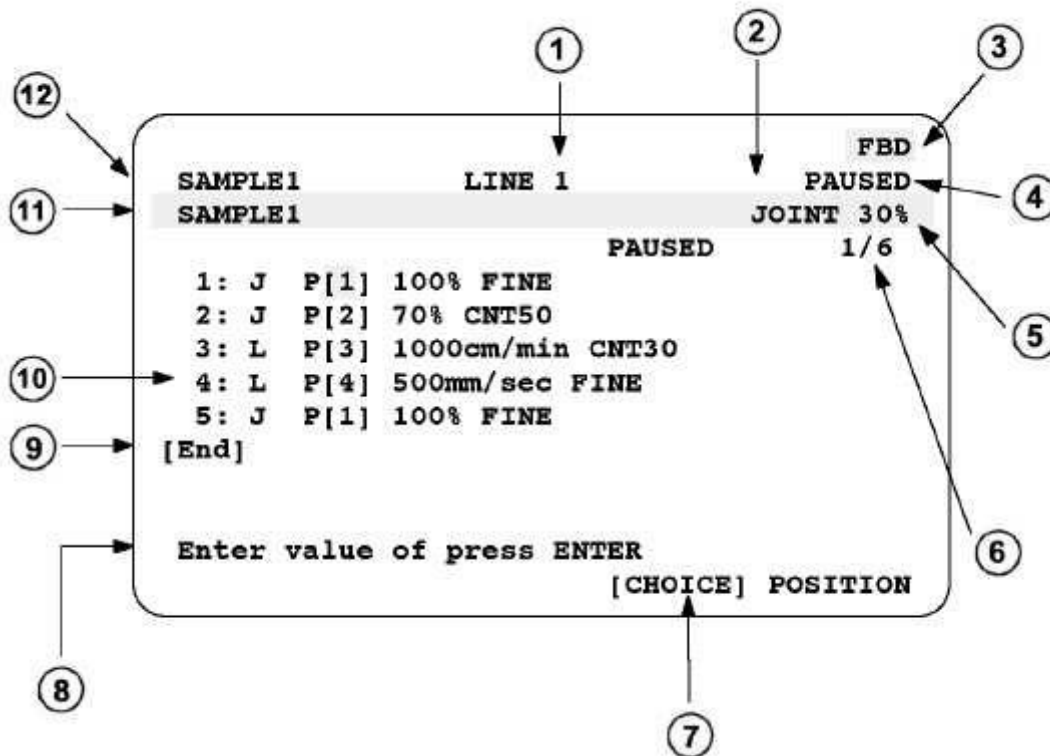
1.3. Upoznavanje sa opremom

- Roboti - FANUC LR Mate 200iC5L / FANUC M10iA
- Upravljačka jedinica R30iA / R30iA Mate– non-Windows operacijski sustav
- Upravljačka konzola – *iPendant* – osnovne funkcije:
 - Ručno pokretanje robota, učenje pozicija, upravljanje signalima
 - Pisanje programa
 - Pokretanje programa
 - ...

2. Osnove rada sa robotom

- **Upravljačka jedinica** →ON/OFF,
 - E_STOP(*emergency stop*) - gljiva
 - način rada:
 - T1 - maksimalna brzina do 250 mm/s
 - T2 - maksimalna brzina
 - AUTO - maksimalna brzina, automatski način rada
- **Upravljačka konzola** →ON/OFF, E_STOP, reguliranje brzine u postotku – *speed override %*
- *Deadman switch (DS)* – tri položaja – samo srednji položaj omogućava ručni rad
- Koordinatni sustavi robota – World (preddefinirani i nepromijenjivi), User (korisnički – 9 mogućih) **Ad I**)
- Koordinatni sustav alata – preddefinirani i korisnički (max 10 mogućih)
 - Koristi za definiranje točke središta alata (TCP – *tool centre point*) **Ad II**)
- Registri – služe za spremanje različitih vrsta podataka (broj registara je proširiv)
 - **R[n]** – registar (n = 1 - 200) – svaki registar može sadržavati jedan cjelobrojan ili realan broj
 - **PR [n]** – pozicijski registar (n = 1 - 100) – svaki pozicijski registar sadrži šest koordinata – translacije po x, y, z osi te redom rotacije oko tih triju osi w, p, r
 - **VR [n]** – vizijski registar (n = 1 - 10) – svaki vizijski registar sadrži šest koordinata – translacije po x, y, z osi te redom rotacije oko tih triju osi w, p, r

2.1. Zaslon



- 1 Aktivan programski red;
- 2 Aktivan koordinatni sustav u ručnom režimu rada;
- 3 Oznaka za onemogućeno izvođenje TP programa nazad (BWD);
- 4 ABORTED, PAUSED, RUNNING Status;
- 5 Brzina rada (OVERRIDE SPEED);
- 6 Trenutni programski red / broj redova;
- 7 Funkcijski meni;
- 8 Poruke za operatera;
- 9 Simbol mjesta programa;
- 10 Broj programskog reda;
- 11 Program, koji se uređuje;
- 12 Program, koji se izvršava.

- Upravljanje prikazom na ekranu – SHIFT + DISP
 - Prikaz 1,2 ili 3 prozora – samo se lijevi prozor može koristiti za pokretanje programa

Busy	Ready	Standby	Fault	TPIF-138 TP: 192.168.123.21 diagnostic 1	3%
Prog	I/O	Prod		AAA LINE 0 T2 ABORTED WORLD	
Select					
611254 bytes free 3/211					
No.	Program name	Comment			
1	-BCKEDT-	[]		
2	A	[]		
3	AAA	[]		
4	AAA_AAA	MR	[]	
5	AAA_BB	[]		
6	AAA_FRM	PC	[]	
7	AAA_FRM8	PC	[]	
8	AAA_FRM9	PC	[]	
9	AAA_LINK	[]		
10	AAA_PAL	[]		

1 prozor

Busy	Ready	Standby	Fault	TPIF-138 TP: 192.168.123.21 diagnostic 1	3%
Prog	I/O	Prod		AAA LINE 0 T2 ABORTED WORLD	
Select					
611122 bytes free 3/211					
No.	Program name	Comment			
1	-BCKEDT-	[]		
2	A	[]		
3	AAA	[]		
4	AAA_AAA	MR	[]	
5	AAA_BB	[]		
6	AAA_FRM	PC	[]	
7	AAA_FRM8	PC	[]	
8	AAA_FRM9	PC	[]	
9	AAA_LINK	[]		
10	AAA_PAL	[]		
11	AAA_PAD	[]		
12	AAA_BOB	[]		
13	AAA_ROT	[]		
14	AAA_PSR	[]		
15	AAA_SS2	[]		
16	AAA_SSS	[]		
17	AAROT	[]		
18	AA_TEST	[]		
19	AA_ZZ	[]		
20	ABC	[]		

