

Praćenje objekta na pokretnoj traci

Kovač, Ivor Marko

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:940352>

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

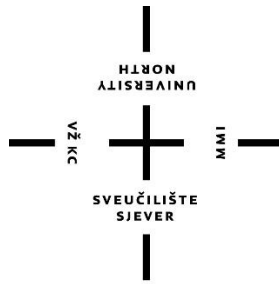
Download date / Datum preuzimanja: **2024-12-23**



Repository / Repozitorij:

[University North Digital Repository](#)





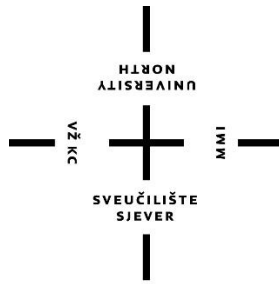
**Sveučilište
Sjever**

Završni rad br. 033/MEH/2024

Praćenje objekta na pokretnoj traci

Ivor Marko Kovač, 0336052980

Varaždin, kolovoz 2024. godine



Sveučilište Sjever

Odjel za Mehatroniku

Završni rad br. 033/MEH/2024

Praćenje objekta na pokretnoj traci

Student

Ivor Marko Kovač, 0336052980

Mentor

Zoran Busija, dipl.ing.stroj.

Varaždin, kolovoz 2024. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za mehatroniku		
STUDIJ	preddiplomski stručni studij Mehatronika		
PRISTUPNIK	Ivor Marko Kovač	IMBAG	0336052980
DAKUM	30.08.2024.	ROLEGIJ	Robotika
NASLOV RADA	Praćenje objekta na pokretnoj traci		
NASLOV RADA NA ENGL. JEZIKU	Object tracking on a conveyor belt		

MENTOR	Zoran Busija, dipl. ing. stroj.	EVANJE	predavač
ČLANOVI POVJERENSTVA	1. Siniša Švoger, mag.ing.mech, predavač		
	2. prof. dr. sc. Ante Čikić		
	3. Zoran Busija, dipl.ing.stroj, predavač		
	4. Josip Srpak, dipl.ing.el, viši predavač		
	5. _____		

Zadatak završnog rada

BR.01	033/MEH/2024
-------	--------------

OPIS

U završnom radu potrebno je:

- oblikovati dijelove i sklopove pokretne trake pomoću 3D CAD programa
- dijelove uvesti u RobotStudio i izraditi robotsko okruženje
- napraviti i opisati simulaciju rada sustava uz pomoć modula "Conveyor tracking"
- analizirati mogućnosti rješenja bez originalnog ABB modula korištenjem RAPID programskog jezika

Ključne riječi: 3D oblikovanje, RobotStudio, conveyor tracking, RAPID

ZADATAK URUČEN 05.09.2024.



Predgovor

Ovaj rad sam izradio koristeći navedene literature i znanjem koje sam stekao tokom svojeg trogodišnjeg studiranja na studiju Sveučilišta Sjever.

Zahvaljujem se svome mentoru Zoranu Busiji dipl.ing.stroj. na uloženom vremenu, trudu i usmjeravanju za vrijeme studija i za vrijeme pisanja završnog rada.

Zahvaljujem se i obitelji koja mi je omogućila i pružala podršku kroz studiranje.

Sažetak

U radu su obrađena dva načina praćenja objekta.

Prvi način je programiranje IRB 1600 industrijskog robota u simulacijskom okruženju programa RobotStudio 2019 na način koji se implementira u praksi. Uz pomoć dodatnih ABB komponenti postiže se praćenje objekta na pokretnoj traci (engl. *conveyor tracking*). Objasnjeno je korak po korak postavljanje, konfiguraciju te programiranje robota, pokretne trake i objekta. Objekt i pokretna traka su modelirani u SOLIDWORKS 2018 programu dok je robot preuzet iz ABB knjižnice. Sve mehaničke komponente obuhvaćene ovim rješenjem nisu dostupne na sveučilištu.

Drugi način je istraživanje o mogućnosti praćenja objekta bez dodatnih ABB komponenti. Uz pomoć dodatne komponente "PLC stol" definirane u RobotStudio programu kao mehanizam, postiže se kretanje objekta. Komponenta je modelirana u programu SOLIDWORKS 2018 prema istom takvom stolu koji se nalazi u laboratoriju sveučilišta. Unutar stola je PLC uređaj koji preko elektromotora upravlja položajem pomične ploče na vrhu stola. Moguće je dovesti ploču u četiri položaja: home , položaj_1, položaj_2 i položaj_3. Ujedno je moguće promijeniti brzinu kretanja pomične plohe. Robot programski upravlja položajem plohe. Sve komponente obuhvaćene ovim rješenjem dostupne su na sveučilištu. Industrijski robot korišten u ovoj izvedbi je IRB 120.

Summary

This paper examines two methods of object tracking.

The first method involves programming the IRB 1600 industrial robot in the simulation environment of RobotStudio 2019 in a way that is implemented in practice. With the help of additional ABB components, object tracking on a conveyor belt is achieved. The setup, configuration, and programming of the robot, conveyor belt, and the object are explained step by step. The object and conveyor belt are modeled in SOLIDWORKS 2018, while the robot is taken from the ABB library. All mechanical components covered by this solution are not available at the university.

The second method involves exploring the possibility of object tracking without additional ABB components. With the help of the additional component "PLC table," defined in the RobotStudio program as a mechanism, object movement is achieved. The component is modeled in SOLIDWORKS 2018, based on a similar table found in the university laboratory. Inside the table is a PLC device that controls the position of the movable plate on top of the table via an electric motor. The plate can be moved to four positions: home, position_1, position_2, and position_3. It is also possible to change the speed of the movable plate. The robot controls the position of the plate with a program. All components covered by this solution are available at the university. The industrial robot used in this solution is IRB 120.

Popis korištenih kratica

PLC	Programmable logic controller – programibilni logički kontroler
IRB	Industrial robot – industrijski robot
ABB	Asea Brown Boveri – švicarsko-švedska tvrtka
WOBJ	Work object – koordinatni sustav definiran unutar programa RobotStudio
MOVEJ	Move jog – pomaknuti nelinearnom kretnjom
MOVEL	Move linear – pomaknuti linearnim kretnjom

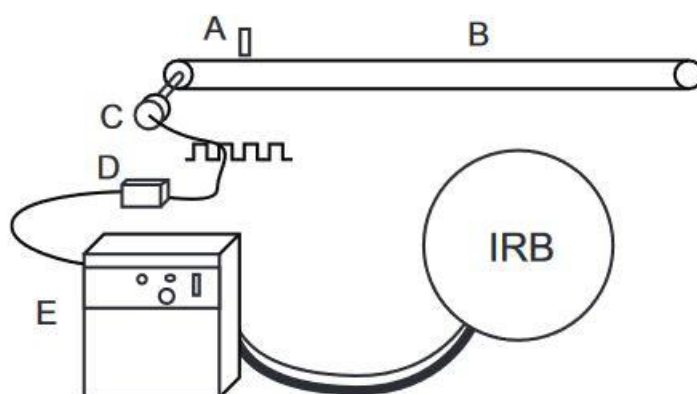
Sadržaj

1.	Uvod.....	1
2.	Praćenje objekta na pokretnoj traci.....	3
2.1.	Postavljanje simulacijske okoline	4
2.2.	Rapid kod	12
3.	Praćenje objekta	14
3.1.	Postavljanje simulacijske okoline	15
3.2.	Rapid kod	20
3.3.	Korištene formule.....	21
3.4.	Mogući način izvedbe kontinuiranog pomaka	22
4.	Analiza rezultata	24
4.1.	Praćenje objekta na pokretnoj traci	24
4.2.	Praćenje objekta	24
5.	Zaključak.....	25
6.	Literatura.....	26

1. Uvod

Rješavanju problema praćenja objekta pristupit će se na dva načina. Prvi način je izvedba praćenja kakvu kao rješenje nudi proizvođač ABB[1]. Drugi je pokušaj postizanja praćenja objekta pomoću alata koje su nam dostupni u sklopu sveučilišta.

Praćenje objekta na pokretnoj traci obuhvaća programiranje ABB robota unutar programa RobotStudio 2019. Ovo rješenje prikazano na slici 1.1 i zahtjeva da ABB-ov robot [IRB] komunicira sa kontrolerom trake [E], enkoderom [C] te njegovim sučeljem [D], pokretnom trakom [B] i prekidačem za sinkronizaciju [A].



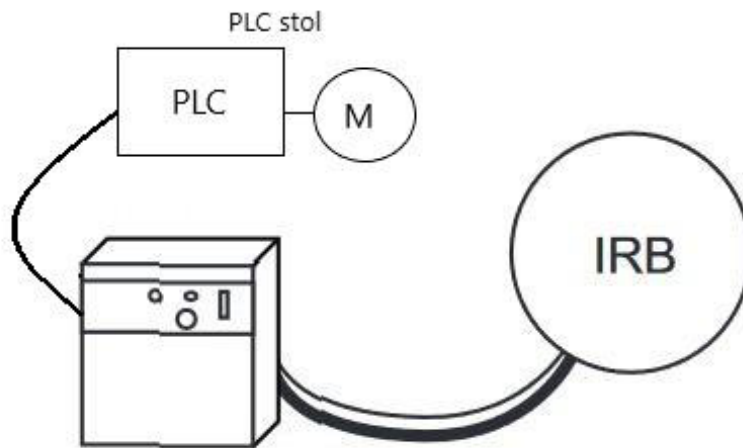
Slika 1.1 ABB paket za praćenje na pokretnoj traci [1]

Kontroler trake upravlja brzinom pokretne trake, dok sučelje enkodera šalje informacije o brzini gibanja trake prema kontroleru robota. Prekidač za sinkronizaciju je senzor koji isto šalje informaciju o položaju objekta prema kontroleru robota. Kontroler robota prema dobivenim informacijama te programu upravlja robotskim manipulatorom. Umjesto senzora može se koristiti i kamera koja omogućuje preciznije praćenje.

Sve navedene komponente moraju biti kupljene od istog proizvođača.

Ukoliko je u ovom sustavu prekidač za sinkronizaciju senzor a ne kamera, sustav prividno prati objekt. Robot ne zna točan položaj objekta, već samo zna da je objekt došao u radni prostor jer je senzor bio aktiviran . Takav sustav će biti obrađen u ovome radu.

