Planiranje istovremenog kretanja dvaju robota s preklapajućim radnim prostorom

Dubić, Krunoslav

Master's thesis / Diplomski rad

2011

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:794546

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-16**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

PLANIRANJE ISTOVREMENOG KRETANJA DVAJU ROBOTA S PREKLAPAJUĆIM RADNIM PROSTOROM

DIPLOMSKI RAD

KRUNOSLAV DUBIĆ

Zagreb, 2011.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

PLANIRANJE ISTOVREMENOG KRETANJA DVAJU ROBOTA S PREKLAPAJUĆIM RADNIM PROSTOROM

DIPLOMSKI RAD

Mentor:
prof. dr. sc. BOJAN JERBIĆ

KRUNOSLAV DUBIĆ

Zagreb, 2011.

Izjavljujem da sam ovaj rad izradio samostalno, primjenom znanja stečenih na Fakultetu Strojarstva i Brodogradnje, te uz korištenje navedene literature.

Zahvaljujem se mentoru prof. dr. sc. Bojanu Jerbić. Posebno se zahvaljujem asistentu dr. sc. Petru Ćurković na nesebičnoj i vrlo korisnoj pomoći te na trudu i vremenu kojeg je uložio vodeći me kroz ovaj rad. Zahvaljujem se i asistentu mag. ing. Marku Švacu na informacijama i podacima koje mi je dao na uvid i korištenje prilikom rada u Laboratoriju za projektiranje izradbenih i montažnih sustava.

Krunoslav Dubić

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

DIPLOMSKI ZADATAK

Student: **KRUNOSLAV DUBIĆ** Mat. br.: 003518720

Naslov rada na hrvatskom jeziku: **PLANIRANJE ISTOVREMENOG KRETANJA DVAJU ROBOTA S PREKLAPAJUĆIM RADNIM PROSTOROM**

Naslov rada na engleskom jeziku: **PLANNING OF SIMULTANEOUS MOTION OF TWO ROBOTS WITH OVERLAPPING WORK SPACE**

Opis zadatka:

Roboti koji dijele radni prostor predstavljaju dinamičku prepreku jedan drugome. U takvim uvjetima problem planiranja kretanja svakog robota postaje teško rješiv u stvarnom vremenu.

U prvome dijelu diplomskog rada potrebno je izraditi pojednostavljeni simulacijski model te algoritam za rješavanje problema planiranja istovremenog kretanja dvaju robota koji dijele radni prostor. Kritičan parameter algoritma je vrijeme procesiranja odnosno pronalaženja prihvatljivog rješenja, te ga je potrebno minimizirati.

U drugome dijelu rada rješenja pronađena u simulacijskom okružju potrebno je implementirati na dva stvarna robota Fanuc LR Mate 200iC.

Ulaz u simulacijsko okružje su konfiguracije robota u početnom i konačnom trenutku vremena. Rješenja algoritma trebaju biti u obliku kontinuiranih krivulja koje opisuju relativne položaje zglobova robota tijekom tranzicije od početne do konačne konfiguracije.

Zadatak zadan:	Rok predaje rada:	Predviđeni datumi obrane:
08. rujna 2011.	10. studenog 2011.	16. – 18. studenog 2011.

Zadatak zadao: Predsjednik Povjerenstva:

Prof. dr. sc. Bojan Jerbić Prof. dr. sc. Franjo Cajner

Referada za diplomske i završne ispite Obrazac DS – 3B/ PDS – 3B

SADRŽAJ

1 UVOD 1

1.1 Pregled dosadašnjih spoznaja 4

1.2 Tradicionalne metode planiranja kretanja 6

1.3 Više robota u radnom prostoru 7

1.4 Cilj rada 9

2. DEFINICIJA PROBLEMA 10

2.1 Ispitivanje radnog područja 13

2.2 Ispitivanje prepreka u radnom području robota 15

2.3 Definiranje najkraće putanje robota 16

2.4 Način izvršavanja zadatka 18

3 METODA RJEŠAVANJA PROBLEMA 22

3.1 Start - pokretanje sustava 23

3.1.1 Definiranje koordinatnih sustava za robote Fanuc LR Mate 200iC 25

3.1.2 Povezivanje robota i računala 29

3.2 Definiranje prepreka za robote Fanuc LR Mate 200iC 30

3.3 Metoda za određivanje pozicija prepreka u radnom prostoru 32

3.4 Granični uvjeti 35

3.4.1 Prepreka sijeće pravce početne i konačne konfiguracije 36

3.4.2 Prepreka sijeće pravac početne konfiguracije 41

3.4.3 Prepreka sijeće pravac krajnje konfiguracije 44

3.4.4 Prepreka se nalazi između pravaca početne i krajnje konfiguracije 46

3.4.5 Prepreka se nalazi izvan pravaca početne i krajnje konfiguracije 47

3.5 Definiranje točaka putanje 48

3.6 Slanje poruke računalo-robot 51

3.7 Simulacija u fizičkom simulacijskom okružju Roboguide 53

4. ZAKLJUČAK 55

5. LITERATURA 57

POPIS SLIKA

Slika 1.1 Dva robota koji dijele radni prostor. Laboratorij za projektiranje izradbenih i monatžnih sustava **.....3**

Slika 1.2 Primjer početne (lijevo) i konačne (desno) konfiguracije za koje decentralizirani pristup ne daje zadovoljavajuće rješenje **.....8**

Slika 2.1 Primjer rješenja evolucijskog algoritma s dva agenta **.....10**

Slika 2.2 Agent s dva segmenta i dva rotacijska zgloba **.....11**

Slika 2.3 Agent s jednim segmentom i jednim rotacijskim zglobom **.....11**

Slika 2.4 Zaobilaženje prepreke **.....12**

Slika 2.5 Prikaz agenta s kromosomom duljine *l*=4 (***a***) i duljine *l*=2 (***b***) **.....13**

Slika 2.6 a) Prikaz agenata koji se sijeku u početnoj konfiguraciji ***b)*** Prikaz agenata koji se sijeku u krajnjoj konfiguraciji **.....14**

Slika 2.7 Drugi agent u radnom prostoru prvog agenta **.....14**

Slika 2.8 Prikaz radnog područja robota omeđenog pravcima **.....15**

Slika 2.9 Prikaz točaka najkraće putanje robota **.....17**

Slika 2.10 Sučelje programskog jezika Matlab **.....18**

Slika 2.11 Fanuc LR Mate 200iC **.....19**

Slika 2.12 Shematski prikaz šest-osnog upravljačkog sklopa (robota) – LR Mate 200*i*C5L **.....19**

Slika 2.13 Sučelje programskog paketa Roboguide **..... 21**

Slika 3.1 Dijagram toka **.....22**

Slika 3.2 Redosljed pokretanja sustava **.....23**

Slika 3.3 Upravljačka konzola spremna za izvođenje programa **.....24**

Slika 3.4 Koordinatni sustav robota, korisnički koordinatni sustav **.....25**

Slika 3.5 a) Točka prihvata alata ***b)*** Koordinatni sustav središta alata **.....26**

Slika 3.6 Koordinatni sustav - pravilo desne ruke **.....26**

Slika 3.7 Međusobni položaj lijevog i desnog robota **.....27**

Slika 3.8 Koordinatni sustavi robota **.....27**

Slika 3.9 Udaljenost između robota **.....28**

Slika 3.10 Komunikacija između upravljačke jedinice robota i računala **.....29**