2 prozora

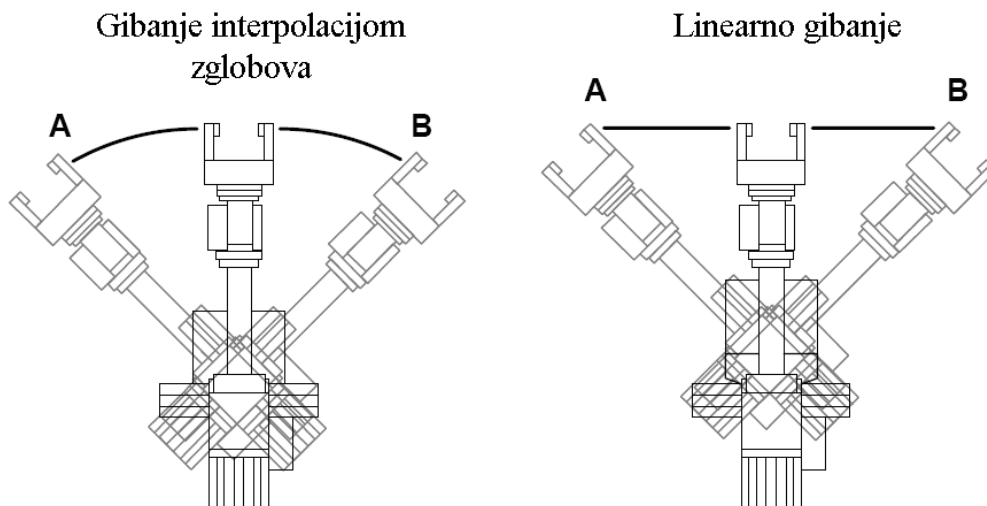
2.2. Ručno pokretanje robota (jog):

- Izbor koordinatnog sustava za ručno pokretanje robota – COORD – TOOL, JOINT, USER, WORLD
- *Deadman switch* (DS) + SHIFT + RESET (DS i SHIFT potrebno držati) + JOG KEYS



2.3. Osnove gibanja

Gibanje robota tj. gibanje središta alata može biti linearno, kružno ili nepravilnog oblika.



Vrste gibanja središta alata robota

2.3.1. Interpolacija gibanja zglobovima

Prilikom interpolacije gibanja zglobovima sve osi robota pokreću se u isto vrijeme te sve zajedno počinju usporavati svoje gibanje. Dobivena trajektorija nije jednostavnog geometrijskog oblika. Gibanje se definira izračunavanjem najdužeg vremena koje je potrebno jednoj od šest osi da napravi zadano gibanje definiranom brzinom. Ova os se naziva i ograničavajuća os (eng. *limiting axis*). Tada se brzina ostalih osi podešava prema najsporoj te se gibanje izvršava.

Sintaksa: J P[1] 50% FINE

Vrh središta alata giba se do točke P[1] 50% od maksimalne brzine robota. Ključna riječ FINE predstavlja način izvršavanja gibanja.

2.3.2. Linearna interpolacija gibanja

Slijedeća pravila vrijede prilikom gibanja središta alata linearnom interpolacijom:

- Središte alata giba se po ravnoj liniji od početne do završne točke definiranom brzinom
- Orijentacija alata vrlo blago se mijenja od početne do završne točke

Sintaksa: L P[2] 200mm/sec CNT 100

Vrh središta alata giba se do točke P[2] brzinom od 200 mm/s. Ključna riječ CNT predstavlja način izvršavanja gibanja.

2.3.3. Način izvršavanja zadanog gibanja

Vrh središta alata prilikom prolaska kroz definiranu točku može proći točno kroz točku ili ju obići u određenom luku. Ova dva načina gibanja prikazana su na slici.

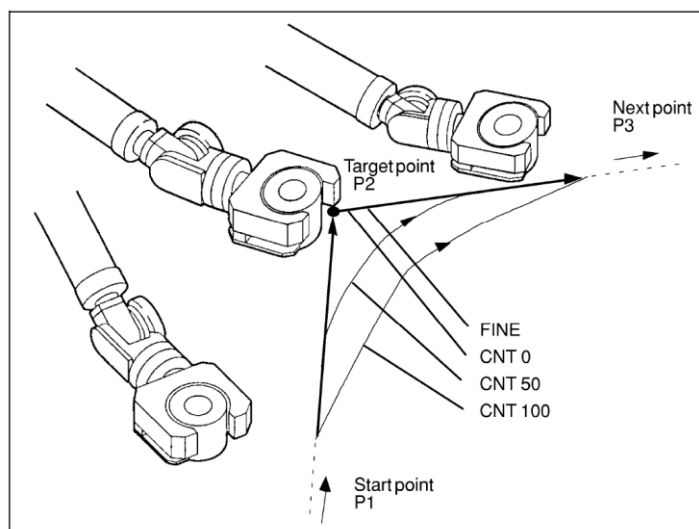


Fig. 4.3.4 Robot motion path using continuous termination type

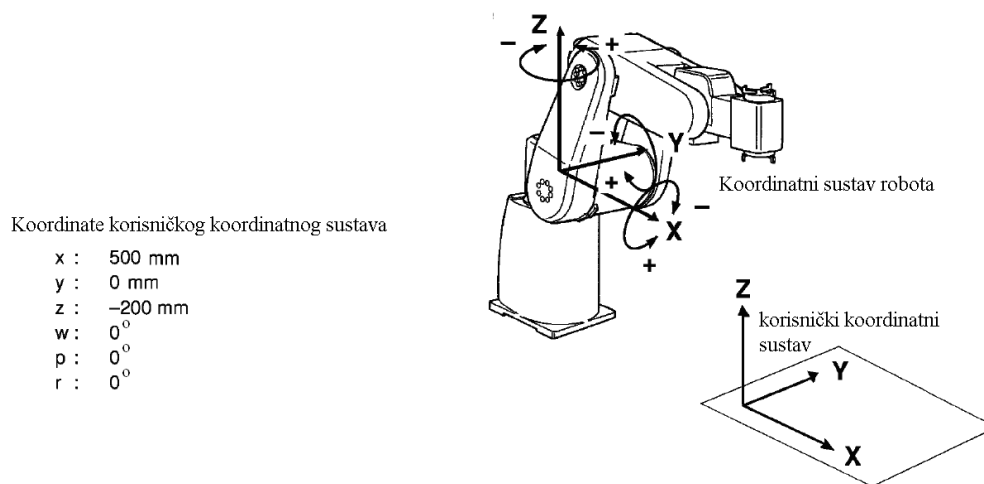
Način izvršavanja zadanog gibanja središta alata

Izvršavanje prekidnog (eng. *Fine*) gibanja dovodi središte alata u svaku definiranu točku te u njoj dolazi do zaustavljanja te kretanja u slijedeću točku. Druga vrsta gibanja je kontinuirano (eng. *Continuous* - *CNT*). Uz ovaj način izvršavanja potrebno je upisati cjelobrojni broj od 0 do 100 gdje veći broj predstavlja veću udaljenost središta alata prilikom njegovog prolaska pored zadane točke. Postavka CNT 100 znači da ne dolazi do deakceleracije vrha alata prilikom prolaska pored zadane točke dok je CNT0 jednak prekidnom (FINE) gibanju.

3. Koordinatni sustavi

3.1. Koordinatni sustav robota

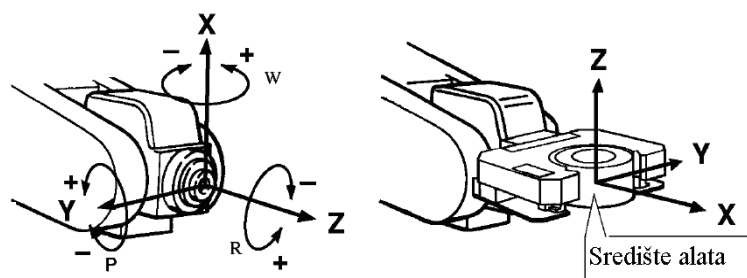
Po standardnim postavkama koordinatni sustav robota preddefiniran je u određenoj točki tijela robota. Ovaj koordinatni sustav je kartezijski, pravokutni, desnokretni koordinatni sustav. Svaki korisnički koordinatni sustav definira se u odnosu na ovaj osnovni koordinatni sustav i to prostornom translacijom po tri osi - x, y i z; te redom rotacijama oko tih triju osi – w, p i r. Moguće je definirati do devet korisničkih koordinatnih sustava. Prikaz koordinatnih sustava prikazan je slikom.