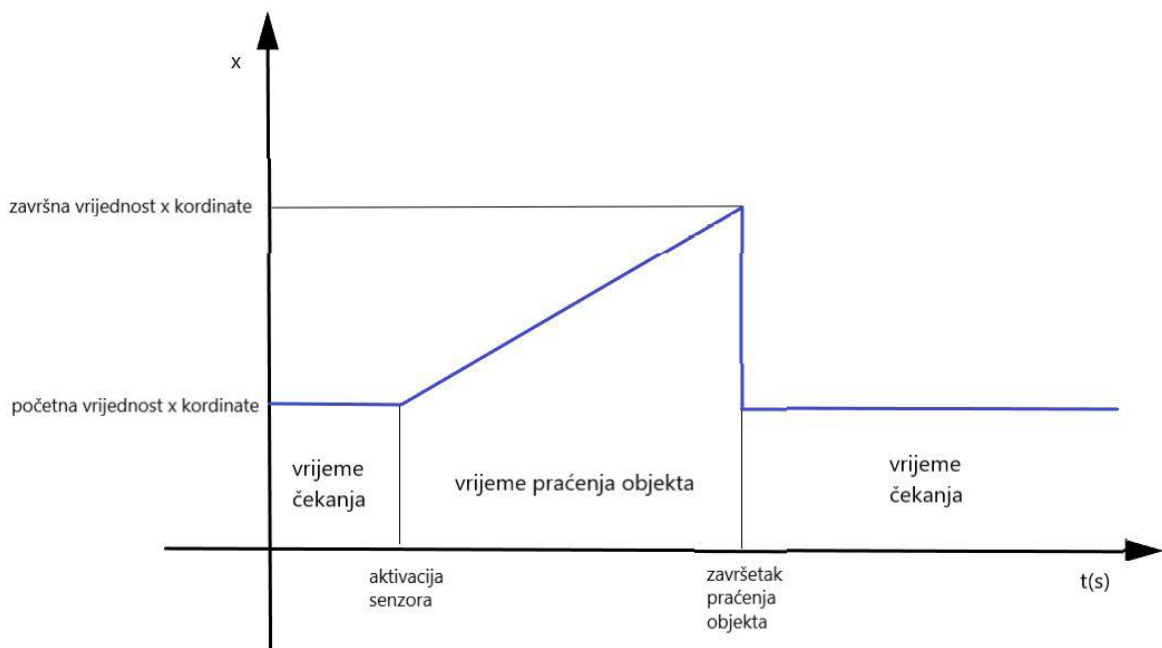
Drugi dio rada obuhvaća programiranje robota za izvršavanje kretnji, te ujedno i upravlja PLC uređajem. Prema programu kontroler šalje signal PLC uređaju kada i u koji položaj da dovede pomičnu ploču PLC stola na kojoj se nalazi praćeni objekt. Postoje dva predodređena položaja: home i krajnji desni položaj. Pozicije imaju granične prekidač koji javlja PLC-u da se ploča nalazi u njoj. PLC ne šalje podatke o položaju ploče kontroleru robota. Mi vjerujemo PLC stolu da je ploča u željenom položaju. Ujedno se može promijeniti brzina kretanja ploče unutar PLC uređaja.



Slika 1.2 Prikaz povezanosti dostupnih komponenti u laboratoriju

2. Praćenje objekta na pokretnoj traci

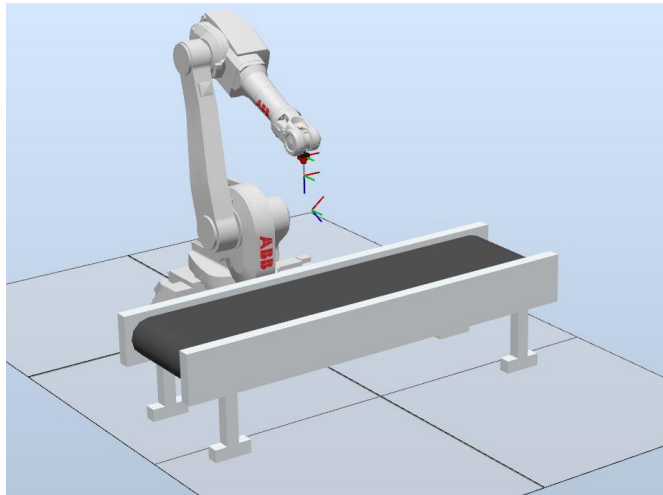
Praćenje objekta na pokretnoj traci radi na principu komunikacije pokretne trake sa robotskim manipulatorom. Robot čeka dolazak objekta u definiranu prostornu kocku koja predstavlja radni prostor u kojem robot može pratiti taj objekt. Aktivacijom senzora prikazano na slici 2.1 pokreće se programirana rutina. Uz pomoć dodatnih komponenti, robot prepoznaje pomak objekta tako što se kontinuirano pomiče kopirani koordinatni sustav (eng. u rapidu *work object*). Osnovni *wobj* u kojem su zadane kretnje robotskog manipulatora ostaje nepromijenjen. Nakon izlaska objekta iz definiranog radnog prostora, kopija *wobj* kojom je zadana koordinata izmijenjena se briše. U slučaju dolaska novih objekata u radni prostor prije izlaska starih, za njih se kreira dodatna kopija *wobj*. Ujedno robot prati pojedine objekte prema redoslijedu njihovih dolazaka u radni prostor. Ulazak više objekata u prostor ne predstavlja problem te nakon odrađenog praćenja prvog objekta kreće praćenje drugog, pa trećeg i tako redom. Praćenje neće biti potpuno odrađeno ako je traka prebrza i objekt izađe iz radnog prostora prije završetka praćenja.



Slika 2.1 Prikaz kontinuiranog pomicanja koordinate *wobj*

2.1. Postavljanje simulacijske okoline

Prvo se u simulacijsku okolinu prikazanoj na slici 2.2 dodaje IRB1600 industrijski robot, te se potom postavlja pokretna traka. Pokretna traka je modelirana unutar SOLIDWORKS 2018 programa i spremljena kao “.SAT“ vrsta datoteke. Uobičajene “*solidworks part*“ datoteke se ne mogu umetnuti u RobotStudio okruženje. Odabran je IRB1600 jer ima povećani radni prostor. Nakon toga odabire se alat koji će se koristiti. Pokretna traka mora biti namještena tako da je unutar radnog prostora robota.



Slika 2.2 Simulacijska okolina

Zatim je potrebno odabrati kontroler koji upravlja industrijskim robotom unutar prozora prikazanog na slici 2.3.

Create Controller From Layout ×

Controller Name and Location
Select the virtual controller location and the RobotWare version to use

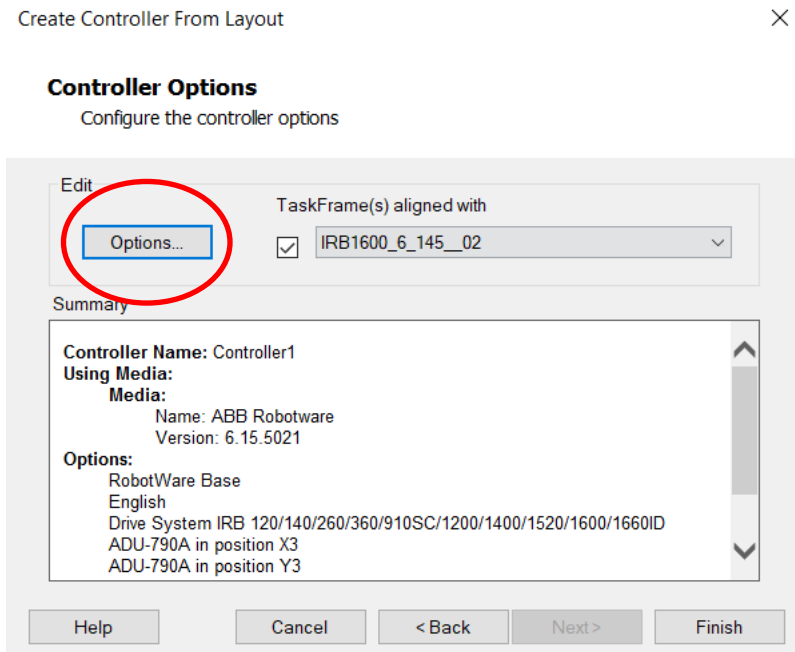
Name

Location

RobotWare: [Locations...](#)

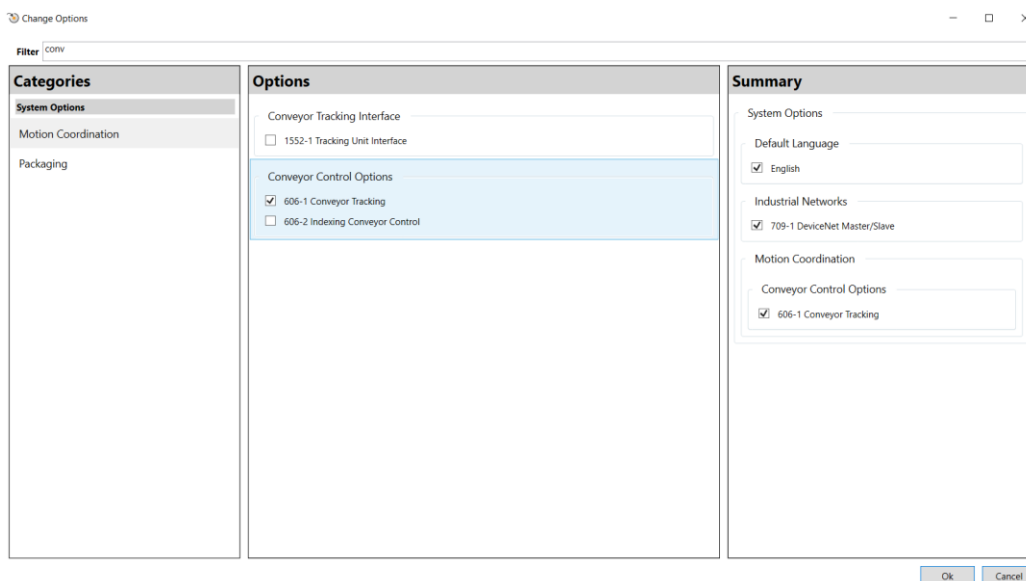
Slika 2.3 Odabir kontrolera

Nakon odabira kontrolera otvara se kartica prikazana na slici 2.4 sa mogućnošću ulaska u njegove postavke.



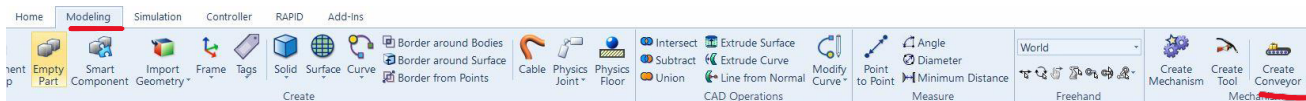
Slika 2.4 Kartica odabira

Unutar postavki prikazanim slikom 2.5 dodaje se modul za praćenje pokretne trake . Nakon odabira “*Conveyor Control Options*“ izlazi prozorčić sa tri opcije u kojoj se odabire “ *DeviceNet Master/Slave*“ dodatak. Taj dodatak omogućava simuliranje enkodera.