Slika 3.11 Početna i krajnja konfiguracija oba robota **.....30**

Slika 3.12 Desni robot kao prepreka lijevom robotu **.....31**
Slika 3.13 Lijevi robot kao prepreka desnom robotu **.....32**
Slika 3.14 Radno područje robota omeđeno pravcima **.....33**
Slika 3.15 Konfiguracija robota **.....33**
Slika 3.16 Karakteristične točke **.....35**
Slika 3.17 Granični uvjet 1 **.....36**
Slika 3.18 Granični uvjet 2 **.....36**
Slika 3.19 Granični uvjet 3 **.....37**
Slika 3.20 Granični uvjet 4 **.....37**
Slika 3.21 Granični uvjet 5 **.....37**
Slika 3.22 Granični uvjet 6 **.....37**
Slika 3.23 Granični uvjet 7 **.....38**
Slika 3.24 Granični uvjet 8 **.....38**
Slika 3.25 Granični uvjet 9 **.....38**
Slika 3.26 Granični uvjet 10 **.....38**
Slika 3.27 Granični uvjet 11 **.....39**
Slika 3.28 Granični uvjet 12 **.....39**
Slika 3.29 Granični uvjet 13 **.....39**
Slika 3.30 Granični uvjet 14 **.....39**
Slika 3.31 Granični uvjet 15 **.....40**
Slika 3.32 Granični uvjet 16 **.....40**
Slika 3.33 Granični uvjet 17 **.....40**
Slika 3.34 Granični uvjet 18 **.....40**
Slika 3.35 Granični uvjet 19 **.....41**
Slika 3.36 Granični uvjet 20**.....41**
Slika 3.37 Granični uvjet 2 **.....41**
Slika 3.38 Granični uvjet 22 **.....41**
Slika 3.39 Granični uvjet 23 **.....42**
Slika 3.40 Granični uvjet 24 **.....42**
Slika 3.41 Granični uvjet 25 **.....42**
Slika 3.42 Granični uvjet 26 **.....42**
Slika 3.43 Granični uvjet 27 **.....43**

Slika 3.44 Granični uvjet 28 **.....43**
Slika 3.45 Granični uvjet 29 **.....43**
Slika 3.46 Granični uvjet 30 **.....44**
Slika 3.47 Granični uvjet 31 **.....44**
Slika 3.48 Granični uvjet 32 **.....44**
Slika 3.49 Granični uvjet 33 **.....44**
Slika 3.50 Granični uvjet 34 **.....45**
Slika 3.51 Granični uvjet 35 **.....45**
Slika 3.52 Granični uvjet 36 **.....45**
Slika 3.53 Granični uvjet 37 **.....45**
Slika 3.54 Granični uvjet 38 **.....46**
Slika 3.55 Granični uvjet 39 **.....46**
Slika 3.56 Granični uvjet 40 **.....46**
Slika 3.57 Granični uvjet 41 **.....47**
Slika 3.58 Definiranje točaka putanje **.....48**
Slika 3.59 Skraćena putanja robota **.....49**
Slika 3.60 Uvećani prikaz skraćivanja putanje **.....49**
Slika 3.61 Skraćivanje putanje za radijus drugog robota **.....50**
Slika 3.62 Skraćivanje putanje za radijus prvog robota **.....50**
Slika 3.63 Konačan izgled skraćene putanje **.....51**
Slika 3.64 Princip rada fanucovog *Roboguide* simulacijskog okružja **..... 53**

POPIS TABLICA

Tablica 1 Osnovne tehničke karakteristike robota Fanuc LR Mate 200iC **.....20**

Sažetak

U ovom radu razvijen je algoritam za rješavanje problema planiranja istovremenog kretanja dvaju robota koji dijele radni prostor i predstavljaju dinamičku prepreku jedan drugome. U prvome dijelu diplomskog rada izrađen je pojednostavljeni simulacijski model te algoritam za rješavanje problema planiranja putanja dvaju robota. U drugome dijelu rada rješenja pronađena u simulacijskom okružju implementirana su na dva stvarna robota Fanuc LR Mate 200iC.

Najveći problem kod planiranja istovremenog kretanja dvaju robota s preklapajućim radnim prostorom je pronaći rješenje njihovih putanja u što kraćem vremenu i sa što većom točnošću rješavanja problema kako ne bi došlo do kolizije između njih. Ta dva parametra usmjerila su sva razmišljanja na rješavanje problema matematičkim putem koji će za svaku moguću poziciju robota uvijek dati sigurno rješenje u što kraćem vremenu.

Kreiranjem vlastitih programa i funkcija koje analiziraju i obrađuju primljene podatke istovremeno od oba robota uspješno je napravljen algoritam za beskonačan broj pozicija u kojima se mogu susreti dva robota u zajedničkom radnom području. Uz glavni program u kojemu se primaju i šalju informacije robotu, poziva se trideset podprograma za rješavanje problema najkraće putanje. Time je znatno ubrzano pronalaženje najkraće putanje jer se ne izvršava jedan program već pojedini podprogrami koji ispunjavaju uvjete u kojima radi robot. Funkcija pronalaženja najkraće putanje svedena je na manje od pola sekunde što omogućuje robotu da radi u realnom vremenu.

Metodologija je verificirana na realnom robotskom sustavu, sastavljenom od dva robota Fanuc LR Mate 200iC za koje je problem planiranja kretanja razvijenom metodologijom uspješno riješen u realnom vremenu.

Ključne riječi: planiranje kretanja, robotika.

1 UVOD

Procjene pokazuju kako se u bliskoj budućnosti očekuje izražen rast tržišta uslužne robotike. Unatoč značajnim resursima uloženim u razvoj robota prihvatljivih širokom krugu korisnika, ovi roboti i dalje nisu zastupljeni s udjelom koji je očekivan na temelju provedenih različitih recentnih analiza. Izazov robotike danas je razvoj robota sposobnih za opsežnu interakciju s ljudima. Pri tome se ističu dva problema: razvoj dovoljno robusnih upravljačkih algoritama, te razvoj komunikacijskog sučelja prihvatljivog najširem krugu korisnika.

Okružje u kojem trebaju djelovati roboti koji izlaze iz okvira proizvodnih okolina, promjenjivo je i nekarakterizirano, dakle kompleksno i neprilagođeno robotu. Robot koji treba izvršiti zadaću u takvoj okolini, pored znanja o samom zadatku mora imati sposobnost spoznaje, interpretacije i interakcije s okolinom. Razvojem kompleksne senzorike, u prvom redu eksteroceptičnih senzora poput strojnog vida, osigurane su pretpostavke za akviziciju informacija iz okoline. Razvojem proprioceptičnih senzora poput *GPS* senzora, osigurava se bolja lokalizacija robota u prostoru. Ipak, ostaju prisutni problemi brzine obrade informacija potrebnih za planiranje budućih akcija robota te problemi pojave šuma pri akviziciji podataka.
Pokazalo se da razvoj općenitog monolitnog upravljačkog programa za robote koji djeluju u opisanim uvjetima nije moguć zbog nesagledivog povećanja kompleksnosti upravljačkog algoritma [1]. Naime, nemoguće je strukturama *ako-onda* obuhvatiti sve situacije u kojima se robot tijekom djelovanja može naći. Iako je takav pristup prihvatljiv u industrijskim okolinama, gdje je prostor stanja robota ograničen na skup unaprijed poznatih stanja, koje se rješavaju skupom poznatih djelovanja, roboti najnovije generacije moraju počivati na drugačijim upravljačkim paradigmama.