Koordinatni sustav robota, korisnički koordinatni sustav

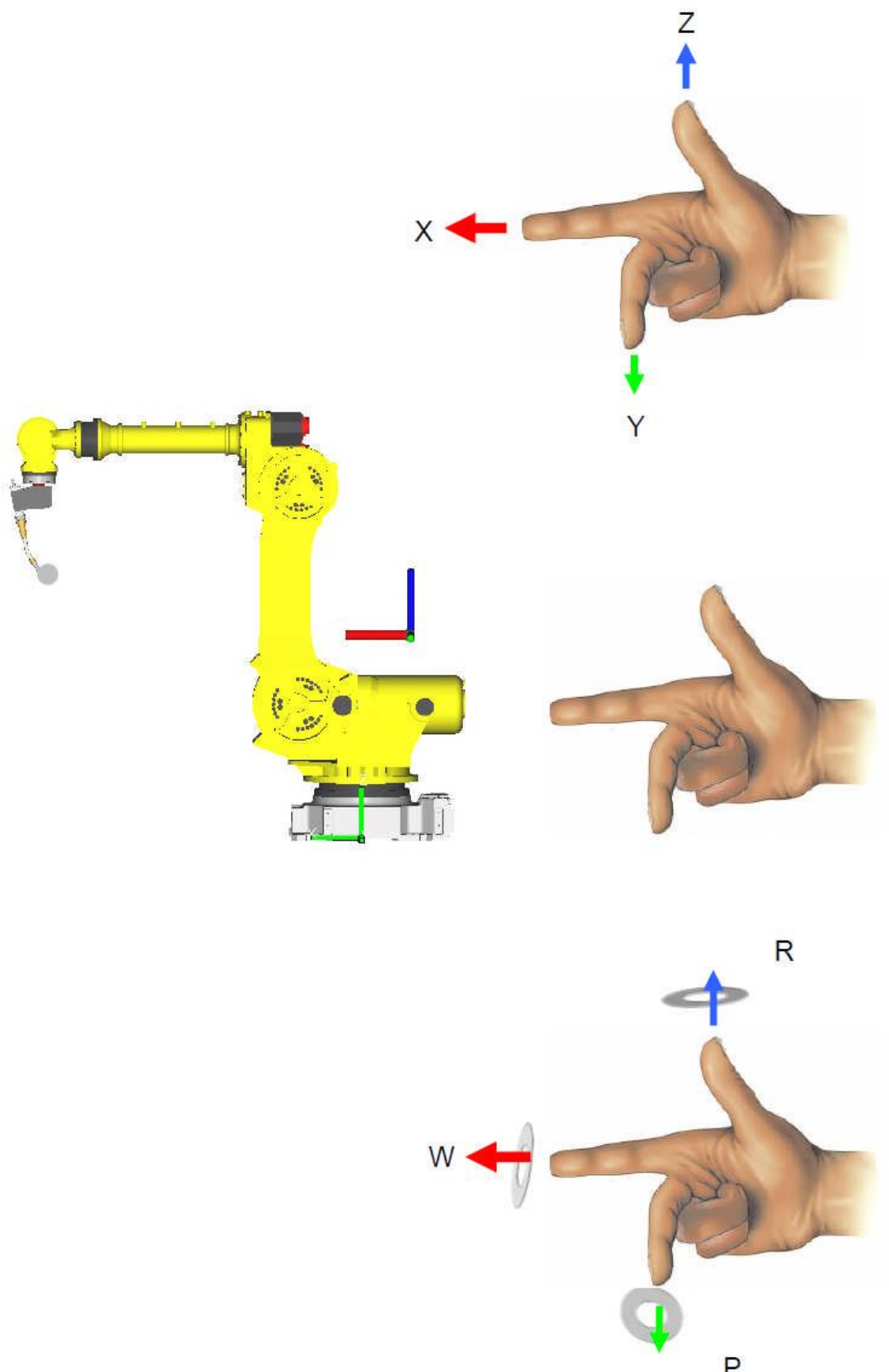
3.2. Koordinatni sustav alata

Ovisno o alatu kojeg robot koristi (npr. različite hvataljke) mora se definirati koordinatni sustav alata kako bi vrh alata uvijek izvršavao željeno gibanje. Analogno preddefiniranom koordinatnom sustavu svijeta robota postoji preddefiniran koordinatni sustav prihvata alata. On se nalazi kako je prikazano na slici u osi šestog zgloba. Prema ovoj točki dalje se definira svaki koordinatni sustav alata. Upravljačka jedinica ima mogućnost definiranja do deset koordinatnih sustava alata. Koordinatni sustav alata definira se pomakom po tri osi i rotacijama od točke prihvata alata.



a) Točka prihvata alata b) Koordinatni sustav središta alata

Pravokutni koordinatni sustav možemo u prostoru jednostavno predočiti pravilom desne ruke.



3.3. Izbor aktivnog koordinatnog sustava

Prije upotrebe moramo odabran koordinatni sustav aktivirati. To možemo napraviti na jedan od načina:

Unutar upravljačkog programa

[INST] -> OFFSET/FRAMES -> UTOOL_NUM=....

Napomena: Poželjno je vršiti pozivanje koordinatnih sustava unutar programa jer tako možemo biti sigurni da je aktiviran koordinatni sustav u kojem su zapisane naše točke.

Prečicom preko COORD tipke

Kombinacijom tipki SHIFT+COORD. U gornjem desnom kutu pojavi se meni. Kursorskim tipkama osvijetlimo željeno polje i upišemo odgovarajući ID broj koordinatnog sustava.

Preko menija za izbor koordinatnih sistema

MENU -> SETUP ->FRAMES ->TYPE -> TOOL FRAME/USER FRAME -> SETIND -> željeni broj

3.4. Offset i tool offset

OFFSET je naredba kojom možemo jednu točku, više točaka ili cjelokupan program pomaknuti bez promjene programa.

```
1:J @P[1] 100% FINE Offset,PR[...]
```

TOOL OFFSET je naredba kojom možemo odrediti odmak pozicije robota od zadane točke i to s obzirom na koordinatni sustav alata.

```
1:J @P[1] 100% FINE  
: Tool_Offset,PR[...]
```


4. Ulazno izlazni signali (I/O)

Upravljačka jedinica ima mogućnost slanja i primanja digitalnih signala, analognih signala, grupnih signala te robotskih signala.

Pomoću digitalnih signala (DI – eng. *digital input* = digitalni ulaz; DO – eng. *digital output* = digitalni izlaz) upravljačka jedinica komunicira sa drugim elementima sustava. Svaki digitalni ulaz ili izlaz može imati dva stanja:

- upaljeno (istina) - *ON* (eng. *true*)
- ugašeno (laž) - *OFF* (eng. *false*)

Uz pomoć ovih signala moguća je jednostavna komunikacija između upravljačke jedinice i uređaja u sustavu. Komunikacija dvije upravljačke jedinice robota pomoću digitalnih signala naziva se rukovanje (eng. *handshaking*).