Slika 2.5 Dodavanje enkodera

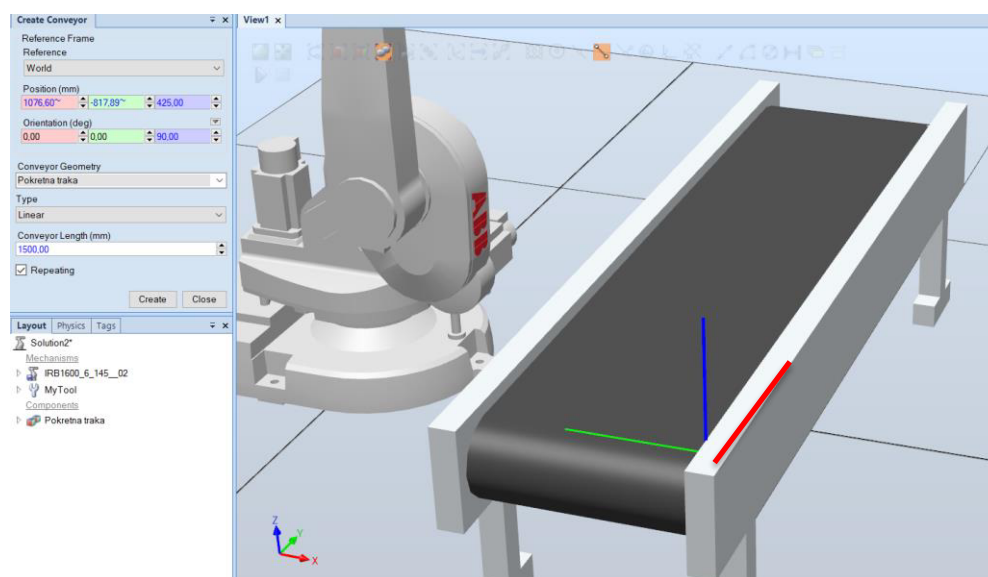
Odlaskom na izbornik “*Modeling*“ prikazano na slici 2.6, u desnom kutu odabire se “*Create Conveyor*“ naredba.



Slika 2.6 Odabir izbornika modeling

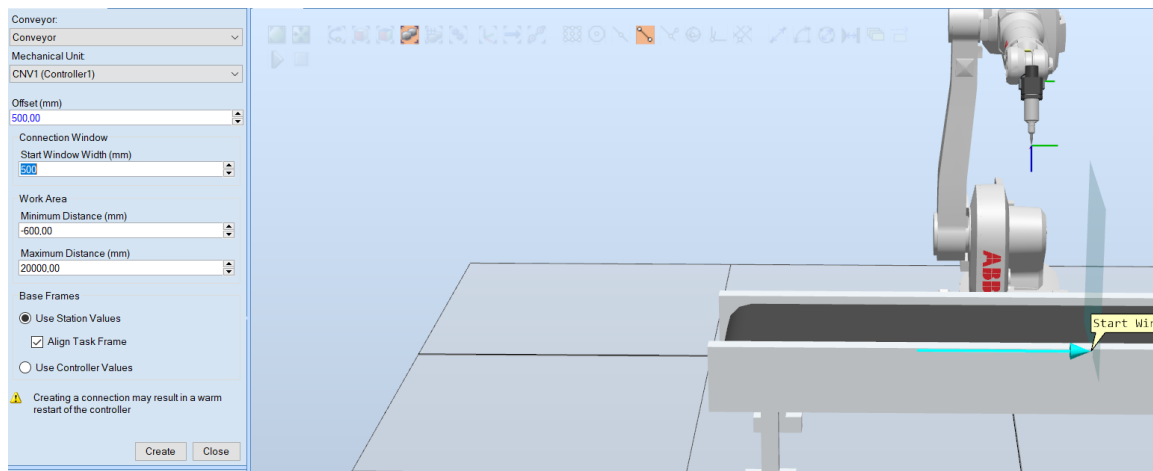
Iskače izbornik koji se koristi za kreiranje pokretne trake. Koordinatni sustav prikazan na slici 2.7 se postavlja na već dodani model pokretne trake. Crvena linija predstavlja x koordinatu koordinatnog sustava koja prikazuje smjer kretanja pokretne trake dok je “*conveyor length*“ parametar koji određuje dužinu pokretne trake.

Ukoliko se uključi naredbu ponavljanje (*eng .repeating*), pokretna traka će sama periodično stvarati odabrani predmet koji će se kretati dužinom pokretne trake.



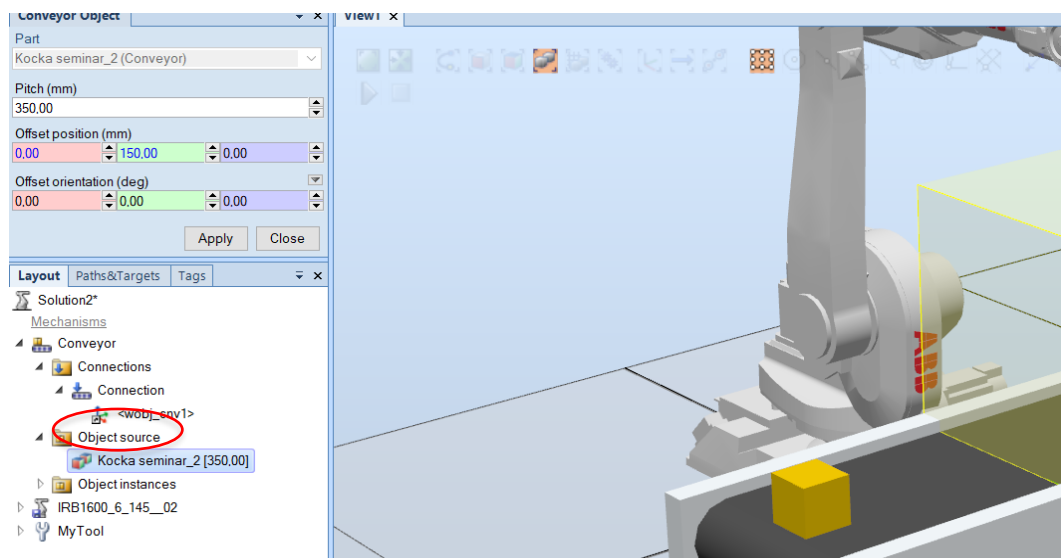
Slika 2.7 Definiranje pokretne trake

Ponovno se odlazi na karticu *MODELING*, pa se u desnom kutu odabire se “*create connection*“. Pod “*Offset*“ parametrom prikazano na slici 2.8, unosi se udaljenost od koordinatnog sustava pokretne trake, do željenog početka radnog prostora. Od mjesta gdje se nalazi senzor započinje praćenje objekta koji se nalazi na pokretnoj traci. *Start Window Width* je parametar koji opisuje širinu radnog prostora (širinu pokretne trake).



Slika 2.8 Definiranje pozicije senzora

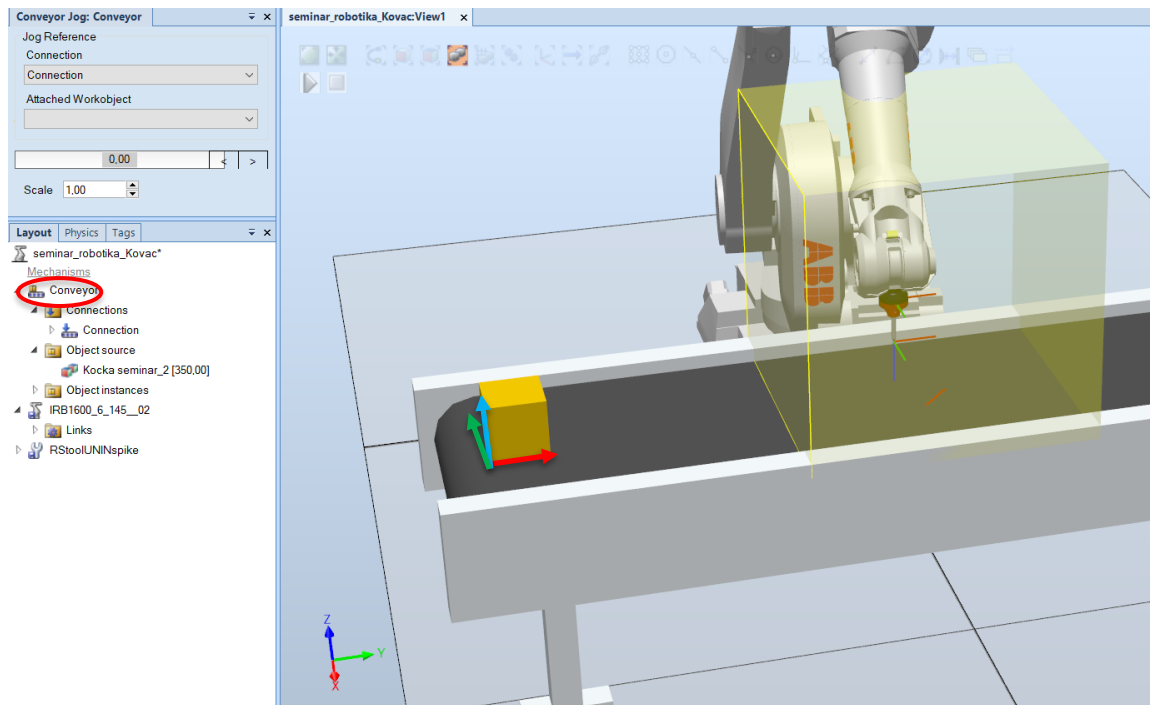
Dodaje se i radni objekt tako da se umetne predmet koji je dizajniran u programu SOLIDWORKS 2018. Kako bi se objekt mogao umetnuti u simulacijski prostor softvera RobotStudio, mora biti spremljen sa nastavkom “.SAT“. Svaki brid kocke iznosi sto milimetara. Prikazan na slici 2.9 “object source“ odabire se desni klikom i dodaje objekt (eng. add object) . U označenom prozoru predmet se dovodi na početak pokretne trake parametrom odmak (eng. offset). Pitch je parametar koji određuje kolika udaljenost je potrebna između početka trake do objekta u kretnji, da se pojavi novi objekt na pokretnoj traci.