Područje istraživanja inteligentnih robota vrlo je aktivno, interdisciplinarno i obuhvaća, među ostalim: razvoj sklopovlja - senzoriku, upravljačka računala, materijale, i upravljačku stranu: arhitekture, algoritme, protokole. Kako se očekivanja postavljena na razvijene robote sve više približavaju željenim svojstvima živih organizama, u razvoju takvih robota sudjeluju i istraživaći iz područja biologije, psihologije, neuroznanosti.

Uspoređujući prirodne, naizgled jednostavne sustave poput mravljih zajednica, iznenađuje koliko su sposobnosti robota koji se danas nalaze na najvišem stupnju razvoja ograničene. Radnje poput kretanja po nekarakteriziranoj podlozi, izbjegavanja prepreka, praćenja pomičnog cilja, prepoznavanja, zajedničkog nošenja tereta, snalaženja i lokalizacije u nepoznatoj okolini samo su neki od problema s kojima se istraživaći danas intenzivno bave.

Pretpostavka za uspješno rješavanje zadatka pri radu autonomnog robota je sposobnost samostalnog kretanja u nepoznatoj okolini. Planiranje putanje općenito je težak problem, a posebice za sustave koji sadrže veći broj robota koji dijele radni prostor. Planiranje putanje u industrijskim je okolinama bitno zbog težnje zamjene ljudskog rada robotskim, posebice kod monotonih ili opasnih radnih mjesta. Unatoč ovoj težnji, ljudi i roboti najčešće dijele radni prostor, pri čemu su roboti unaprijed programirani i njihovo je kretanje moguće predvidjeti. No, u slučaju implementacije autonomnih robota, kretanje robota nije unaprijed poznato, a koliziju s čovjekom, koja najčešće rezultira ozljedom potrebno je bezuvjetno izbjegavati.

Većinom uzroci poteškoćama planiranju putanje leže u nekarakteriziranoj okolini. Ukoliko robot koristi prethodno učitanu mapu prostora, izgledno je da je mapa nepotpuna, ne sadrži informacije o promjenama prostornih odnosa koje mogu nastupiti među objektima, također i sama metrika mape može se razlikovati od stvarnog stanja. Sljedeći problem nalazi se u činjenici da su informacije ekstrahirane iz okoline osjetilima robota ograničeno pouzdane.

Ograničen prostor očitanja u kombinaciji s varijacijama koje je nemoguće unaprijed modelirati, poput promjene intenziteta svjetla, temperature, pojave vibracija neminovno dovodi do šuma kod očitanja informacija iz okoline. Treće, stvarne okoline karakterizirane su kompleksnom i nepredvidivom dinamičkom prirodom: objekti se mogu pomicati, okolina se može mijenjati čak i u naizgled stabilnim detaljima primjerice uslijed promjene vremenskih uvjeta ili godišnjeg doba. I četvrti problem, samo izvođenje zadaća izvršnim elementima nije potpuno pouzdano, kotač može proklizati, hvataljka može izgubiti kontakt s objektom.

Predmet ovog rada je razvoj algoritma za upravljanje jednoručnim robotskim sustavom. Promatrano sa stanovišta robotskih sustava, jednoručni robotski sustav, uz određena geometrijska ograničenja svodi se na višerobotski sustav. Pri tome jedan robot predstavlja pokretnu, dinamičku prepreku drugom robotu s

kojim dijeli radni prostor. Zadatak je pronaći takve putanje za svaki robot, koje eliminiraju pojavu kolizije među segmentima robota.

Ovaj scenarij pojavljuje se kod problema uzimanja po slučaju raspoređenih objekata iz prostora kojemu pristup imaju izvršni članovi dva robota, eng. *bin picking problem*. Ovaj se problem uspješno rješava primjenom jednog robota i stereovizijskog sustava u općem slučaju.

Slika 1.1 Dva robota koji dijele radni prostor. Laboratorij za projektiranje izradbenih i monatžnih sustava

U slučaju kada dva ili više robota imaju pristup objektima, problem se komplicira te je potrebno uvesti prioritizaciju djelovanja robota (u kojem se slučaju smanjuje vremenska učinkovitost) ili rješavati istodobno za dva robota problem planiranja putanje. Postav kod kojega se ovaj problem želi integralno riješiti, a za čije je rješenje neophodna komponenta istodobnog planiranja putanje za dva robota nalazi se u Laboratoriju za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje u Zagrebu.

1.1 Pregled dosadašnjih spoznaja

Iako jedinstvena općeprihvaćena definicija robota ne postoji, svim definicijama zajedničko je da na neki način podrazumijevaju kretanje i djelovanje robota u radnom prostoru. Očekuje se da autonoman robot ima sposobnost planiranja i izvršavanja zadataka, koji u općenitom slučaju mogu zahtijevati i kretanje u nestacionarnim i nekarakteriziranim uvjetima. Zato ne iznenađuje činjenica da je problem planiranja kretanja privukao pažnju istraživačke zajednice već u ranoj fazi razvoja robotike. Podatak da danas još ne postoji robot koji se samostalno kreće ulicom ukazuje na težinu rješavanja ovog problema u uvjetima kada se okolina značajno mijenja u realnom vremenu.

Osnovna podjela pri planiranju kretanja odnosi se na stupanj poznavanja okoline u kojoj se kretanje izvodi. Kod klasičnog planiranja temeljenog na modelu okoline pretpostavlja se da sustav planiranja ima unaprijed učitanu mapu prostora. Kod planiranja temeljenog na senzorskoj slici okolina je nepoznata, a vođenje robota temelji se samo na očitanju sa senzorskih ulaza, bez konstruiranja mape prostora. Kod stvarnih se robotskih sustava ova dva pristupa mogu kombinirati i često se kombiniraju jer razne zadaće postavljene robotu mogu uzrokovati kontakt s okolinom.

S ciljem konzistentnog korištenja terminologije, želi se ovdje definirati sljedeće pojmove: radni prostor je fizički prostor u kojem se robot nalazi i djeluje. Konfiguracija robota predstavlja položaj svih točaka robota u radnom prostoru. Skup svih konfiguracija čini konfiguracijski prostor robota, u literaturi poznat pod engleskom kraticom cspace robota, oznake $\mathcal{C}$. Najmanji broj neovisnih parametara kojima je moguće u potpunosti odrediti konfiguraciju naziva se brojem stupnjeva slobode gibanja (SSG). Slobodni prostor odnosi se na područja radnog prostora koja nisu zauzeta preprekama ili područja u $\mathcal{C}$ u kojima robot ne kolidira s preprekom, što se u literaturi naziva i slobodni cspace, ili $\mathcal{C}_f$. Putanja je krivulja u $\mathcal{C}$. Putanja je prihvatljiva ako ne rezultira kolizijama između robota i prepreka. Ukoliko se nizu točaka s putanje dodijeli dimenzija vremena, takva se struktura naziva trajektorijom.

Ovi koncepti koriste se za formuliranje osnovnog problema planiranja kretanja [6]:
„Neka je $\mathcal{A}$ kruto tijelo - robot - koji se miče u Euklidskom prostoru $\mathcal{W}$, nazvanom radnim prostorom $\mathbf{R}^N$ pri čemu je N = 2 ili 3.
Neka su $\mathcal{B}_1,\ldots,\mathcal{B}_q$ nepomični kruti objekti raspoređeni u $\mathcal{W}$. $\mathcal{B}_i$ se naziva preprekom.