Uz pomoć robotskih signala RO (eng. *robot output* – robotski izlaz) i RI (eng. *robot input* – robotski ulaz) kontrolira se stanje hvataljke robota. Moguće je definirati grupe digitalnih signala (GO[], GI[]) te koristiti analogne ulaze i izlaze ako postoji ugrađena analogna signalna ploča unutar upravljačke jedinice robota (AI[] i AO[])

I/O statement	1	I/O statement	2
1 DO[]=...		R[]=AI[]	
2 R[]=DI[]		F[]=(...)	
3 RO[]=...		R[]=(...)	
4 R[]=RI[]			
5 GO[]=...			
6 R[]=GI[]			
7 AO[]=...			
8 --next page--		--next page--	

Slika 4.1. Ulazno izlazni signali – naredbe unutar TPP-a

```

1: DO[1]=ON
2: DO[2]=PULSE,1.0sec
3: DO[1]=(DI[1] AND DI[2])

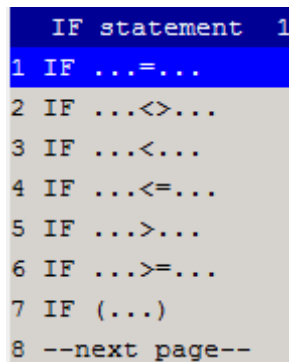
```

Slika 4.2. Primjer sintakse korištenja ulazno izlaznih signala

5. Napredne upravljačke strukture i programski elementi

5.1. Uvjetne IF naredbe

Uvjetnu IF naredbu moguće je dodati iz izbornika INST.



Slika 5.1. IF naredba – INST izbornik

Primjer IF naredbe:

```
1: IF R[1]=0, JMP LBL[10]
2: IF R[2]<>2, JMP LBL[20]
```

Slika 5.2. Primjer sintakse IF naredbe

U izborniku IF naredbe nalazi se i naredba SELECT (u nekim drugim programskim jezicima ovo je CASE naredba) pomoću koje je moguće ispitati stanje određenog signala ili registra te na temelju toga izvršiti skok na dodijeljenu oznaku ili pozvati određeni program.

```
3: SELECT R[3]=1, JMP LBL[1]
4:           =0, JMP LBL[2]
5:           =3, JMP LBL[5]
6:           ELSE, JMP LBL[999]
```

Slika 5.3. Primjer sintakse SELECT naredbe

5.2. Petlje

5.2.1. Beskonačna petlja

Beskonačnu petlju moguće je napraviti dodavanjem oznake (*LBL[x]*) i skoka na oznaku (*JMP LBL[x]*) iz *INST* izbornika.

Instruction 1	
1	Registers
2	I/O
3	IF/SELECT
4	WAIT
5	JMP/LBL
6	CALL
7	Miscellaneous
8	--next page--

Slika 5.4. JMP/LBL naredba

Prikazan je primjer beskonačne petlje unutar koje robot prolazi kontinuiranim gibanjem interpolacijom zglobovima kroz točke P[1] i P[2].

```

1:  LBL[1]
2:
3:J @P[1] 100% CNT100
4:J P[2] 100% CNT100
5:
6:  JMP LBL[1]

```

Slika 5.5. Sintaksa beskonačne petlje

5.2.2. FOR petlja

FOR petlju moguće je napraviti kombinacijom naredbi: *R[]*, *LBL*, *JMP[]* *LBL[]* i *IF*. Prikazan je primjer programa koji izvršava zadano gibanje kroz točke 1 i 2 deset puta te se nakon toga završava.

```

1:  R[1]=0
2:
3:  LBL[1]
4:
5:  R[1]=R[1]+1
6:J @P[1] 100% CNT100
7:J P[2] 100% CNT100
8:  IF R[1]=10,JMP LBL[2]
9:
10:  JMP LBL[1]
11:
12:  LBL[2]

```

Slika 5.6. Sintaksa FOR petlje

Na ovaj način moguće je izraditi korisnički definirane petlje kao i ugniježdene (*Nested*) petlje tj. *petlju u petlji*.

5.3. Poziv drugog programa

Poziv programa moguće je naredbama CALL i RUN iz INST izbornika. Postoji velika razlika između ove dvije vrste poziva te će se unutar ovog seminara koristiti samo naredba *CALL*.

- *Call* – standardni poziv drugog programa gdje se program iz kojeg je izvršen poziv zaustavlja (pauzira) na liniji koda CALL naredbe toliko dugo dok se izvršava pozvani program
- *Run (Multitasking)* – paralelan rad pozvanog programa

5.4. Numerički Registri

Registri služe za pohranu cijelih ili decimalni brojeva. Upravljačka jedinica ima mogućnost pohrane do 200 registara ove vrste. Sintaksa za korištenje broja koji se nalazi u i-tom registru je R[i]. Sa registrima je moguće raditi cijeli niz operacija kji će biti prikazan u daljnjim poglavljima. Prikaz izbornika sa registrima prikazan je na slici .

DATA Registers			1/200
R[1:]	=1.123
R[2:]	=12548
R[3:]	=0
R[4:]	=0
R[5:]	=0
R[6:]	=0
R[7:]	=0

Slika 5.7. Numerički registri

5.5. Izrada brojača

U svrhu brojača poslužiti će numerički registar koji se u određenom dijelu programa inicijalizira u početnu vrijednost. Aritmetičkom operacijom zbrajanja ili oduzimanja mijenja se vrijednost registra tj. brojača. Brojač je prikazan u primjeru FOR petlje.

5.6. Čekanje – Wait

```

1: WAIT 1.00(sec)
2: WAIT DI[1]=ON
3: WAIT R[1]<10
4: WAIT (DI[1] AND DI[2])

```

Slika 5.8. Primjer sintakse naredbe čekanja - Wait

5.7. Pozicijski registri

Pristupanje pozicijskim registrima:

MENU -> NEXT -> DATA -> POSITION REG

Slika 5.9. Pozicijski registri

Važno:

Pozicijski registri koriste su u programima. Promjeni podataka zapisanih u pozicijskim registrima valja pristupiti s oprezom zbog mogućih izmjena upravljačkih programa u kojima se pozivaju isti.

Pozicijske podatke moguće je izmijeniti na dva načina:

1. Opcijom “RECORD” - učitava trenutnu poziciju
2. Ručnim upisom koordinata i definiranje konfiguracije robotske ruke

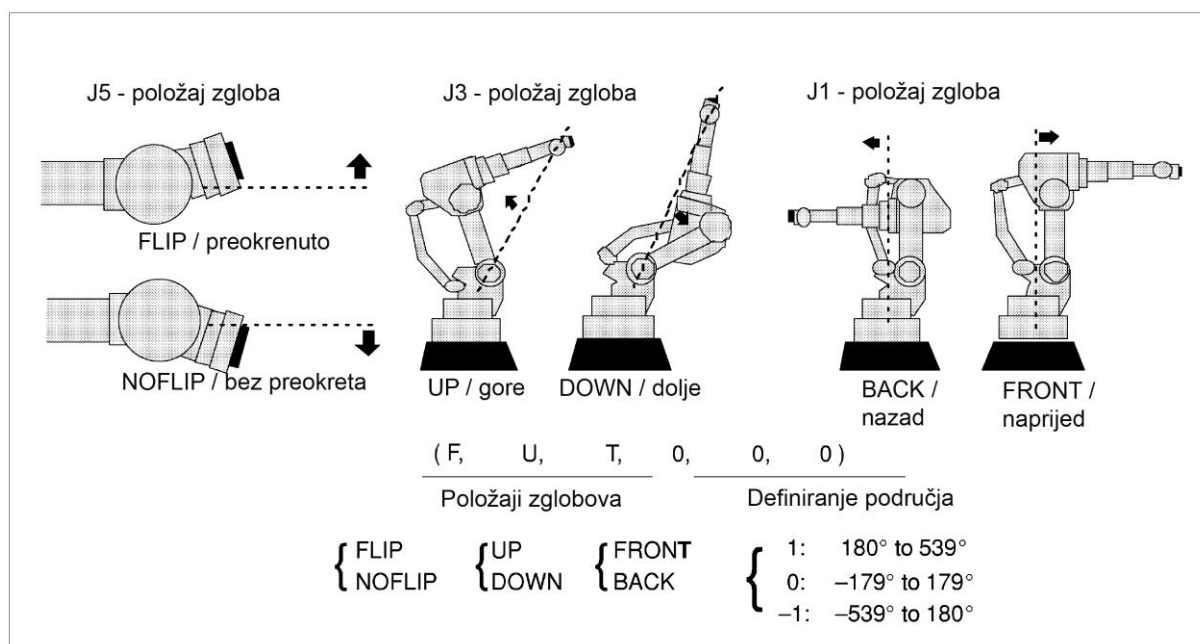
Pozicijski registri mogu biti zapisani na dva različita načina – kartezijski prikaz (x,y,z,w,p,r) ili kao zakreti svakog pojedinog zgloba (J1-J6)