Slika 2.9 Dodavanje radnog objekta

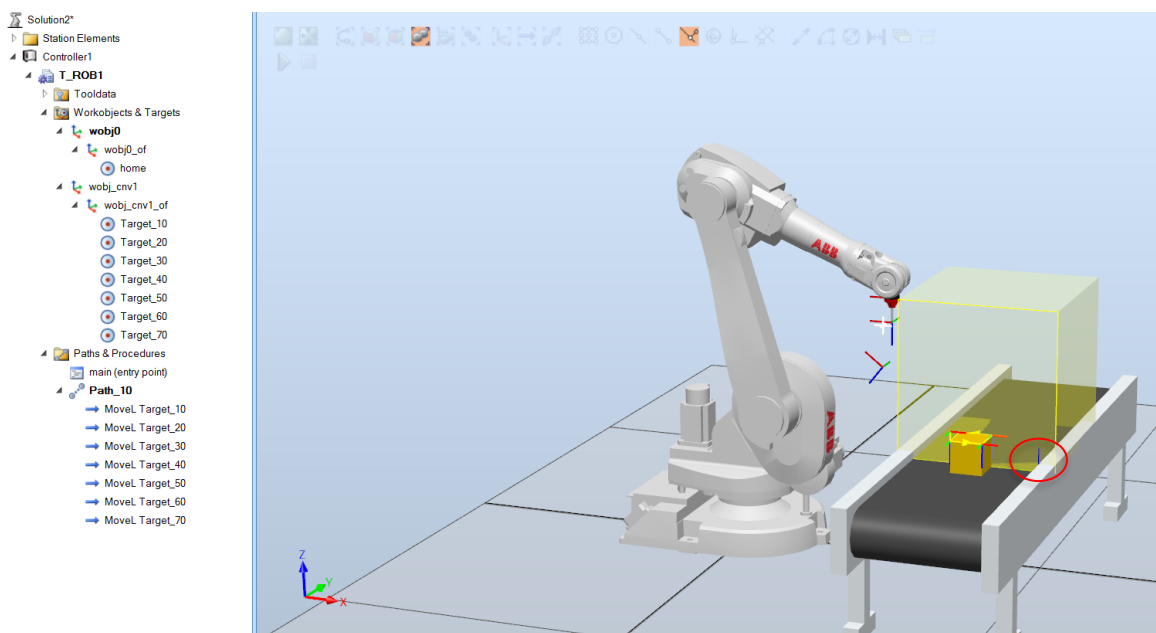
Objekt se na traci pomiče do početka zadanog radnog prostora (tu se nalazi senzor) pomoću naredbe “linear jog“. Desni klik na conveyor i u otvorenom izborniku se nalazi potrebna naredba. Senzor se aktivira, tj. robot prepoznaje da je objekt ušao u radni prostor tek kada koordinatni sustav

objekta dođe u dodir sa “ žutom kutijom“ koja predstavlja radni prostor na pokretoj traci i prikazan je na slici 2.10.



Slika 2.10 Pomak objekta

Kada se objekt pomakne do početka prostora, pomoću naredbe “*auto path*“ u *wobj* pokretne trake (*naziv wobj_cnv1*) programiraju se željene kretnje alata. Taj koordinatni sustav je na početku radnog prostora označen na slici 2.11. Označavanjem željenim točkama, program automatski generira predviđenu putanju alata.

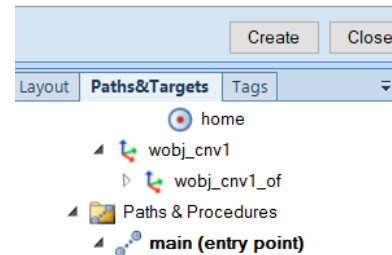
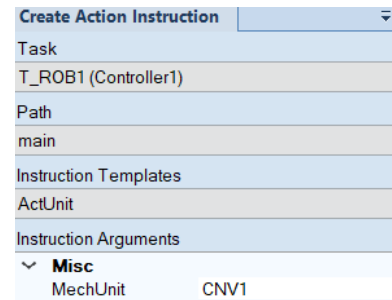


Slika 2.11 Definiranje putanje alata

Desnim klikom na novonastalu putanju “*main (entry point)*“ otvara se izbornik u kojem se odabire opcija “*Create Action Instruction*“.

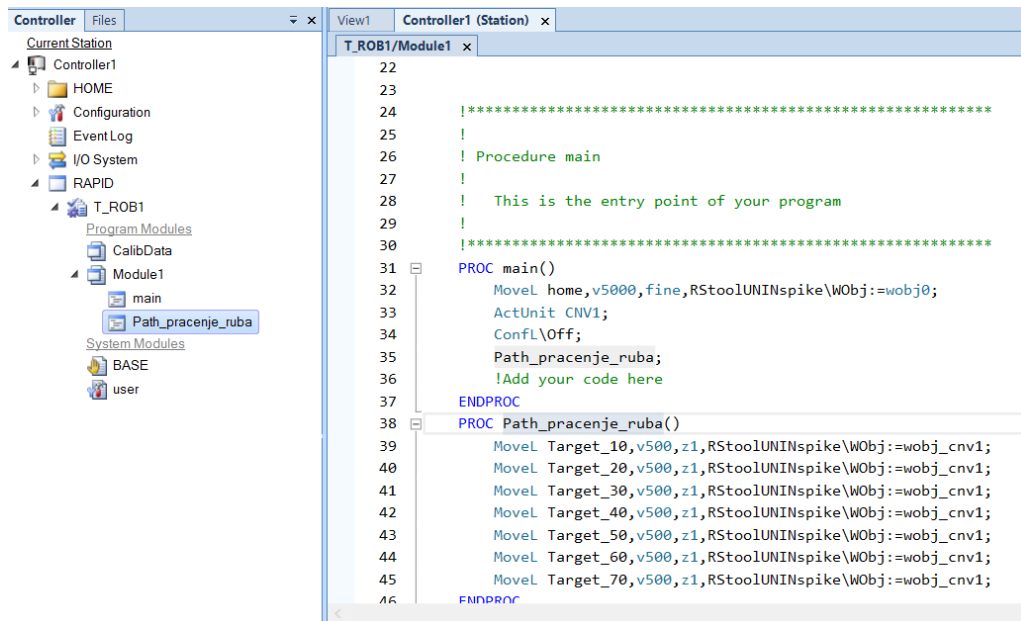
Otvora se izbornik prikazan na slici 2.12 u kojem se stvaraju naredbe “*ActUnit*“ CNV1 i ConfL Off.

Prva naredba uključuje pokretnu traku dok druga omogućuje kretanja *wobj* unutar simulacije. Kada objekt uđe unutar definiranog prostora za praćenje, koordinatni sustav *wobj_cnv1* će ga pratiti istom brzinom i smjerom kretanja.



Slika 2.12 Izbornik

Sinkroniziranjem simulacijske okoline u RAPID JEZIK se prebacuje sve naredbe koje su zadane u RobotStudio grafičkom okruženju. Kod je prikazan na slici 2.13.



Slika 2.13 Početni kod

Naredba “*WaitWObj*“ znači da program čeka da objekt uđe u zadani radni prostor. Naredba “*DropWObj*“ postavlja objekt nazad na pokretnu traku nakon izvršenja postupka praćenja. U kod se dodaje programska petlja prikazanu na slici 2.14.

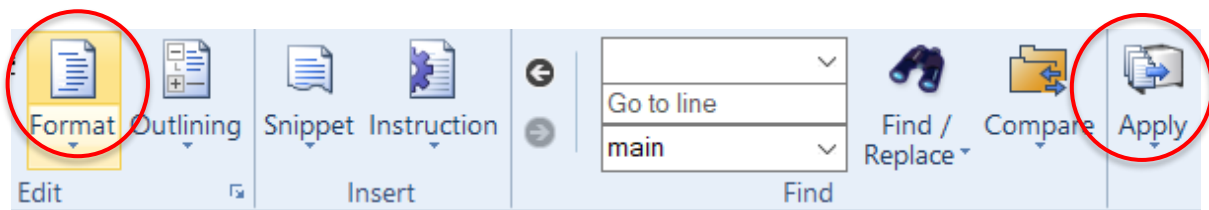
```

26      ! Procedure main
27      !
28      !   This is the entry point of your program
29      !
30      !*****
31  PROC main()
32      MoveJ home,v1000,z100,RStoolUNINspike\WObj:=wobj0;
33      ActUnit CNV1;
34      Confl\Off;
35
36      WHILE TRUE DO
37
38          WaitWObj wobj_cnv1;
39          MoveJ home,v5000,fine,RStoolUNINspike\WObj:=wobj0;
40          Path_pracenje_ruba;
41          DropWObj wobj_cnv1;
42
43      ENDWHILE
44
45
46  ENDPROC
47
48  PROC Path_pracenje_ruba()
49      MoveL Target_10,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
50      MoveL Target_20,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
51      MoveL Target_30,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
52      MoveL Target_40,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
53      MoveL Target_50,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
54      MoveL Target_60,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
55      MoveL Target_70,v500,z1,RStoolUNINspike\WObj:=wobj_cnv1;
56  ENDPROC
57  ENDMODULE

```

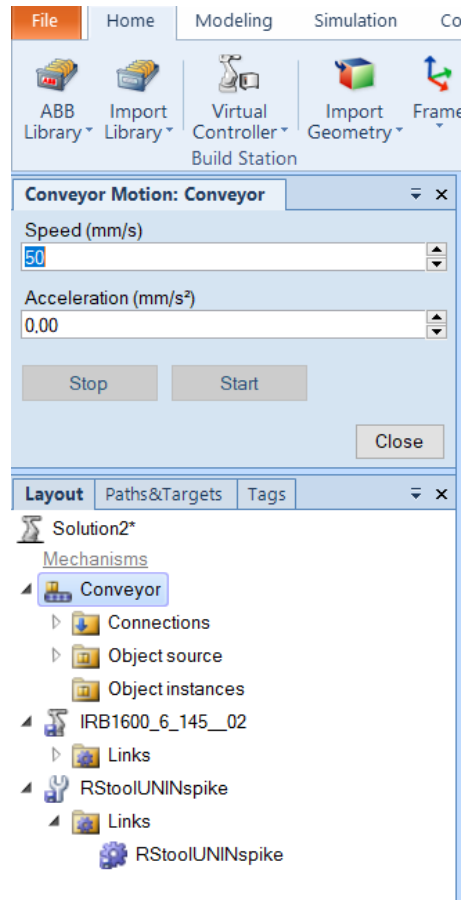
Slika 2.14 Uređeni kod

Naredba za formatiranje (*eng. format*) je označena na slici 2.15. Ta naredba uređuje kod da ljepše izgleda a naredba primjeni (*eng. apply*) prenosi kod iz “rapida“ nazad u simulacijsku okolinu (*eng. station*).



Slika 2.15 Prijenos rapid koda u simulacijsku okolinu

Desni klik na “Conveyor” otvara opciju “Conveyor Motion” prikazano na slici 2.16 . Tu se unosi brzina i ubrzanje pokretne trake . Zaključno sa tim korakom program je spreman za testiranje.



Slika 2.16 “Conveyor motion”

2.2. Rapid kod

Unutar modula prikazanog na slici 2.17 se nalazi novonastali program. U prvom redu se definira sam modul dok redovi od dva do devet prikazuju definirane koordinate korištenih točaka. U redu jedanaest se nalazi početak glavnog programa.