Pretpostavlja se da su u danom trenutku poznati geometrija objekata $\mathcal{A}$, $\mathcal{B}_1,\ldots,\mathcal{B}_q$, kao i položaji $\mathcal{B}_i$ u $\mathcal{W}$. Dalje se pretpostavlja da ne postoje kinematska ograničenja kretanja $\mathcal{A}$.
Problem je: za danu početnu poziciju i orijentaciju i krajnju poziciju i orijentaciju $\mathcal{A}$ u $\mathcal{W}$, treba generirati putanju *p* koja određuje neprekinuti niz pozicija i orijentacija $\mathcal{A}$, izbjegavajući dodire s $\mathcal{B}_i$, počevši s početnom pozicijom i orijentacijom i završavajući s krajnjom pozicijom i orijentacijom.“

Prikazani se problem svodi na geometrijsko planiranje putanje za jedno kruto tijelo. Iako je pojednostavljen, ovaj problem je težak za rješavanje. Postoje mnoga proširenja ovog izvornog problema, kao i pristupi za rješavanje ovih proširenja. Prikazani je problem statičke prirode: informacije o položajima svih prepreka prisutne su na početku planiranja i ne mijenjaju se u bilo kojem sljedećem trenutku. Kod dinamičke varijante ovog problema, dodatne informacije o preprekama dobivaju se tijekom faze planiranja, npr. uslijed interakcije sa senzorima. U slučaju kada se okolina mijenja ili se prepreke kreću, problem postaje vremenski varijabilan. Kada se u radnom prostoru nalazi više robota, problem se naziva problemom višestrukog kretanja. Ukoliko objekti u radnom prostoru mogu mijenjati oblik, problem postaje konformabilan. Važnu podklasu konformabilnih problema čine roboti koji se sastoje od više krutih segmenata povezanih zglobovima, rotacijskog ili translacijskog karaktera i takvi se roboti nazivaju artikuliranim robotima. Robotski sustav može imati inherentne restrikcije na dostupno područje, što vodi na problem planiranja s ograničenjima. Kinematske veze ograničavaju kretanje robota ili njegovih dijelova. Ukoliko se neka ograničenja mogu ukloniti reparametrizacijom konfiguracije, takva se ograničenja nazivaju holonomnim ograničenjima. Holonomna ograničenja reduciraju broj parametara koji su potrebni za specifikaciju konfiguracije. Ona fundamentalno ne mijenjaju prirodu problema, nego samo smanjuju dimnezionalnsot prostora $\mathcal{C}$. Ograničenje koje je neintegrabilno s obzirom na parametre konfiguracije i njihove derivacije (parametre brzine) naziva se neholonomnim ograničenjem.

Računska složenost problema planiranja putanje ograničila je značajno razvoj praktičnih algoritama i njihovu primjenu [2]. S ciljem razumijevanja kompleksnosti problema planiranja kretanja i motivacije za primjenu heurističkih algoritama, potrebno je prikazati nekoliko teoretskih analiza problema planiranja kretanja. U takvim analizama, činjenica da se robot mora nalaziti u slobodnom dijelu radnog prostora reprezentirana je skupom jednakosti i nejednakosti, kojima se osigurava da bilo koji dio robota ne dodiruje ili presijeca bilo koju značajku prepreke.

1.2 Tradicionalne metode planiranja kretanja

Tradicionalni pristupi planiranju kretanja mogu se podijeliti u tri osnovne skupine [3]: *metode mape putova*, *metode dijeljenja prostora*, i *metode potencijalnog polja*.

Neka je $\mathcal{C}$ oznaka skupa svih konfiguracija robota, a $\mathcal{C}_f$ slobodan konfiguracijski prostor, odnosno podskup skupa $\mathcal{C}$ gdje robot ne kolidira s preprekama. Osnovna ideja metode mape putova je pronalaženje mreže povezane jednodimenzionalnim krivuljama - putovima koji se nalaze u $\mathcal{C}_f$. Nakon konstrukcije mape putova $\mathcal{R}$, planiranje putanje svodi se na povezivanje početne i konačne konfiguracije u $\mathcal{R}$, i pretrage $\mathcal{R}$ za pronalaženje putanje.

Metoda dijeljenja prostora odvija se u tri koraka. U prvom koraku slobodan prostor $\mathcal{C}_f$ dijeli se na skup podprostora - ćelija, koje se ne preklapaju, a čija unija je upravo $\mathcal{C}_f$. U drugom se koraku konstruira graf povezanosti koji predstavlja veze između ćelija te se graf povezanosti pretražuje. Ukoliko je pretraga uspješna, izlaz je odgovarajući niz ćelija, koji se naziva kanalom, kojim su povezane ćelija koja sadrži početnu konfiguraciju i ćelija koja sadrži konačnu konfiguraciju. Putanja se jednostavno rekonstruira iz ovog niza ćelija.
Kako bi se problem planiranja putanje uspješno riješio dvjema izloženim metodama, podrazumijeva se raspolaganje podacima o promatranom radnom prostoru unaprijed. Ovi se podaci mogu upotrijebiti za rješavanje jednog ili više problema planiranja, uz promjenu početne i konačne konfiguracije robota. Značajno i veoma poželjno svojstvo ove dvije metode je da u pravilu pokazuju svojstvo kompletnosti, odnosno, ukoliko postoji putanja koja povezuje početnu i konačnu konfiguraciju, ona će biti pronađena. Nedostatci su:

- izražena potreba za računalnim resursima, kako memorijskim tako i vremenskim i
- metode ne pokazuju dobra svojstva kod neholonomnih robota.

Kod metode potencijalnih polja robot se tretira kao čestica koja se nalazi pod utjecajem umjetnog polja potencijala $\boldsymbol{U}$. Deformacijom tog polja reflektira se narav slobodnog konfiguracijskog prostora $\mathcal{C}_f$. Potencijalno polje uobičajeno se definira kao funkcija $\mathcal{C} \rightarrow \mathbb{R}$ što predstavlja ponderiranu sumu privlačnog potencijala koji privlači robota prema krajnjoj konfiguraciji i odbojnog potencijala koji

odbija robota od prepreka. Planiranje kretanja izvodi se ponavljanim izračunom umjetne sile pod čijim se djelovanjem robot kreće.

Osnovni nedostatak ove metode je to što robot može ostati zarobljen u lokalnom minimumu potencijalnog polja, odnosno, može se dogoditi da robot dosegne konfiguraciju S pri kojoj je ponderirana suma svih potencijala nul-vektor. Kako bi se ovakve pojave svele na najmanju moguću mjeru, potrebno je odabrati odgovarajuće potencijalne funkcije s minimalnim brojem lokalnih optimuma te primijeniti metode izlaženja iz suboptimuma. Danas postoje implementirane metode planiranja temeljene na potencijalnim poljima kojima je moguće planirati putanje za robote s većim brojem stupnjeva slobode, iako je problem pronalaska funkcija bez lokalnih minimuma izuzetno težak.

1.3 Više robota u radnom prostoru

Proširenje osnovnog problema planiranja putanje je slučaj kada u radnom prostoru zajednički djeluju najmanje dva robota, ili jedan robot i najmanje jedna pomična prepreka. Kod ove vrste problema, koja se naziva dinamičkim problemom planiranja kretanja nije primjenjiv pristup planiranja statičke, vremenski neovisne putanje. Potrebno je u obzir uzeti i vremensku komponentu, koja je ireverzibilna. Dodatno se model može približiti realnoj situaciji tako da se postavljaju odgovarajuća ograničenja na maksimalnu brzinu i akceleraciju robota.

Pri razmatranju problema dinamičkog planiranja razlikuju se dvije inačice problema:

- upravljiv je jedan ili više robota uz prisutnost neupravljivih dinamičkih objekata - prepreka,
- u radnom prostoru se nalaze samo roboti te su svi upravljivi.