Slika 5.10. Zapisi pozicijskih registara

Primjer korištenja pozicijskog registra unutar upravljačkog programa:

1:J PR[...] 100% FINE

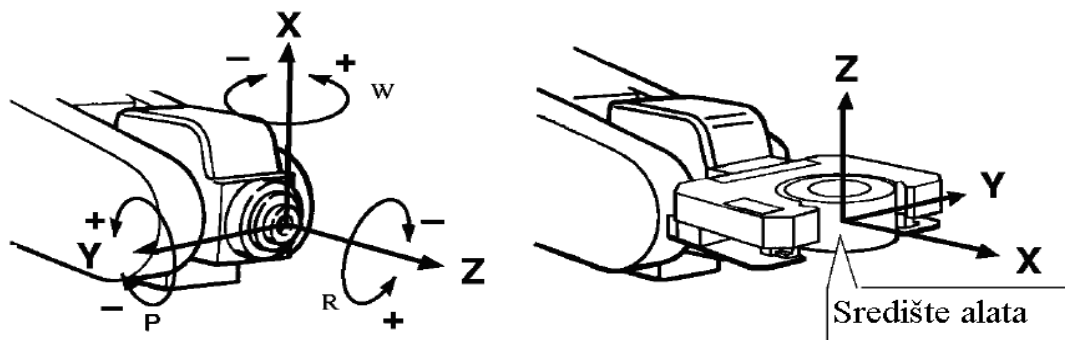
6. Konfiguracije robotske ruke i izrada koordinatnih sustava



Slika 6.1. Konfiguracije robotske ruke

6.1. Izrada korisničkog koordinatnog sustava alata(UTOOL)

Koordinatni sustav alata određuje TCP (Tool Center Point). Koordinatno ishodište je tvornički postavljeno na sredinu pribornice osi 6.



Slika 6.2. Središte alata

Točnost robotske ruke uvelike ovisi o nizu faktora. Neki od njih su:

- tvornička kalibracija robota
- kalibracija koordinatnog sustava alata
- kalibracija korisničkog koordinatnog sustava

Iz tog razloga s ciljem postizanja što veće točnosti robotske ruke potrebno je što je moguće preciznije izvršiti kalibracije korisničkog i koordinatnog sustava alata.

Koordinatnim sustavima pristupamo uz pomoć izbornika MENU -> SETUP -> FRAMES

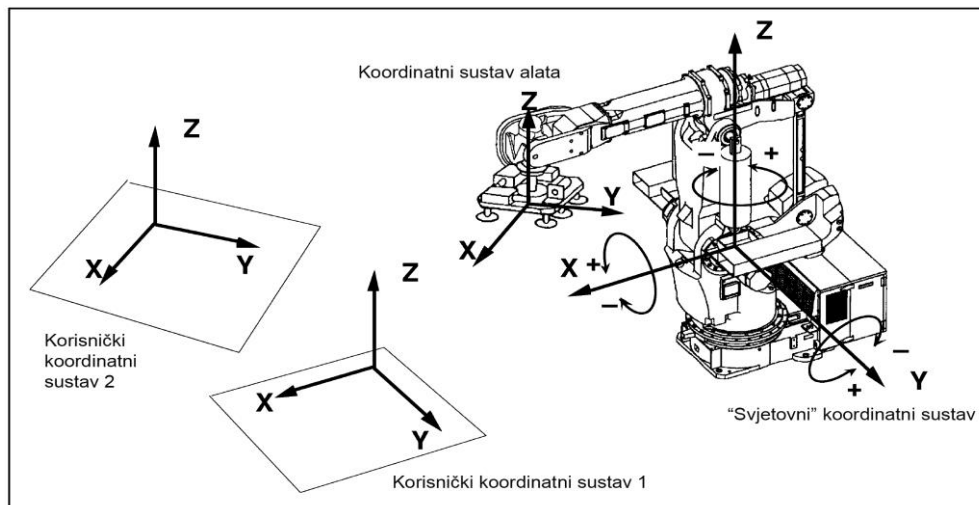
Busy	Stop	Reset	Feed	TEST LINE 0	AUTO	ABORTED	JOINT	100%
Run	I/O	Prod	Stop					
SETUP Frames								
Tool Frame		/ Three Point			1/10			
	X	Y	Z	Comment				
1	0.0	0.0	180.0	[Eoat1]				
2	0.0	0.0	0.0	[Eoat2]				
3	0.0	0.0	0.0	[Eoat3]				
4	0.0	0.0	0.0	[Eoat4]				
5	0.0	0.0	0.0	[Eoat5]				
6	0.0	0.0	0.0	[Eoat6]				
7	0.0	0.0	0.0	[Eoat7]				
8	0.0	0.0	0.0	[Eoat8]				
9	0.0	0.0	0.0	[Eoat9]				
10	0.0	0.0	0.0	[Eoat10]				
Active TOOL \$MNUTOOLNUM[1] = 1								
[TYPE]		DETAIL		[OTHER]		CLEAR		SETIND