Sljedeći red sadrži naredbu koja šalje robota u njegov home položaj. To je točka u kojoj on čeka dolazak objekta u njegov radni prostor. Naredba *“ActUnit CNV1;“ [4]* uključuje pokretnu traku. Naredba *“ConfL/Off“* isključuje praćenje položaja robota. Time se izbjegava prestanak izvršavanja programa u slučaju da robot ne može dosegnuti određen položaj. Umjesto željenog položaja, automatski se odabire drugi najsličniji zadanom.

```
T_ROB1/Module1* x
1  MODULE Module1
2  CONST robtarget home:=[[317.895,578.660315262,343.687823852],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
3  CONST robtarget Target_10:=[[0,150,130],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
4  CONST robtarget Target_20:=[[0,150,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
5  CONST robtarget Target_30:=[[100,150,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
6  CONST robtarget Target_40:=[[100,250,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
7  CONST robtarget Target_50:=[[0,250,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
8  CONST robtarget Target_60:=[[0,150,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
9  CONST robtarget Target_70:=[[0,150,130],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
10
11 PROC main()
12     MoveJ home,v1000,z100,RStoolUNINspike\WObj:=wobj0;
13     ActUnit CNV1;
14     ConfL\Off;
15
16     WHILE TRUE DO
17
18         WaitWObj wobj_cnv1;
19         Path_pracenje_ruba;
20         MoveJ home,v5000,fine,RStoolUNINspike\WObj:=wobj0;
21         DropWObj wobj_cnv1;
22
23     ENDWHILE
24
25
26 ENDPROC
27
28 PROC Path_pracenje_ruba()
29     MoveL Target_10,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
30     MoveL Target_20,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
31     MoveL Target_30,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
32     MoveL Target_40,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
33     MoveL Target_50,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
34     MoveL Target_60,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
35     MoveL Target_70,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
36 ENDPROC
37 ENDMODULE
```

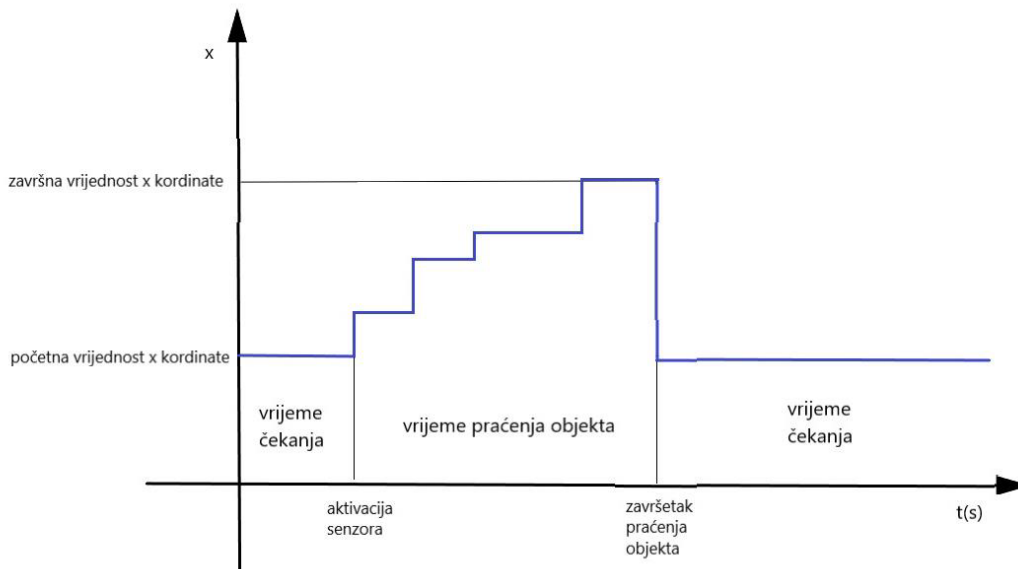
Slika 2.17 Rapid kod praćenja objekta na pokretnoj traci

Naredba “ *while true do* “ omogućuje da se kod unutar petlje ponavlja beskonačno u krug.

Prva naredba u petlji je naredba koja čeka dolazak *wobj* objekta u zadani radni prostor robota. Prilikom ulaska koordinatnog sustava objekta u radni prostor, program se nastavlja dalje izvršavati. Red devetnaest poziva pod program u kojem se nalazi gibanje po putanji. Nakon izvršavanja pod programa, “*MoveJ home*“ pomiče robotski manipulator u njegov početni home položaj. “*DropWObj*“ od spaja trenutni *wobj* objekta od programa za praćenje te čeka dolazak sljedećeg objekta u radni prostor. Pojednostavljeno, briše se *wobj* iz memorije kojemu se kontinuirano povećavala vrijednost koordinate predstavljajući pomak po traci.

3. Praćenje objekta

Praćenje objekta postignuto ovom metodom nije kontinuirano pomicanje koordinata kao u već dosad obrađenoj metodi. Uz pomoć formula i trigonometrije, proračunava se pomak objekta tako da se poklopi vrh alata sa točkom u kojoj će tek biti. Željena koordinata pomiče se nakon svake izvršene kretnje te je dobiveno isprekidano pomicanje koordinate za razliku u prošloj metodi gdje je taj pomak kontinuiran prikazano na slici 3.1. Vrijednost za koju se pomiče koordinata je različita ovisno o geometriji objekta (put po kojem se alat kreće), brzini kretnje objekta i brzini kretnje alata. Izuzimanjem ubrzavanja i usporavanja robotskog manipulatora iz jednadžbe javljaju se odstupanja prilikom dodavanja svake zasebne kretnje (od početne do završne točke) ali i pojednostavljuje izračun.

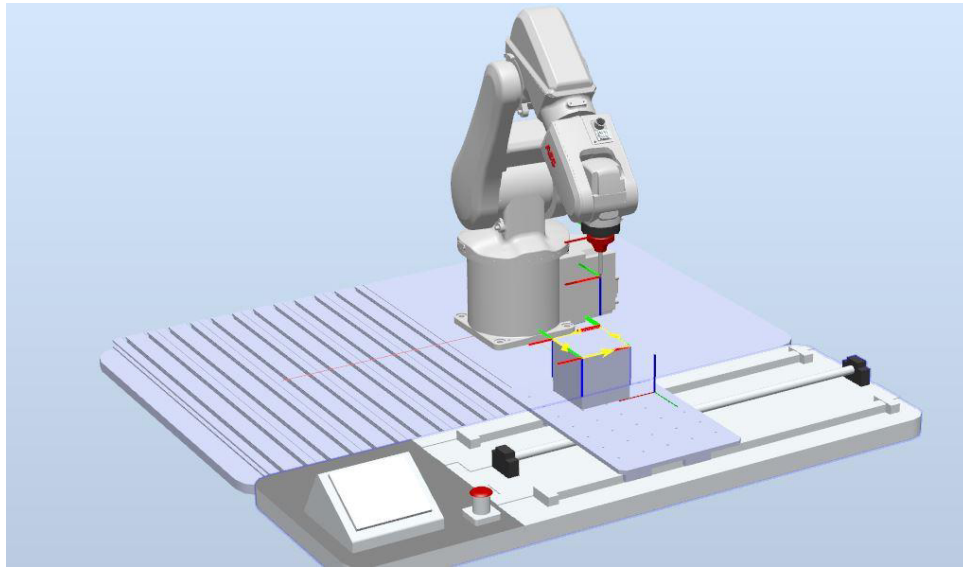


Slika 3.1 Prikaz isprekidanog pomicanja koordinate wobj

3.1. Postavljanje simulacijske okoline

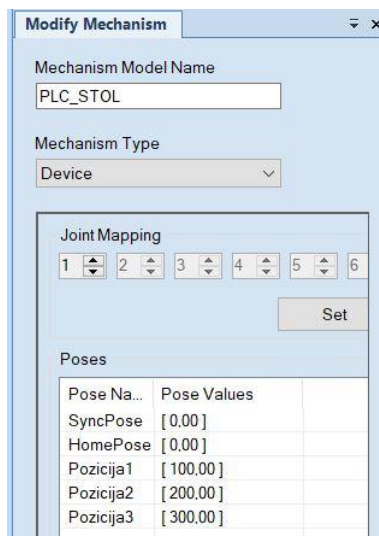
Prvo se dodaje IRB120 robot, stol na kojem je robot montiran, PLC stol, alat i radni objekt. Model PLC stola je definiran kao mehanizam koji je konstruirao kolega Patrik Sivec Starinec te je dostupan na sveučilištu zajedno sa modelom alata i modelom stola robota. Zatim se odabire odgovarajući kontroler.

Radni objekt se zbog potrebe simulacije fiksira na pomičnu ploču.



Slika 3.2 Simulacijska okolina dizajnirana prema laboratoriju

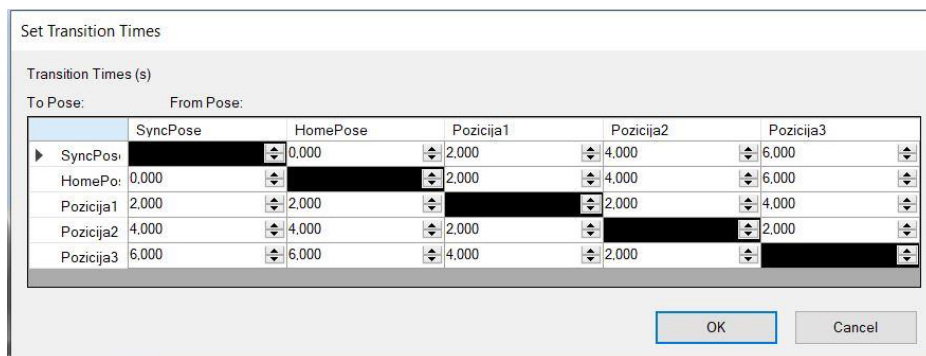
Ulaskom u postavke mehanizma (slika 3.3) mogu se mijenjati definicije položaja i brzina pomicanja ploče.