U ovom radu razmatrana je druga inačica, odnosno u radnom prostoru se ne nalaze neupravljive dinamičke prepreke. Točnije, razmatraju se takvi roboti koju su konačnih dimenzija, sastavljeni od pomičnog elementa povezanog rotacijskim zglobom.

Općenito se problem višerobotskog planiranja putanje može formulirati ovako: u radnom prostoru nalazi se p robota, $\mathcal{A}_i$, i=1,…,p, koji se kreću u istom radnom prostoru $\mathcal{W}=\mathbf{R}^N$, N=2 ili 3, između statičkih prepreka $\mathcal{B}_j$, j=1,…,q. Roboti se kreću neovisno jedan o drugome, ali ne mogu se u istom

trenutku nalaziti na istom mjestu. Zadatak je pronaći putanju bez kolizija za svakog robota od njegove početne do konačne konfiguracije.
Postoje dva osnovna pristupa rješavanju izloženog problema: centralizirane metode i decentralizirane metode.

Kod centraliziranog pristupa, svi roboti grupiraju se zajedno tvoreći jednog jedinstvenog, kompozitnog robota. Nakon toga, problem se svodi na planiranje gibanja u prostoru koji sadrži jednog robota. Problem je što u ovom slučaju kompozitni robot uobičajeno ima puno stupnjeva slobode. To je značajno ograničenje jer je pokazano da kompleksnost planiranja putanje raste eksponencijalno s brojem stupnjeva slobode gibanja. Razvijene su specijalne, manje vremenski zahtjevne centralizirane metode planiranja putanje za višerobotske sustave, za slučaj kada zasićenost radnog prostora nije velika.

Slika 1.2 Primjer početne (lijevo) i konačne (desno) konfiguracije za koje decentralizirani pristup ne daje zadovoljavajuće rješenje

Složenost centraliziranog pristupa dovela je do razvoja decentraliziranih metoda planiranja putanje za višerobotske sustave. Kod decentraliziranog pristupa se potpunost rješenja svjesno žrtvuje s ciljem redukcije složenosti problema. Kod osnovnog decentraliziranog pristupa, planiranje se putanje provodi u dvije razdvojene faze. U prvoj se fazi za svakog robota izvodi planiranje uzimajući u obzir samo statičke prepreke iz radnog prostora, izuzimajući potpuno druge robote. Samo izbjegavanje kolizije rješava se u drugoj fazi, modulacijom brzine robota. Brzina robota mijenja se tako, da se robot koji se kreće po putanji dobivenoj u prvoj fazi, ne dodiruje u nijednom trenutku s drugim robotima. Ova

metoda je inherentno nepotpuna, iako metode primijenjene u prvoj i drugoj fazi mogu biti potpune, što ima za posljedicu da za određene početne i konačne uvjete, za koje postoji rješenje u obliku slobodne trajektorije za *p* robota, nije ovim metodama moguće pronaći to rješenje.

Situacija prikazana na slici 1.2 općenito je kompliciran slučaj kod kojega roboti moraju značajno odstupiti od optimalnih trajektorija (najčešći je kriterij dužina putanje). Decentralizirani pristup rješavanju ovog problema najvjerojatnije nije u mogućnosti pronaći rješenje uslijed razdvojenosti faza planiranja i faze modulacije brzine.

1.4 Cilj rada

U prvom dijelu cilj rada je razviti pojednostavljeni simulacijski model te algoritam za rješavanje problema planiranja istovremenog kretanja dvaju robota koji dijele radni prostor. U slučaju kada oba robota imaju pristup određenom dijelu radnog prostora, oni ulaskom u taj prostor predstavljaju dinamičku prepreku jedan drugome.

Zahtjevi koji se pri tome postavljaju na trajektorije pronađene matematičkim algoritmom su sljedeći:

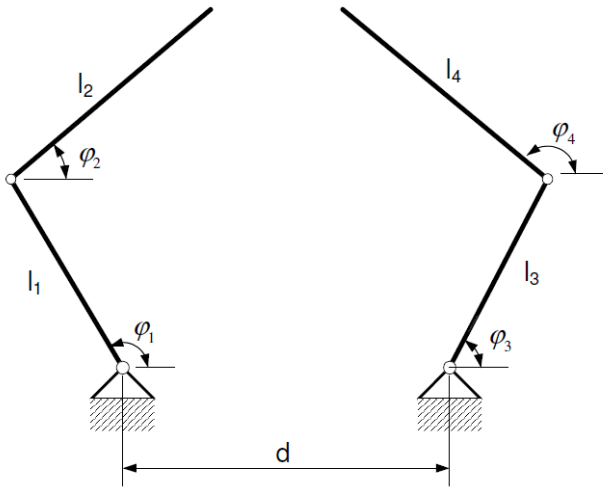
- trajektorije trebaju biti izvedive na realnom robotskom sustavu,
- trajektorije ne smiju rezultirati kolizijom bilo kojeg segmenta lijevog robota sa segmentom desnog robota ili sa statičkom preprekom u radnom prostoru,
- trajektorije moraju dovoljno dobro zadovoljavati ostale kriterije.

U drugom dijelu rada potrebno je rješenja pronađena u simulacijskom okružju implementirati na dva stvarana robota Fanuc LR Mate 200iC.

U radu se razmatra sustav sastavljen od dva robota koji su upravljani jednim upravljačkim računalom.

2. DEFINICIJA PROBLEMA

Razvijena je metodologija za planiranje kretanja dva robota 2R (2 rotacijska zgloba) konfiguracije koji dijele radni prostor i predstavljaju dinamičku prepreku jedan drugome [4]. Razvijena metodologija planiranja kretanja temeljena je na istodobnom rješavanju problema planiranja kretanja za dva robota paralelnim, dijelom neovisnim evolucijskim - koevolucijskim algoritmom.
Formulirana je funkcija dobrote i odgovarajuća metoda izbora agenata, čime je omogućena njihova paralelna evolucija i međusobna prilagodba, koja rezultira neprekinutim, vremenski i energetski učinkovitim gibanjem bez sudara od početne do konačne konfiguracije u radnom prostoru.

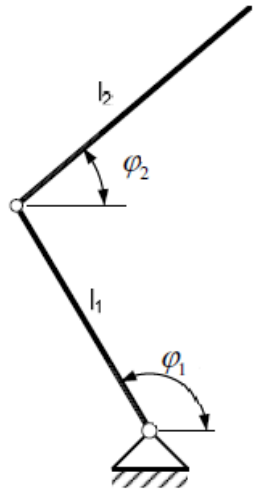


Slika 2.1 Primjer rješenja evolucijskog algoritma s dva agenta

Ukoliko se predoči SCARA konfiguracija robota, jasno je da kada se putanje oba robota prilagode tako da u horizontalnoj ravnini nema kolizije, tada niti u vertikalnoj ravnini između dva robota ne može doći do kolizije.

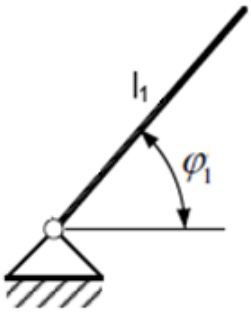
Postojanjem dvaju rotacijskih zglobova postoji mogućnost da robot dođe iza neke prepreke ili drugog robota, dok kod pojednostavljenog modela pojavit će se problemi kod nekih gibanja jer će postojati ograničenja pojednostavljene konfiguracije.

Svaki agent se sastoji od dva segmenta (l_1 i l_2), pri čemu postoje rotacijski zglobovi između donjeg segmenta svakog robota s nepomičnom podlogom (φ_1), te postoji rotacijska veza između donjeg i gornjeg segmenta robota (φ_2) (slika 2.2).