Slika 6.3. Izbornik koordinatnih sustava

Metode kalibracije koordinatnog sustava alata

1. Metoda tri točke;
2. Metoda šest točaka;
3. Direktan unos.

6.2. Korisnički koordinatni sustavi



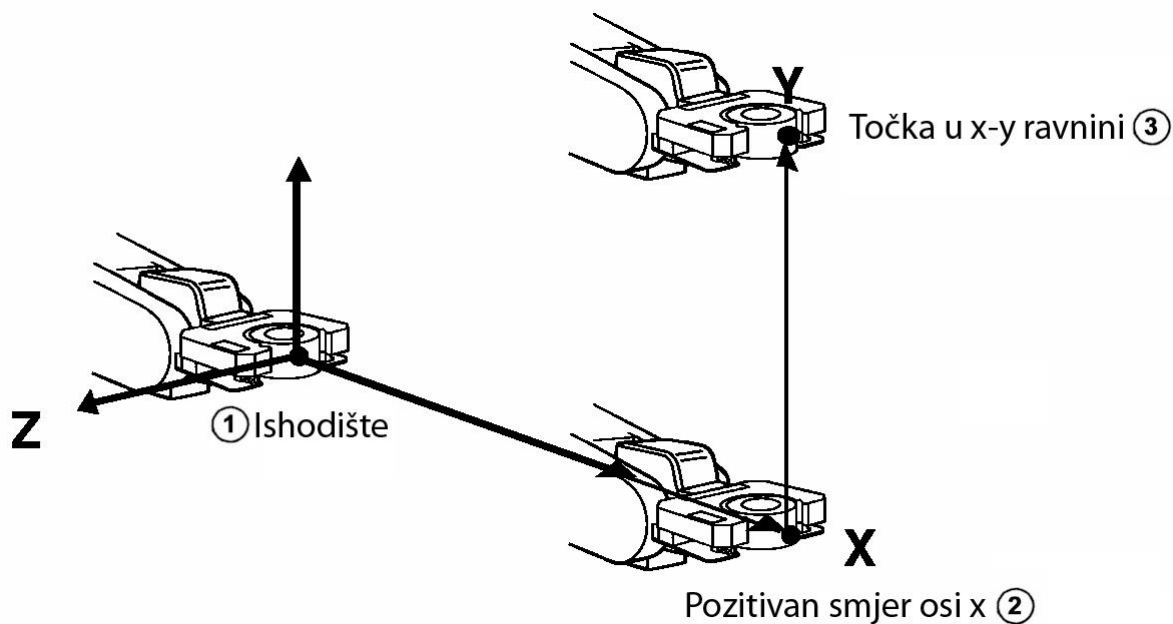
Slika 6.4. Koordinatni sustavi

Metode kalibracije korisničkog koordinatnog sustava

1. Metoda tri točke
2. Metoda četiri točke
3. Direktan unos.

6.2.1. Metoda tri točke

Potrebno je definirati: (1) Ishodište, (2) Pozitivan smjer osi x i (3) Točku u x-y ravnini.

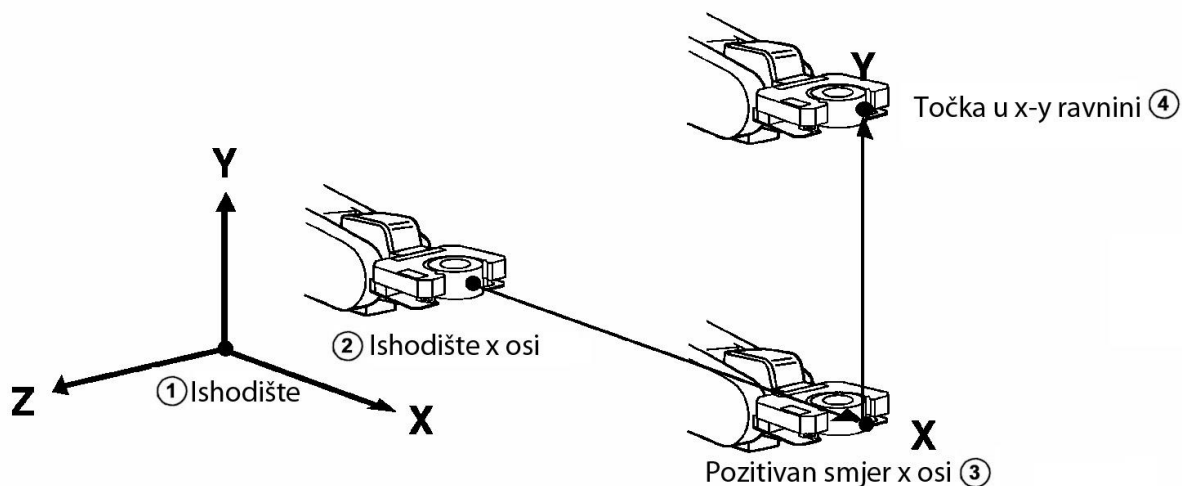


Slika 6.5. Metoda tri točke

6.2.2. Metoda četiri točke

Potrebno je definirati četiri točke:

(1) Ishodište koordinatnog sustava, (2) ishodište za određivanje pozitivnog smjera x osi, (3) pozitivan smjer x osi i (4) Točku u x-y ravnini (s obzirom na točku 2)



Slika 6.6. Metoda četiri točke

6.3. Izbor aktivnog koordinatnog sustava

Prije upotrebe moramo odabran koordinatni sustav aktivirati. To možemo napraviti na jedan od načina:

Unutar upravljačkog programa

[INST] -> OFFSET/FRAMES -> UTOOL_NUM=....

Napomena: Poželjno je vršiti pozivanje koordinatnih sustava unutar programa jer tako možemo biti sigurni da je aktiviran koordinatni sustav u kojem su zapisane naše točke.

Prečicom preko COORD tipke

Kombinacijom tipki SHIFT+COORD. U gornjem desnom kutu pojavi se meni. Kursorskim tipkama osvijetlimo željeno polje i upišemo odgovarajući ID broj koordinatnog sustava.

Preko menija za izbor koordinatnih sistema

MENU -> SETUP ->FRAMES ->TYPE -> TOOL FRAME/USER FRAME -> SETIND -> željeni broj

6.4. Offset i tool offset

OFFSET je naredba kojom možemo jednu točku, više točaka ili cjelokupan program pomaknuti bez promjene programa.

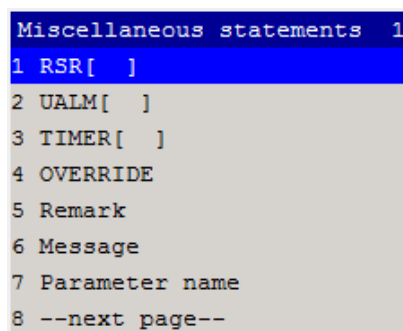
```
1:J @P[1] 100% FINE Offset,PR[...]
```

TOOL OFFSET je naredba kojom možemo odrediti odmak pozicije robota od zadane točke i to s obzirom na koordinatni sustav alata.

```
1:J @P[1] 100% FINE
: Tool_Offset,PR[...]
```

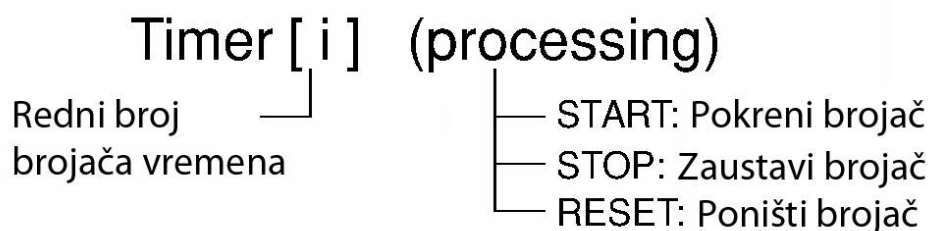
7. Razno

[INST] -> Miscellaneous



Slika 7.1. Miscellaneous izbornik

7.1. Brojač vremena (Timer)



Primjer

```
1: TIMER [1]=START
   TIMER [1]=STOP
   TIMER [1]=RESET
```

Slika 7.2. Brojač vremena

Vrijednost brojača može se koristiti u programu uz pomoć registara.

Primjer: R[1]=TIMER[1]

7.2. Override

$$\text{OVERRIDE} = (\text{vrijednost}) \%$$

$$+ R[i]$$

$$+ \text{Const}$$

$$+ AR[]$$

Slika 7.3. Faktor brzine

Naredba mijenja faktor brzine

7.3. Komentari (Remark)

!komentar

Uz pomoć funkcije remark moguće je dodavati komentare u program. Komentari nemaju nikakvog utjecaja na izvođenje programa. Komentari mogu sadržavati do 32 znaka uključujući alfanumeričke znakove (*,_,@, itd.). Za uređivanje komentara potrebno je pritisnuti tipku ENTER.