Slika 3.3 Postavke mehanizma

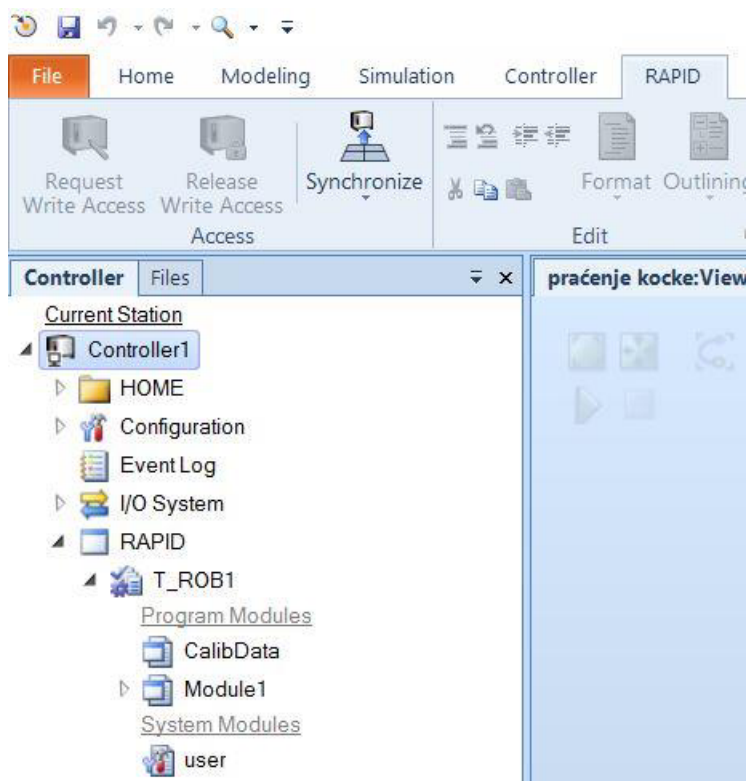
Iz slike 3.3 se može očitati da se između dva položaja stola nalazi sto milimetara razmaka. Za dobivanje željene brzine od 50mm/s, u postavke prikazane na slici 3.4 unosimo prikazane vrijednosti.

Te vrijednosti predstavljaju vrijeme koje treba mehanizmu da dođe iz jednog u drugi položaj.



Slika 3.4 Postavljanje vremena prijelaza iz položaja u položaj

Postavlja se home pozicija u kojoj će se robot nalaziti dok čeka izvršenje programa te nakon izvršavanja programa. Definira se novi *wobj* unutar kojeg naredbom “*autopath*” iscertavamo kretnju po bridovima objekta. Robot koristi dodatak kontroleru kojim preko svojih digitalnih izlaza upravlja radom PLC uređaja. Odlaskom na kontroler prikazano na slici 3.5, desni klik te otvaranjem opcije “*change options*“ dolazimo do željenog izbornika.



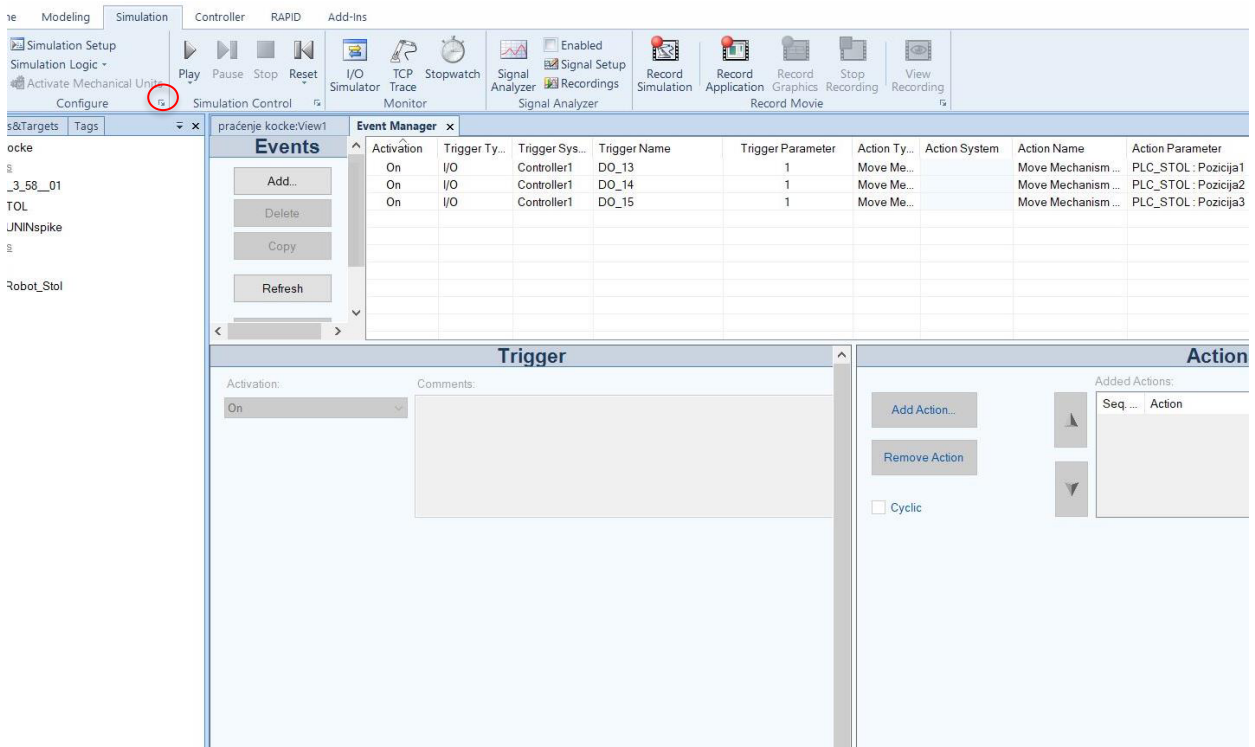
Slika 3.5 Odabir kontrolera

U izborniku prikazanom na slici 3.6 tražimo dodatak “*DeviceNet Master/Slave*“. *DeviceNet* je uređaj koji se koristi u ovom slučaju za komunikaciju između robota (*master*) i slave (PLC uređaja).



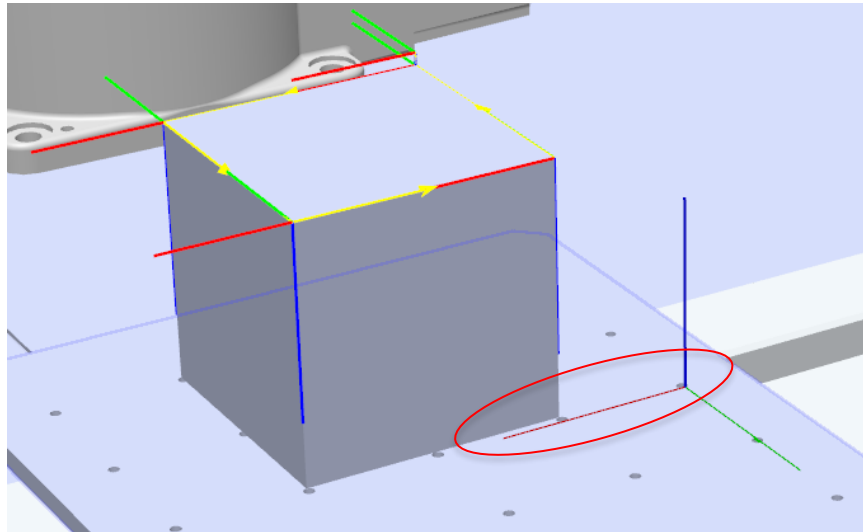
Slika 3.6 Dodatak

Klikom na dodatne postavke konfiguracije istaknuto na slici 3.7, otvora se “*Event Manager*“. Unutar ovog prozora možemo povezati digitalni izlaz sa pomicanjem mehanizma unutar simulacije. Dodavanje veza unutar ovog prozora predstavlja žično spajanje robota i PLC uređaja u laboratoriju.



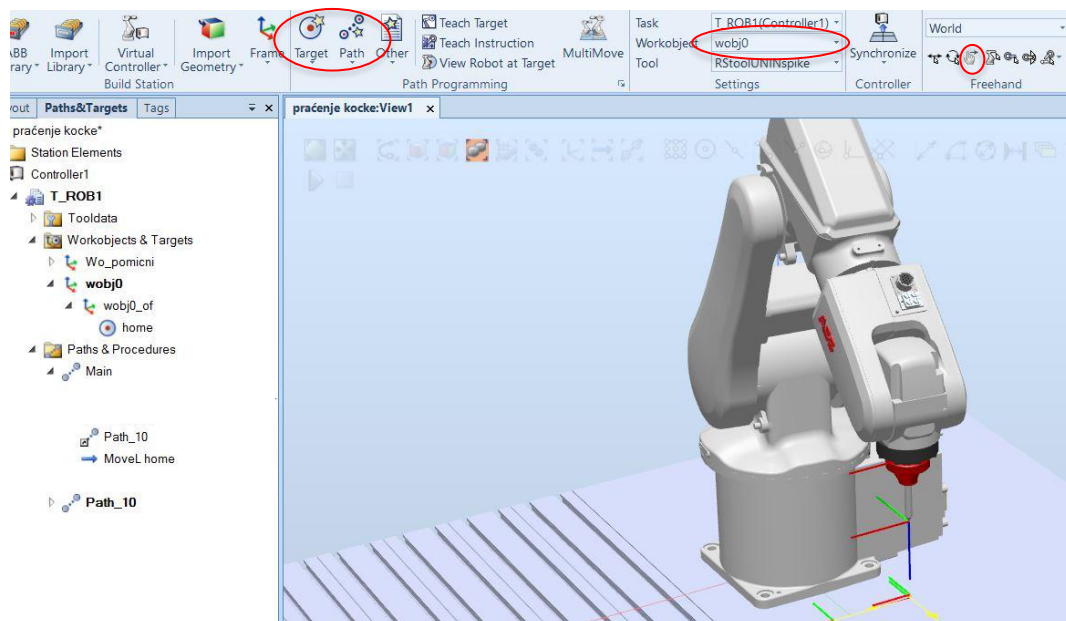
Slika 3.7 Event manager

Definira se novi koordinatni sustav pod nazivom *wo_pomicni*. Postavlja se na pomični stol mehanizma prikazan na slici 3.8. Unutar ovog *wobj* generira se putanja alata naredbom “*auto path*“. Po crveno označenoj x osi kreće se stol.



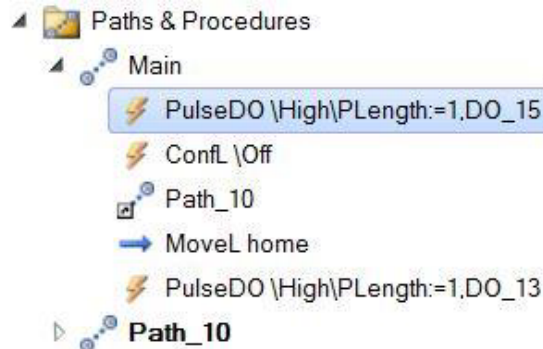
Slika 3.8 *Wo_pomicni*

Robot se prije izvršavanja praćenja nalazi u *home* položaj prikazan slikom 3.9. Isto tako nakon izvršavanja praćenja odlazi nazad u *home* položaj. Sa naredbom “*Jog Linear*”, pomiče se alat iznad radnog objekta. Sljedećom naredbom “*Create Target*” stvaramo položaj home koji se nalazi u glavnom *wobj0*. Mora se još napraviti “*moveL*” naredba u taj položaj pomoću naredbe “*path*”.