Slika 2.2 Agent s dva segmenta i dva rotacijska zgloba

Razvijen je vrlo složeni evolucijski algoritam s puno kriterija optimizacije koji iziskuje puno vremena za izračunavanje idealne putanje robota. Stoga se javlja potreba za pronalaženje jednostavnijeg rješavanja problema koji će biti pouzdaniji i jednostavniji te koji će davati rješenja putanje robota u što kraćem vremenu. Promatrajući konfiguraciju robota s tlocrta u horizontalnoj ravnini može se zaključiti kako se robot može prikazati pomoću jednog segmenta promjenjive duljine (l_1) i jednog rotacijskog zgloba između tog segmenta i nepomične podloge (φ_1) (slika 2.3).



Slika 2.3 Agent s jednim segmentom i jednim rotacijskim zglobom

S tom novom konfiguracijim smanjio se broj segmenata i broj rotacijskih zglobova s dva na jednog. Time se dobila jednostavnija konfiguracija s kojom je jednostavnije upravljati. No također se javlja

nedostatak u odnosu na predhodnu konfiguraciju s dva roracijska zgloba u tome što robot s ovom konfiguracijom ne može ići iza prepreke jer postoji samo jedan rotacijski zglob, dok se u predhodnom slučaju to moglo izvesti (slika 2.4).

Slika 2.4 Zaobilaženje prepreke

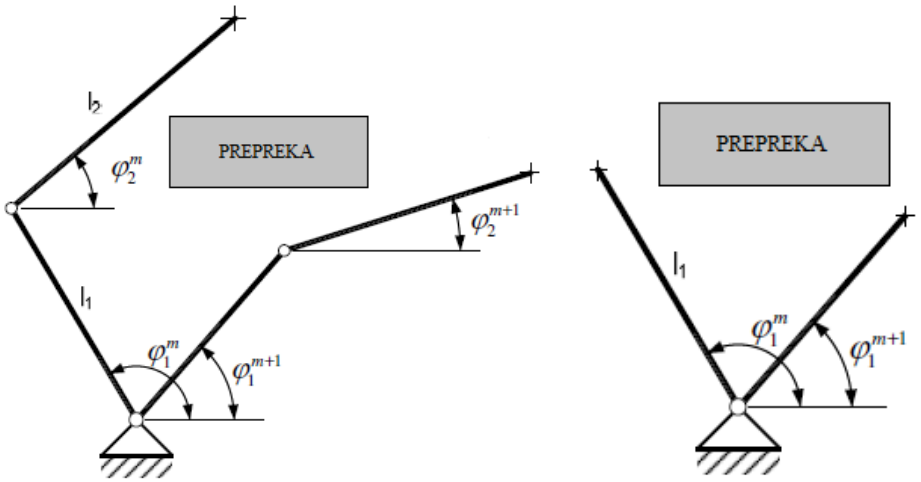
Svako među-stanje potrebno za opisivanje putanje robota bit će jedan dio kromosoma koji će predstavljati rješenje (točke) putanje.

Konfiguracija iz prvog dijela za opisivanje jednog među-stanja ima kromosom duljine l=2 i oblika $\{\varphi_1^{t=t+1}, \varphi_2^{t=t+1}\}$. Nova pojednostavljena konfiguracija za to isto stanje ima kromosom duljine l=1 i oblik $\{\varphi_1^{t=t+1}\}$.

Ovim pojednostavljenjem dobilo se puno kraće vrijeme pronalaska putanje robota primjenom evolucijskih algoritama jer se duljina kromosoma iz prvog dijela upola smanjila. To znači da ako postoji putanja robota s dva međustanja (dva položaja) robota kroz koja mora proći od starta do cilja, za prvu konfiguraciju sa dva segmenta potreban je kromosom duljine l=4, dok je za drugu konfiguraciju kromosom duljine l=2.

No kako je vrijeme pronalaska putanje robota primjenom evolucijskih algoritama dosta veliko, potrebno je pronaći algoritam koji će to vrijeme minimizirati kako bi se ostvarila kontinuirana putanja robota u stvarnom vremenu. Tim uvjetom krenulo se razmišljati kako matematički opisati konfiguraciju

robota i kako dobiti jednostavne jednadžbe koje će dati pouzdana rješenja kako ne bi došlo do kolizije prilikom preklapajućih putanja dvaju robota.



a) ***b)***

Slika 2.5 Prikaz agenta s kromosomom duljine *l*=4 (***a***) i duljine *l*=2 (***b***)

Oblik kromosoma za robot s dva segmenta (slika 2.5a):

$$\{\varphi_1^m, \varphi_2^m, \varphi_1^{m+1}, \varphi_2^{m+1}\} \quad (1)$$

Oblik kromosoma za robot s jednim segmentom (slika 2.5b):

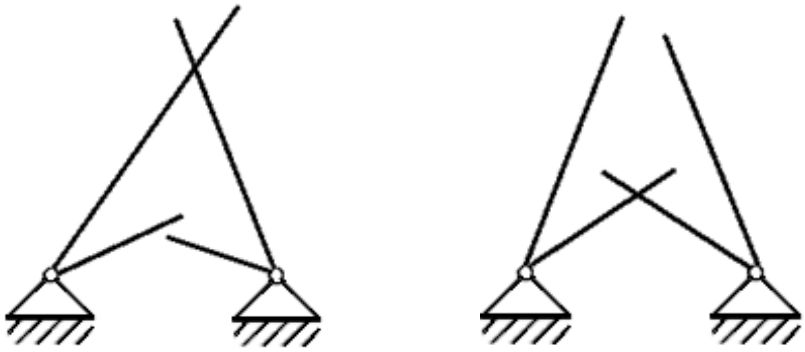
$$\{\varphi_1^m, \varphi_1^{m+1}\} \quad (2)$$

Ovim pojednostavljenjem dobiveno je rješenje za izradu pojednostavljenog simulacijskog modela za kojeg je potrebno razviti algoritam za planiranje istovremenog kretanja dvaju robota s preklapajućim radnim prostorom.

2.1 ispitivanje radnog područja

U ovom koraku potrebno je ispitati u kakvom se međusobnom odnosu nalaze prvi robot *R1* i drugi robot *R2*. Da bi se to učinilo, mora se uzeti u obzir početna i krajnja konfiguracija oba robota te provjeriti da li se sjeku početne konfiguracije robota, te isto tako ponoviti i za njihove krajnje

konfiguracije. Ukoliko se njihove početne odnosno krajnje kofiguracije sijeku, nije moguće izvesti zadatak jer se u stvarnosti roboti ne mogu nalaziti u istom trenutku jedan iza drugoga zbog jednog rotacijskog zgloba kao što je navedeno u poglavlju 2. Prikaz pozicija početne i krajnje konfiguracije kod kojih nije moguće izvršiti zadatak prikazan je na slici 2.6a i 2.6b.

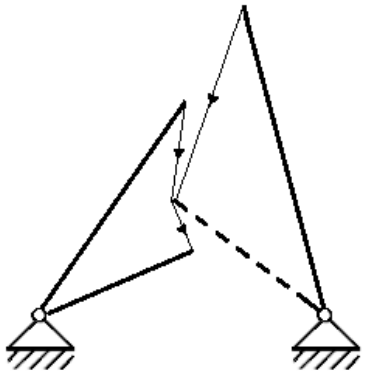


a) ***b)***

Slika 2.6 a) Prikaz agenata koji se sijeku u početnoj konfiguraciji

b) Prikaz agenata koji se sijeku u krajnjoj konfiguraciji

Ako se njihove početne i krajnje konfiguracije međusobno ne sijeku, potrebno je provjeriti da li se drugi robot nalazi u radnom prostoru prvog robota i da li se prvi robot nalazi u radnom prostoru drugog robota. Ako se dogodi jedan od ovih slučajeva, konfiguracija robota koja se nalazi u radnom području drugog robota definira se kao virtualna prepreka koju taj drugi robot mora zaobići.