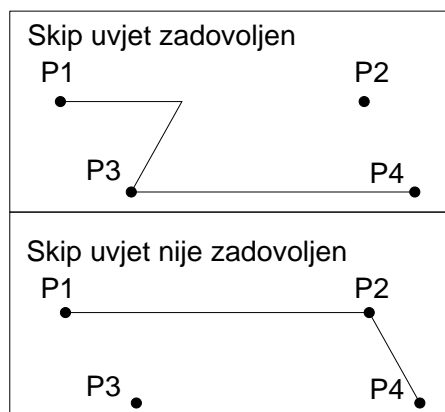
8. Skip condition

Naredba SKIP izvršava preskakanje na definiranu oznaku (LBL) u slučaju da definirani uvjet nije ispunjen. Ako je uvjet ispunjen naredba SKIP zaustavlja trenutno kretanje prema zadanoj točki i izvršava programsku naredbu u sljedećem redu. Ako uvjet nije ispunjen skok na oznaku se izvršava nakon završetka trenutnog kretanja.

```

1: SKIP CONDITION DI[ R[1] ] <> ON
2: J P[1] 100% FINE
3: L P[2] 1000mm/sec FINE Skip, LBL[1]
4: J P[3] 50% FINE
5: LBL[1]
6: J P[4] 50% FINE

```



Slika 8.1. Skip uvjet

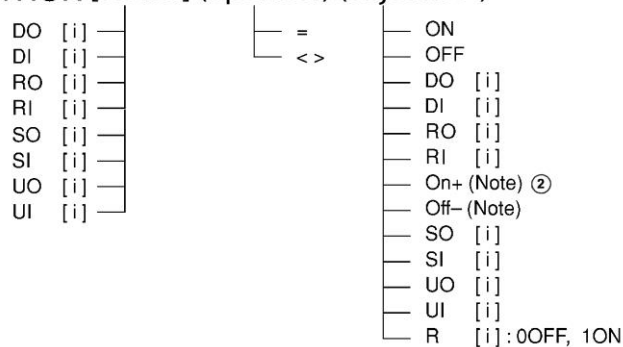
SKIP CONDITION [vrijednost] (operator) (vrijednost)



SKIP CONDITION [varijabla] (operator) (vrijednost)



SKIP CONDITION [stavka] (operator) (vrijednost)



Slika 8.2. Moguće postavke skip uvjeta

Zadatak 1

Naziv zadatka u upravljačkoj jedinici robota – ROBOTIKA_1

Izraditi jednostavan „pick & place“ program, te izraditi "macro" rutine za otvaranje i zatvaranje hvataljke koje će se pozivati u glavnom programu.

SELECT → F2(Create) → Upis imena programa → ENTER

Spremanje trenutne pozicije → F1(Point)

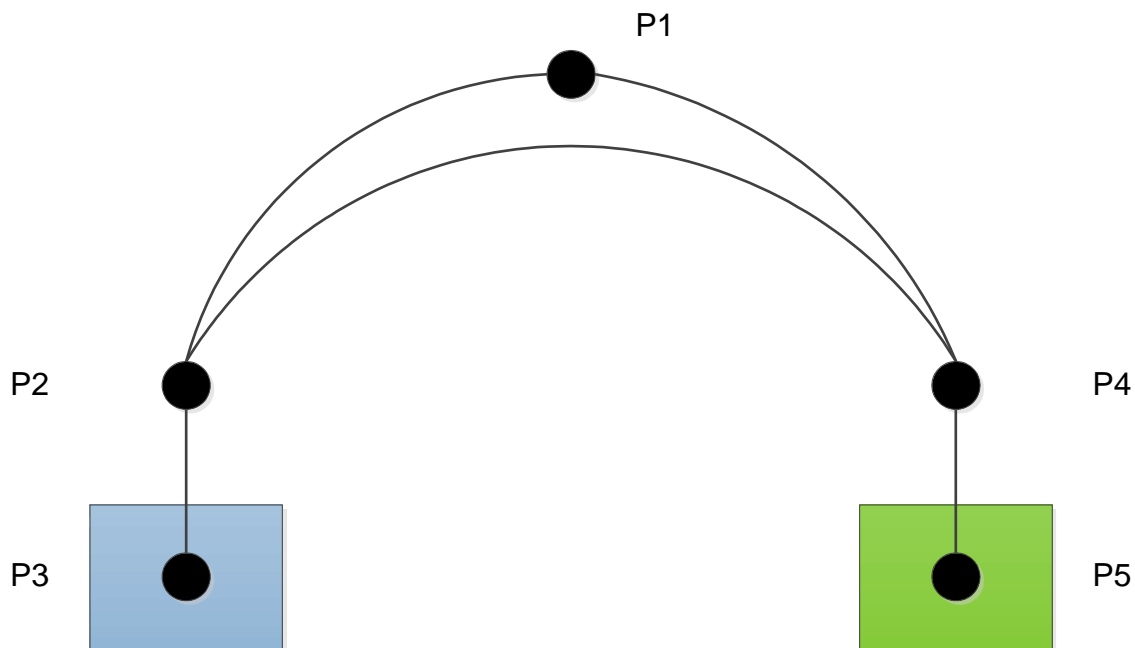
@P[x] - oznaka da se robot nalazi u poziciji x

Faze izrade jednostavnog programa orijentiranog gibanju

- Napraviti plan gibanja robota.
- Dovedi robota u planirane točke, zapamtiti ih te paralelno pisati program

Programski kod gibanja čija je shema prikazana na slici:

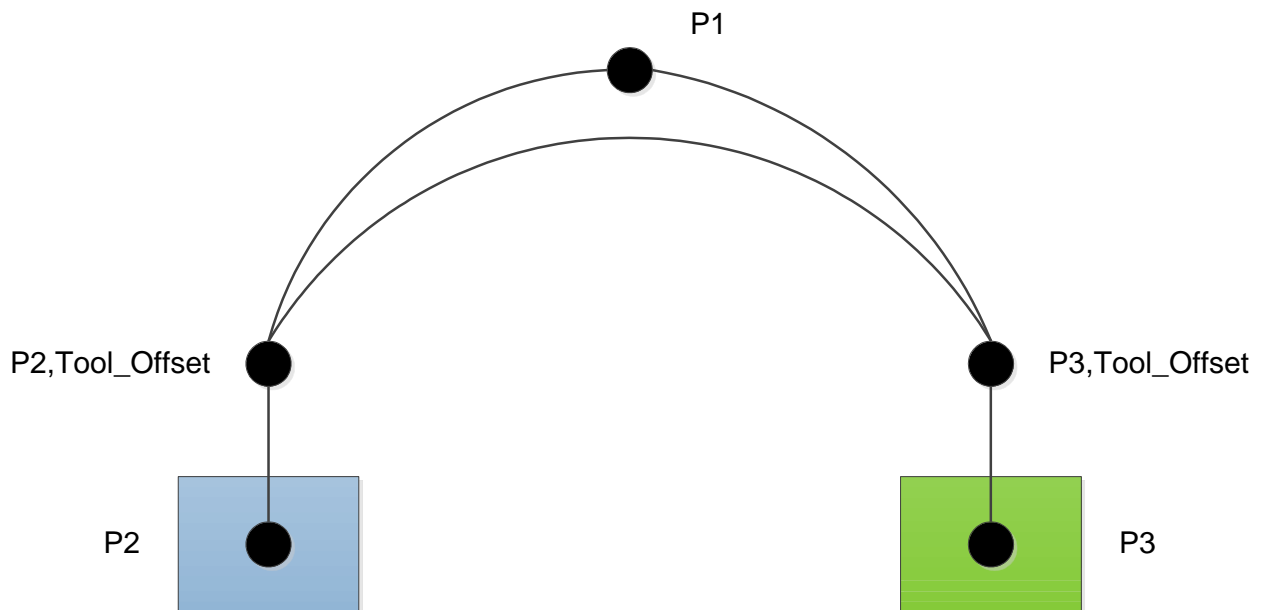
1:	UFRAME_NUM=0	Definiranje korisničkog koordinatnog sustava
2:	UTOOL_NUM=1	Definiranje koordinatnog sustava alata
3:	CALL GR_OP	Poziv programa za otvaranje hvataljke
4:		
5:		
6:	J P[1] 50% FINE	Početna točka u programu
7:		
8:	J P[2] 50% FINE	Točka prilaska
9:	L P[3] 100mm/sec FINE	Točka hvatanja
10:	CALL GR_CL	Poziv programa za zatvaranje hvataljke
11:	L P[2] 100mm/sec FINE	Povratak u točku prilaska
12:		
13:	J P[1] 50% CNT50	Interpolacijska točka (početna točka)
14:		
15:	J P[4] 50% FINE	Točka prilaska odlaganju
16:	L P[5] 100mm/sec FINE	Točka odlaganja
17:	CALL GR_OP	Poziv programa za otvaranje hvataljke
18:	L P[4] 100mm/sec FINE	Povratak u točku prilaska
19:		
20:	J P[1] 50% FINE	Povratak u početnu točku programa