Slika 3.9 Home položaj

Dodavanjem "PulseDO" naredbi, šalje se digitalni signal koji pomiče PLC mehanizam. Aktivacija digitalnog izlaza petnaest šalje stol u krajnji položaj dok aktivacija digitalnog izlaza trinaest šalje stol u njegov početni položaj. Naredbe se stvaraju desnim klikom na "Main" prikazan na slici 3.10 i odabirom na opciju "insert action instruction".



Slika 3.10 Redoslijed naredbi

Sinkroniziranjem simulacijske okoline u RAPID JEZIK se prebacuje sve naredbe koje su zadane u RobotStudio grafičkom okruženju. Nakon sinkronizacije kreće ručno izračunavanje pomaka *wobj* prilikom izvršavanja kretanja po putanji. U konačnici kod mora izgledati kao prikazan na slici 3.11. Nakon prebacivanja koda iz rapid jezika u simulacijsku okolinu naredbom "Synchronize" može započeti testiranje.

```

T_ROB1/Module1 x
1  MODULE Module1
2
3  PERS robtargt Target_home:=[[32.937886515,410.000015073,434.550061686],[0.0.707106778,-0.707106774,-0.000000012],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
4  CONST robtargt Target_10:=[[150,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
5  CONST robtargt Target_20:=[[150,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
6  CONST robtargt Target_30:=[[150,-4.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
7  CONST robtargt Target_40:=[[150,-4.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
8  CONST robtargt Target_50:=[[150,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
9  CONST robtargt Target_60:=[[150,-104.2,105],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
10 CONST robtargt home:=[[49.999991828,397.899993007,250.121456331],[0.000000007,1,-0.000000014,0.000000001],[0,-1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
11
12 PROC Main()
13   PulseDO\High\PLength:=1,DO_15;
14   ConfL\OFF;
15   Path_10;
16   MoveL home,v200,z1,RStoolLUNInspike\Wobj:=wobj0;
17   PulseDO\High\PLength:=1,DO_13;
18 ENDPROC
19
20 PROC Path_10()
21   VAR num x1:=0;
22   x1:=x1-27;
23   Wo_pomicni.oframe.trans.x:=x1;
24   MoveL Target_10,v200,z0,RStoolLUNInspike\Wobj:=Wo_pomicni;
25   x1:=x1-20;
26   Wo_pomicni.oframe.trans.x:=x1;
27   MoveL Target_20,v200,z0,RStoolLUNInspike\Wobj:=Wo_pomicni;
28   x1:=x1-25.8198;
29   Wo_pomicni.oframe.trans.x:=x1;
30   MoveL Target_30,v200,z0,RStoolLUNInspike\Wobj:=Wo_pomicni;
31   x1:=x1-33.4;
32   Wo_pomicni.oframe.trans.x:=x1;
33   MoveL Target_40,v200,z0,RStoolLUNInspike\Wobj:=Wo_pomicni;
34   x1:=x1-25.82;
35   Wo_pomicni.oframe.trans.x:=x1;
36   MoveL Target_50,v200,z0,RStoolLUNInspike\Wobj:=Wo_pomicni;
37   MoveL Target_60,v200,z0,RStoolLUNInspike\Wobj:=Wo_pomicni;
38   Wo_pomicni.oframe.trans.x:=0;
39 ENDPROC
40
41 ENDMODULE

```

Slika 3.11 Završni rapid kod

3.2. Rapid kod

Unutar mape “calibdata“ prikazane na slici 3.12 potrebno je promijeniti vrstu definiranja “Wo_pomicni“ kojeg prebacujemo iz “PERS“ u “TASK PERS“. Time mijenjamo konstantu u varijablu koja zadrži svoju vrijednost i nakon gašenja kontrolera.

```
T_ROB1/CalibData x
1  MODULE CalibData
2  PERS tooldata RStoolUNINspike:=[TRUE,[[0,0,80],[1,0,0,0]], [1,[0,0,1],[1,0,0,0],0,0,0]];
3  TASK PERS wobjdata Wo_pomicni:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[0,502.1,51],[1,0,0,0]]];
4  ENDMODULE
```

Slika 3.12 Rapid kod “Calibdata“

Unutar modula prikazanog na slici 3.13 nalazi se program. U prvom redu se definira modul dok redovi od tri do deset prikazuju zapis koordinata korištenih točaka. U redu dvanaest se nalazi početak glavnog programa.

Glavni dio programa se stvara naredbom “PROC Main“. Naredba “PulseDO“ se koristi za slanje pulse digitalnog signala na izlaz robota. Naredba “Confl/Off“ isključuje praćenje položaja robota. Time se izbjegava prestanak izvršavanja programa u slučaju da robot ne može dosegnuti određen položaj. Umjesto željenog položaja, automatski drugi najbližiji zadanom. Pozivom pod programa “Path_10“ započinje praćenje objekta.

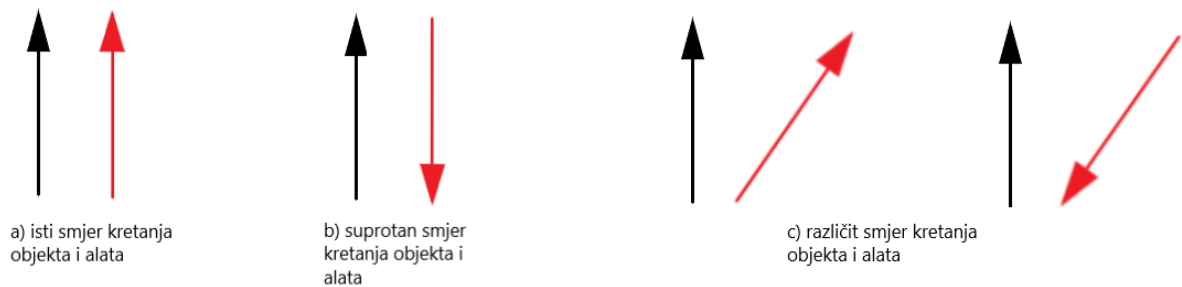
```
T_ROB1/Module1 x
1  MODULE Module1
2
3  PERS robtargat Target_home:=[[32.937886515,410.000015073,434.550061686],[0,0.707106778,-0.000000012],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
4  CONST robtargat Target_10:=[[50,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
5  CONST robtargat Target_20:=[[150,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
6  CONST robtargat Target_30:=[[150,-4.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
7  CONST robtargat Target_40:=[[50,-4.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
8  CONST robtargat Target_50:=[[50,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
9  CONST robtargat Target_60:=[[50,-104.2,105],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
10 CONST robtargat home:=[[49.999991828,397.899993007,250.121456331],[0.000000007,1,-0.000000014,0.000000001],[0,-1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
11
12 PROC Main()
13 PulseDO\High\PLength:=1,DO_15;
14 Confl\Off;
15 Path_10;
16 MoveL home,v200,z1,RStoolUNINspike\Wobj:=wobj0;
17 PulseDO\High\PLength:=1,DO_13;
18 ENDPROC
19
20 PROC Path_10()
21 VAR num x1:=0;
22 x1:=x1-27;
23 Wo_pomicni.oframe.trans.x:=x1;
24 MoveL Target_10,v200,z0,RStoolUNINspike\Wobj:=Wo_pomicni;
25 x1:=x1-20;
26 Wo_pomicni.oframe.trans.x:=x1;
27 MoveL Target_20,v200,z0,RStoolUNINspike\Wobj:=Wo_pomicni;
28 x1:=x1-25.8198;
29 Wo_pomicni.oframe.trans.x:=x1;
30 MoveL Target_30,v200,z0,RStoolUNINspike\Wobj:=Wo_pomicni;
31 x1:=x1-33.4;
32 Wo_pomicni.oframe.trans.x:=x1;
33 MoveL Target_40,v200,z0,RStoolUNINspike\Wobj:=Wo_pomicni;
34 x1:=x1-25.82;
35 Wo_pomicni.oframe.trans.x:=x1;
36 MoveL Target_50,v200,z0,RStoolUNINspike\Wobj:=Wo_pomicni;
37 MoveL Target_60,v200,z0,RStoolUNINspike\Wobj:=Wo_pomicni;
38 Wo_pomicni.oframe.trans.x:=0;
39 ENDPROC
40
41 ENDMODULE
42 ..
```

Slika 3.13 Rapid kod praćenja objekta

Unutar pod programa definira se varijabla čija je početna vrijednost nula. To predstavlja početni položaj pomičnog koordinatnog sustava. Izračunavanjem položaja pomoću formuli o prijeđenom putu i brzini dobijemo vrijednost za koju pomičemo x koordinatu *wobj* prije izvršene kretnje po bridu predmeta. Za pomicanje koordinate korištena je naredba “Wo_pomicni.oframe.trans.x:=“. Na kraju programa potrebno je *wobj* vratiti u početni položaj kako bi dolaskom sljedećeg predmeta moglo opet biti odrađeno praćenje.

3.3. Korištene formule

Moguće linearne kretnje se mogu svrstati u tri grupe prema kojim se izvode formule. Formule su dobivene koristeći pojedine brzine i prijeđenog puta. Traži se broj za koji se pomiče koordinata korištenog *wobj*.



Slika 3.14 Grupe linearnih kretnji

Slučaj u kojem je smjer kretanja objekta i alata isti:

$$x - y = z \quad (1)$$

$$\frac{x}{v_1} = \frac{y}{v_2} \quad (2)$$

x – iznos za koji se pomiče koordinata *wobj*

y – put koji pređe početna točka od koje alat i objekt oboje kreću

z – iznos dužine po kojem putuje alat

*v*₁ – brzina kretnje alata

*v*₂ – brzina kretnje objekta (pokretne trake)

Suprotan smjer kretanja objekta i alata:

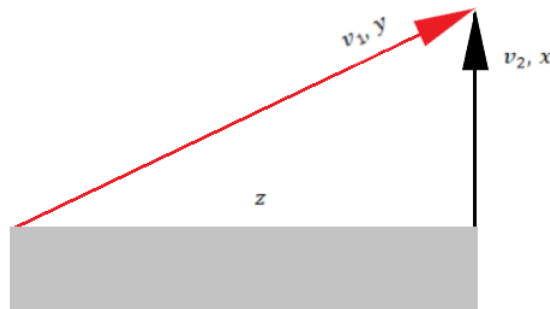
$$x - y = z \quad (1)$$

$$\frac{y}{v_1} = \frac{x}{v_2} \quad (3)$$

Za različit smjer kretnje objekta i alata se koriste formule (4) i (5). Slika 3.15 prikazuje crvenom bojom smjer kretanja alata dok crna boja prikazuje smjer kretanja objekta.

$$y = \sqrt{z^2 + x^2} \quad (4)$$

$$\frac{\sqrt{z^2 + x^2}}{v_1} = \frac{x}{v_2} \quad (5)$$



Slika 3.15 Prikaz različitog smjera kretnji

3.4. Mogući način izvedbe kontinuiranog pomaka

Normalno praćenje objekta na pokretnoj traci koristi dodatne ABB komponente kako bi se *wobj* mogao kontinuirano pomicati po jednoj od osi *wobj*. Isprekidana metoda izračunom nagađa gdje će se nalaziti željena točka nakon određenog vremena. Uz pomoć dodatnog programa koji će se odvijati sa već glavnim rapid kodom moglo bi se kontinuirano povećavati ili smanjivati koordinata željene osi.