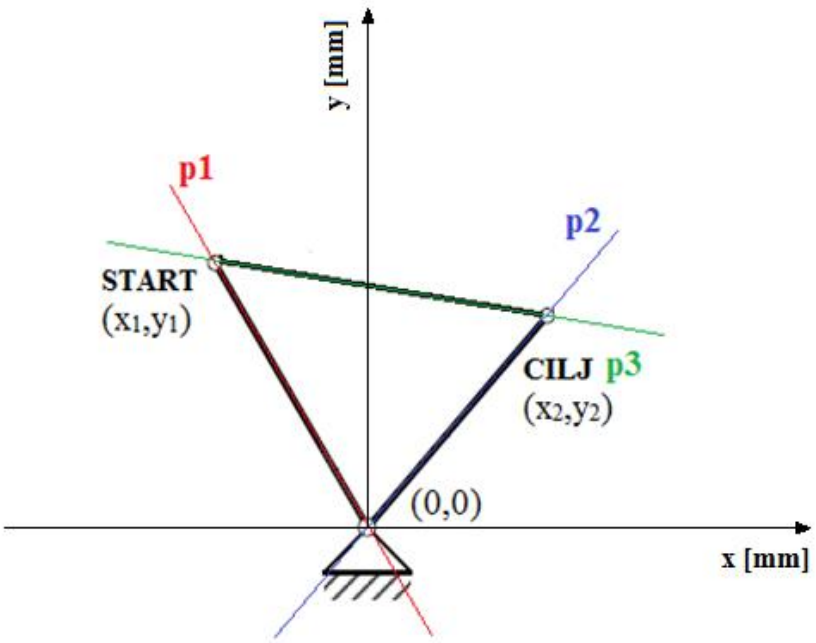
Slika 2.7 Drugi agent u radnom prostoru prvog agenta

Na slici 2.7 vidljivo je da se drugi robot sa svojom krajnjom konfiguracijom (isprekidana linija) nalazi u radnom području prvog robota. U tom slučaju krajnja konfiguracija drugog robota se definira kao prepreka koju prvi robot mora savladati. Gibanja robota na slici su prikazana tankom linijom. Prvi

robot se giba od svoje početne konfiguracije do krajnje konfiguracije drugog robota i dalje do svoje krajnje konfiguracije čime je robot zaobišao prepreku, tj. drugog robota. Drugi robot se nesmetano može gibati od svoje početne do krajnje konfiguracije.
Kako se pronalazi putanja oko prepreka opisano je u poglavlju 2.3, a detaljan matematički prikaz dan je u poglavlju 3.

2.2 Ispitivanje prepreka u radnom području robota

Sljedeći korak je uvrštavanje prepreka u radno područje robota. Treba pronaći putanju kojom neće doći do kolizije robota i prepreke. Kod prijašnjih razmatranja kolizije se provjeravaju izračunom položaja sjecišta pravaca koje tvore bridovi prepreke i referentnog niza pravaca koji definira dvije uzastopne konfiguracije. Kod toga se pojavljuje veliki broj pravaca što povećava vrijeme izračuna putanje. Uz to se javlja još i problem ukoliko je prepreka vrlo mala, niz pravaca može mimoići tu prepreku.



Slika 2.8 Prikaz radnog područja robota omeđenog pravcima

Da bi se riješio problem velikog broja pravaca i mimoilaženja prepreka, matematičim opisivanjem radnog područja robota uz pomoć tri pravca koji definiraju radni prostor gibanja robota, može se

definirati niz graničnih uvjeta (slika 2.8). Nizom uvjeta može se ispitati da li se prepreka nalazi unutar radnog područja robota, izvan ili da li možda prepreka presjeca putanju i prema tome odlučiti kao izvesti zadatak i da li je uopće to moguće (ako prepreka sijeće putanju robota početne ili krajnje pozicije, nije moguće izvesti zadatak).

Ovako matematički opisan radni prostor robota omogućuje da se pronađe i najmanja prepreka u radnom prostoru koja može biti definirana čak kao i jedna točka te izračun putanje robota se smanjuje na samo nekoliko milisekundi. Matematički proračin kao i granični uvjeti detaljno su opisani u poglavlju 3.3 i 3.4.

2.3 Definiranje najkraće putanje robota

Matematički program za izračun najkraće putanje za dva robota podijeljen je u nekoliko podprograma koji se pozivaju ovisno o vrsti problema koji se mora riješiti.
Nakon ispitivanja u kakvom su međusobnom odnosu roboti, da li postoji presjecanje prepreka i početnih i krajnjih konfiguracija i dobivanjem rezultata o preprekama koje se nalaze unutar ili izvan radnog područja, potrebno je definirati točke kroz koje se giba robot od svoje početne do krajnje konfiguracije.

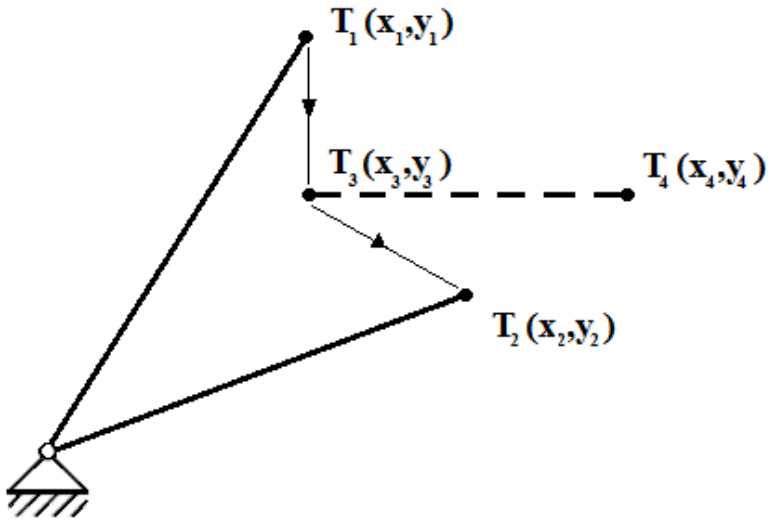
Nizom pokušaja oko definiranja tih točaka, spoznajom da robot uvijek može ići samo do prepreke, a ne i iza nje zbog jednog rotacijskog zgloba, došlo se do zaključka da krajnje točke prepreke predstavljaju točke kroz koje se smije kretati robot, a da ne dođe do njihove kolizije.

Konfiguracija robota u početnom i konačnom trenutku nam je poznata i definirana je točkom $T_1(x_1,y_1)$ koja predstavlja kraj početne konfiguracije i točkom $T_2(x_2,y_2)$ koja predstavlja kraj krajnje konfiguracije. Prepreka je definirana krajnjim točkama. Prva točka prepreke uvijek će biti ona točka koja ima manju vrijednost x koordinate.

Ukoliko se prepreka nalazi unutar radnog područja robota i spoznajom da su od važnosti samo krajnje točke prepreke, kod ispitivanja pozicija prepreka ispituju se samo krajnje točke svake prepreke. Ako se

te točke nalaze unutar radnog područja uzimaju se kao točke koje robot mora zaobići, a ako su izvan radnog područja ne uzimaju se u obzir u danjem ispitivanju.