Zadatak 2

Naziv zadatka u upravljačkoj jedinici robota – ROBOTIKA_2

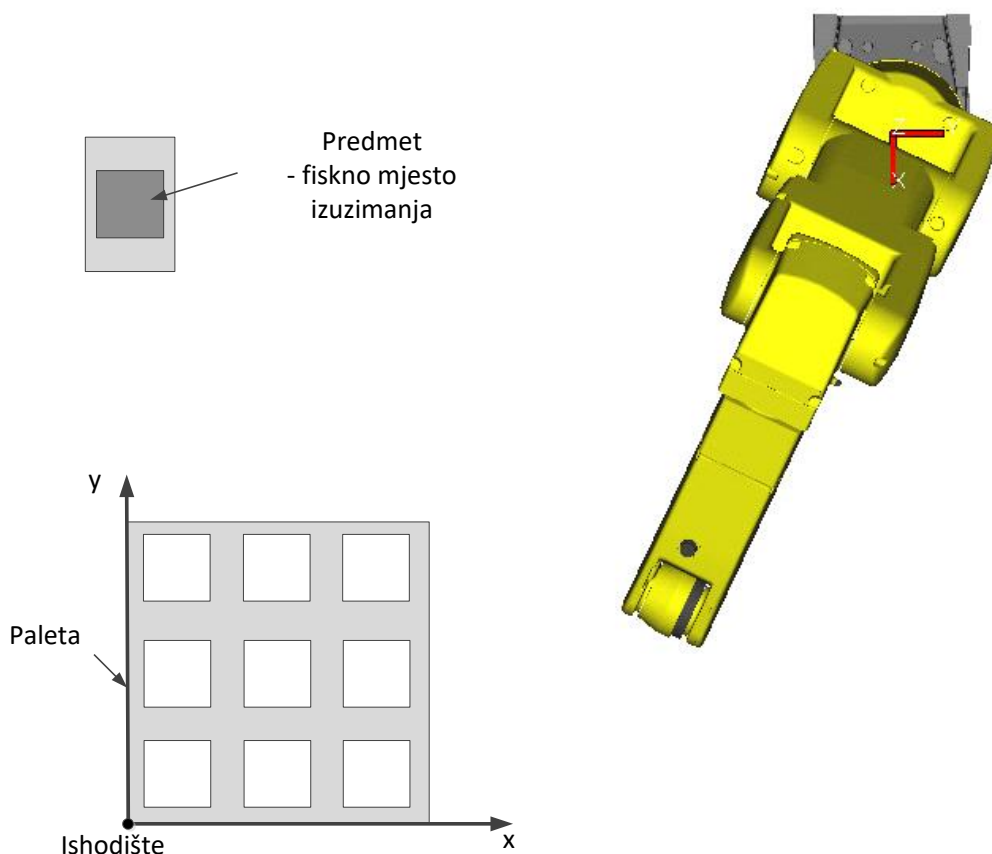
Potrebno je nadograditi osnovni program tako da se pick & place radnja izvrši 5x. Potrebno je svesti program na svega tri točke definirajući prilazne točke naredbom "tool offset"



Zadatak 3

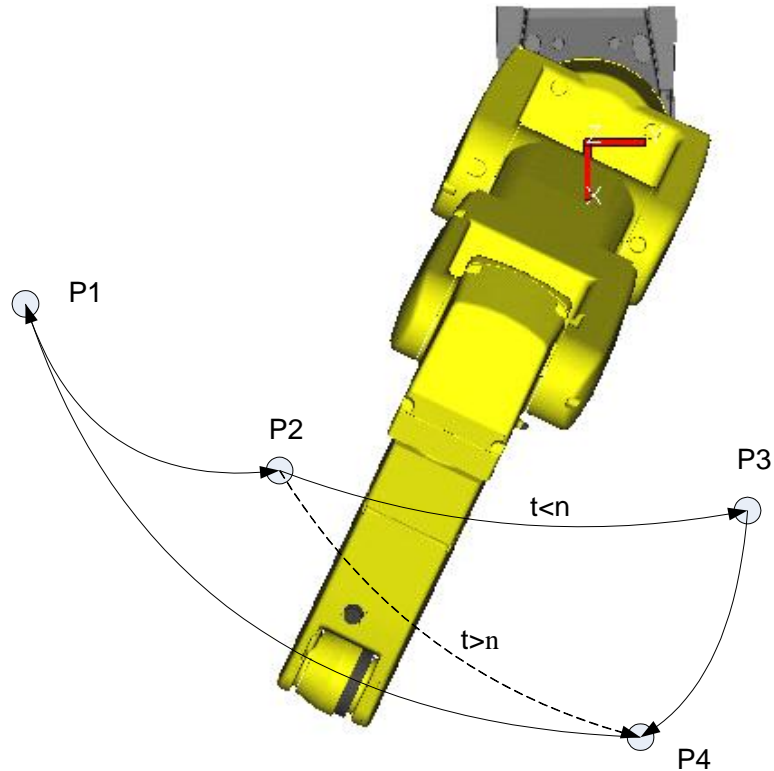
Naziv zadatka u upravljačkoj jedinici robota – ROBOTIKA_3

Koristeći napredne upravljačke strukture i programske elemente (pozicijske registre, registre, petlje i korisničke koordinatne sustave) potrebno je izraditi vlastiti program za paletizaciju dijelova. Predmet se izuzima s istog radnog mjesta i odlaže u paletu. Odrediti korisnički koordinatni sustav palete tako da je rub palete ishodište koordinatnog sustava a njeni bridovi smjerovi koordinatnih osi. Zatim je potrebno u novom definiranom koordinatnom sustavu definirati točke odlaganja predmeta. Kada su definirane točke potrebno je fizički premjestiti kutiju za odlaganje u prostoru i izvršiti rekalkulaciju koordinatnog sustava. Prilazne i izlazne točke definirati naredbom tool offset.



Zadatak 4

Naziv zadatka u upravljačkoj jedinici robota – ROBOTIKA_4



Potrebno je izraditi program koji prolazi kroz nekoliko točaka. Potrebno je u skladu sa vremenom t izvršiti željeno gibanje – u slučaju da je vrijeme $t < n$ robot nakon točke 2 prolazi kroz točku 3 i nastavlja kretanje prema točki 4. U slučaju da je željeno vrijeme izvođenja isteklo robot iz točke 2 odlazi u točku 4. Testirati izvođenje programa pri različitim brzinama koristeći funkciju override.