Kod na slici 3.16 prikazuje ideju modula koji uz pomoć drukčijeg *timera* bi mogao postići željeno. Primjer ovog koda je preuzet *timer* sa ABB foruma [3]. *Timer* napravi jedan ciklus za najmanje jednu sekundu sa dodatnim vremenom izvršavanja svakog reda naredbi. Jedna sekunda je prevelika vremenska jedinica da bi se kontinuirano pomicala koordinata.

```

MODULE Module2
  VAR num brojac:=0;
  PROC testwhile()
    WHILE NOT bExit DO
      WaitTime 1;
      Wo_pomicni.oframe.trans.x:=brojac;
      Incr brojac;
      IF brojac = 10 THEN
        bExit:=TRUE
      ENDIF
    ENDWHILE
  ENDPROC
ENDMODULE

```

Slika 3.16 Kod modul2

Ideja je da se napravi tajmer koji će se na temelju vremena potrebnog za obavljanje pojedine naredbe tj. ciklusa procesora, staviti u petlju i koristiti za pomicanje koordinate. Ovime bi se postigla manja vremenska jedinica od sekunde te bi se time pomicanje izvodilo ravnomjernije.

4. Analiza rezultata

4.1. Praćenje objekta na pokretnoj traci

Snimka provedene simulacije [5].

Proučavanjem rapid koda može se primijetiti da ne postoje naredbe koje definiraju brzinu kretnje pokretne trake. Simulacijsko okruženje unutar programa simulira pokretnu traku i enkoder usporedno sa izvršavanjem rapid koda. Iako se zasebno izvode međusobno komuniciraju robot i pokretna traka. To se može zaključiti jer se koordinatni sustav *wobj* pomiče bez zadanih naredbi unutar rapid koda.

4.2. Praćenje objekta

Snimka provedene simulacije [6].

Prilikom izvršavanja svake zasebne kretnje alata od početne do završne točke, javlja se odstupanje. To odstupanje se javlja zbog neuračunatog usporavanja i ubrzavanja alata. Što se više linearnih kretnji izvrši uzastopno povećava se kašnjenje alata. Zbog nedostatka senzora i graničnih prekidača u sustavu, robot nema povratne informacije o tome gdje se predmet stvarno nalazi.

5. Zaključak

Iako se u drugoj metodi postignuto praćenje objekta bez pokretne trake i enkodera, ograničeno je na linearne kretnje i javljaju se veća odstupanja. Metoda korištena u praksi obrađena u prvom dijelu rada se može bez problema primijeniti i na nelinearnim putanjama. Ujedno odstupanja su minimalna zbog kontinuiranog pomicanja koordinatnog sustava. Dolazak novog radnog objekta prije završetka praćenja prethodnog ne stvara problem, već će se praćenje drugog objekta krenut izvršavati nakon dovršetka praćenja prvog. Ponovljivost u obje metode je odlična.

Metoda praćenja objekta bez pokretne trake i enkodera bi se ipak mogla iskoristiti kod industrijskih procesa u kojima preciznost nije toliko potrebna. Postigla bi se manja ušteda sredstava ali je teže za programiranje. Dolazak novog radnog objekta prije završetka praćenja prethodnog stvara problem. Iako ova druga metoda nije savršena mogla bi se primijeniti prilikom nanošenja boje zrakom. Možda je moguće i bolje rješenje, ali druge metode će ostati predmet istraživanja u narednom periodu.

6. Literatura

- [1] <https://library.e.abb.com/public/9423568222564896af57d2109611c955/3HAC066561%20AM%20Conveyor%20tracking%20RW%207-en.pdf?x-sign=zhUueAeOmfagn9EFea3vf5Jz8BeHIfgi8anz062So13GGvMdLTX7MWxk/UzfeV7b> dostupno 15.7.2024.
- [2] <https://forums.robotstudio.com/> dostupno 21.8.2024.
- [3] Z., Car. (07.05.2021.), Robotika - Uvod u robotiku, Sveučilište u Rijeci, Rijeka
- [4] https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf dostupno 26.8.2024.
- [5] <https://www.youtube.com/watch?v=s0y4TzUBWM4&t=1s> dostupno 27.8.2024.
- [6] <https://www.youtube.com/watch?v=fEXJvcTm2W4&t=73s> dostupno 27.8.2024.

Popis slika

Slika 1.1 ABB paket za praćenje na pokretnoj traci [1]	1
Slika 1.2 Prikaz povezanosti dostupnih komponenti u laboratoriju	2
Slika 2.1 Prikaz kontinuiranog pomicanja koordinate wobj	3
Slika 2.2 Simulacijska okolina.....	4
Slika 2.3 Odabir kontrolera.....	4
Slika 2.4 Kartica odabira.....	5
Slika 2.5 Dodavanje enkodera	5
Slika 2.6 Odabir izbornika modeling	6
Slika 2.7 Definiranje pokretne trake	6
Slika 2.8 Definiranje pozicije senzora	7
Slika 2.9 Dodavanje radnog objekta	7
Slika 2.10 Pomak objekta	8
Slika 2.11 Definiranje putanje alata.....	8
Slika 2.12 Izbornik.....	9
Slika 2.13 Početni kod	9
Slika 2.14 Uređeni kod	10
Slika 2.15 Prijenos rapid koda u simulacijsku okolinu.....	10
Slika 2.16 “Conveyor motion“.....	11
Slika 2.17 Rapid kod praćenja objekta na pokretnoj traci	12
Slika 3.1 Prikaz isprekidanog pomicanja koordinate wobj.....	14
Slika 3.2 Simulacijska okolina dizajnirana prema laboratoriju	15
Slika 3.3 Postavke mehanizma	15
Slika 3.4 Postavljanje vremena prijelaza iz položaja u položaj	16
Slika 3.5 Odabir kontrolera.....	16
Slika 3.6 Dodatak.....	17
Slika 3.7 Event manager	17
Slika 3.8 Wo_pomicni	18
Slika 3.9 Home položaj.....	18
Slika 3.10 Redoslijed naredbi	19
Slika 3.11 Završni rapid kod.....	19
Slika 3.12 Rapid kod “Calibdata“.....	20
Slika 3.13 Rapid kod praćenja objekta	20
Slika 3.14 Grupe linearnih kretnji.....	21
Slika 3.15 Prikaz različitog smjera kretnji.....	22
Slika 3.16 Kod modul2	23

Prilozi

```
T_ROB1/Module1* x
1  MODULE Module1
2  CONST robtarget home:=[[317.895,578.660315262,343.687823852],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
3  CONST robtarget Target_10:=[[0,150,130],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
4  CONST robtarget Target_20:=[[0,150,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
5  CONST robtarget Target_30:=[[100,150,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
6  CONST robtarget Target_40:=[[100,250,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
7  CONST robtarget Target_50:=[[0,250,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
8  CONST robtarget Target_60:=[[0,150,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
9  CONST robtarget Target_70:=[[0,150,130],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,0]];
10
11 PROC main()
12   MoveJ home,v1000,z100,RStoolUNINspike\WObj:=wobj0;
13   ActUnit CNV1;
14   Confl\Off;
15
16   WHILE TRUE DO
17
18     WaitWObj wobj_cnv1;
19     Path_pracenje_ruba;
20     MoveJ home,v5000,fine,RStoolUNINspike\WObj:=wobj0;
21     DropWObj wobj_cnv1;
22
23   ENDWHILE
24
25
26 ENDPROC
27
28 PROC Path_pracenje_ruba()
29   MoveL Target_10,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
30   MoveL Target_20,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
31   MoveL Target_30,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
32   MoveL Target_40,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
33   MoveL Target_50,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
34   MoveL Target_60,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
35   MoveL Target_70,v5000,fine,RStoolUNINspike\WObj:=wobj_cnv1;
36 ENDPROC
37 ENDMODULE
```

```
T_ROB1/Module1 x
1  MODULE Module1
2
3  PERS robtarget Target_home:=[[32.937886515,410.000015073,434.550061686],[0,0,707106788,-0.707106774,-0.000000012],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
4  CONST robtarget Target_10:=[[50,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
5  CONST robtarget Target_20:=[[150,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
6  CONST robtarget Target_30:=[[150,-4.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
7  CONST robtarget Target_40:=[[50,-4.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
8  CONST robtarget Target_50:=[[50,-104.2,100],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
9  CONST robtarget Target_60:=[[50,-104.2,105],[-0.000000007,1,-0.000000014,0.000000001],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
10 CONST robtarget home:=[[49.999991828,397.899993007,250.121456331],[0.000000007,1,-0.000000014,0.000000001],[0,-1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
11
12 PROC Main()
13   PulseDO\High\PLength:=1,DO_15;
14   Confl\Off;
15   Path_10;
16   MoveL home,v200,z1,RStoolUNINspike\WObj:=wobj0;
17   PulseDO\High\PLength:=1,DO_13;
18 ENDPROC
19
20 PROC Path_10()
21   VAR num x1:=0;
22   x1:=x1-27;
23   Wo_pomicni.oframe.trans.x:=x1;
24   MoveL Target_10,v200,z0,RStoolUNINspike\WObj:=Wo_pomicni;
25   x1:=x1-20;
26   Wo_pomicni.oframe.trans.x:=x1;
27   MoveL Target_20,v200,z0,RStoolUNINspike\WObj:=Wo_pomicni;
28   x1:=x1-25.8198;
29   Wo_pomicni.oframe.trans.x:=x1;
30   MoveL Target_30,v200,z0,RStoolUNINspike\WObj:=Wo_pomicni;
31   x1:=x1-33.4;
32   Wo_pomicni.oframe.trans.x:=x1;
33   MoveL Target_40,v200,z0,RStoolUNINspike\WObj:=Wo_pomicni;
34   x1:=x1-25.82;
35   Wo_pomicni.oframe.trans.x:=x1;
36   MoveL Target_50,v200,z0,RStoolUNINspike\WObj:=Wo_pomicni;
37   MoveL Target_60,v200,z0,RStoolUNINspike\WObj:=Wo_pomicni;
38   Wo_pomicni.oframe.trans.x:=0;
39 ENDPROC
40
41 ENDMODULE
42
```

Sveučilište
SjeverSVEUČILIŠTE
SJEVER

IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Ivan Marko Kovač (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog/specijalističkog (obrisati nepotrebno) rada pod naslovom Pracenje objekta na pokretnoj traci (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.