Početna i krajnja konfiguracija na slici 2.9 nacrtana je punom debelom linijom, a njihove krajnje točke su $T_1(x_1,y_1)$ i $T_2(x_2,y_2)$. Prepreku u radnom prostoru predstavlja isprekidana linija s početnom $T_3(x_3,y_3)$ i krajnjom $T_4(x_4,y_4)$ točkom.



Slika 2.9 Prikaz točaka najkraće putanje robota

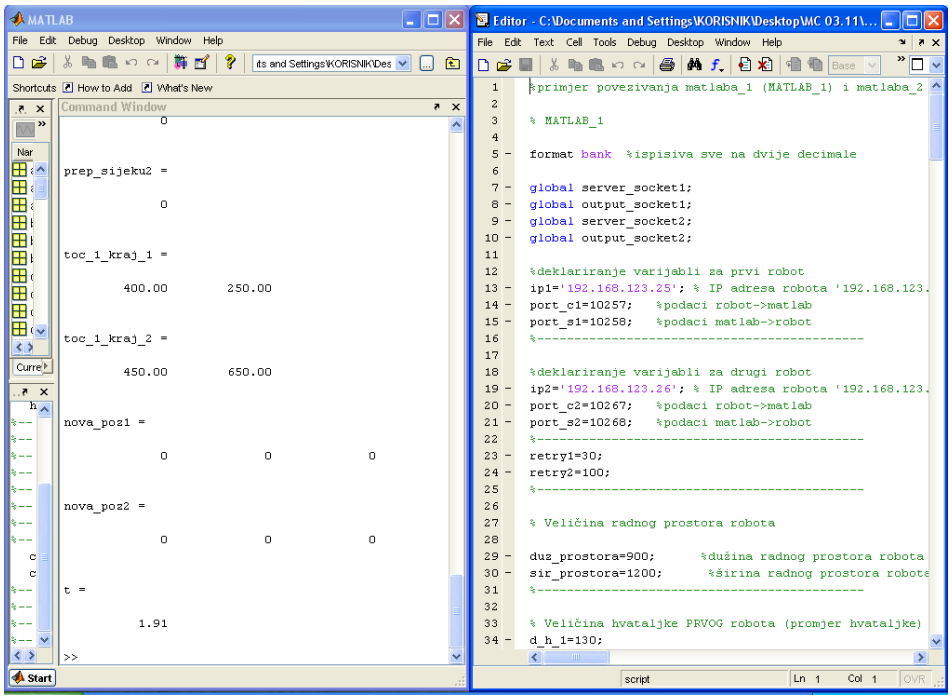
Ispitivanjem radnog područja robota dobije se da se lijeva točka prepreke $T_3(x_3,y_3)$ nalazi u radnom području robota što je vidljivo na slici 2.9. Sa predhodno navedenim zaključkom definira se krajnja točka prepreke kao točka kroz koju mora proći robot da bi zaobišao prepreku. Putanja robota od njegove početne do krajnje konfiguracije prikazana tankom linijom izražava se pomoću x i y koordinata točaka kroz koje se robot giba u obliku:

$$[x_1 \quad y_1 \quad x_2 \quad y_2 \quad x_3 \quad y_3] \tag{3}$$

Prolaskom kroz te točke robot izvršava zadatak gibanja od početne do krajnje konfiguracije uz zaobilaženje prepreke. Kod zaobilaženja se javljaju problemi koji proizlaze zbog dimenzija robota, tj. širine segmenata. Do rješenja tih problema došlo se nakon niza ispitivanja koja su provedena na stvarnim robotima i opisana su u poglavlju 3.5.

2.4 Način izvršavanja zadatka

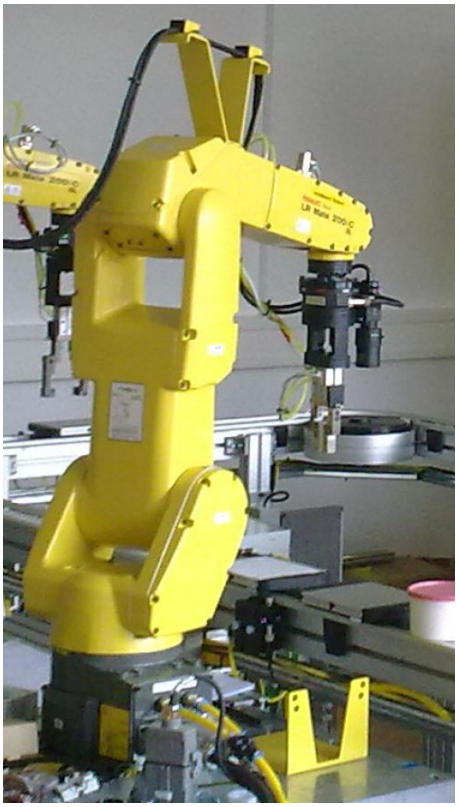
Za pronalaženje najkraće putanje robota te zaobilaženje prepreka od početne do krajnje konfiguracije potrebno je naći matematičko rješenje kroz koje se točke treba gibati robot. Za taj dio rada koristi se programski jezik Matlab (slika 2.10). Kreiranjem vlastitih programa i funkcija koje analiziraju i obrađuju primljene podatke od robota uspješno je napravljen algoritam za beskonačan broj pozicija u kojima se mogu susreti dva robota u zajedničkom radnom području. Uz glavni program u kojemu se primaju i šalju informacije robotu, poziva se trideset podprograma za rješavanje problema najkraće putanje. Time je znatno ubrzano pronalaženje najkraće putanje jer se ne izvršava jedan program već pojedini podprogrami koji ispunjavaju uvjete u kojima radi robot. Funkcija pronalaženja najkraće putanje svedena je na manje od pola sekunde što omogućuje robotu da radi u realnom vremenu.



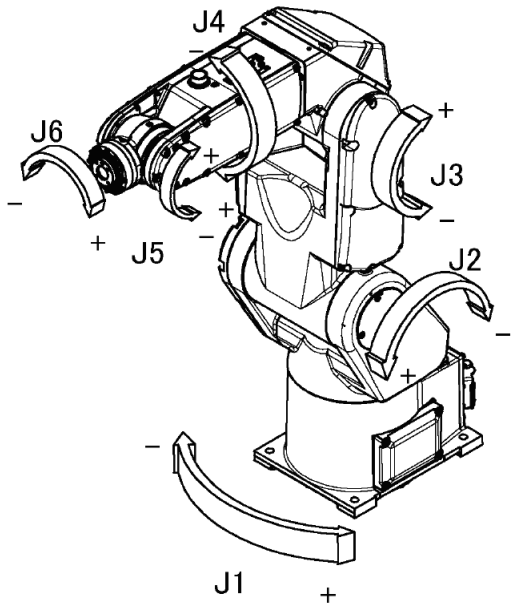
Slika 2.10 Sučelje programskog jezika Matlab

Kako bi se rješenja pronađena u simulacijskom okružju u programskom jeziku Matlab mogla konačno potvrditi, u drugom dijelu rada ta rješenja implementirana su na opremi dostupnoj u Laboratoriju za projektiranje izradbenih i montažnih sustava. Odabrana su dva robota s po šest stupnjeva slobode gibanja proizvođača Fanuc, model LrMate 200iC (slika 2.11).

Obzirom da su ovo roboti s više stupnjeva slobode gibanja nego što se razmatra u razvijenom modelu, svi zglobovi osim zglobova *J*1 držani su fiksnima, tako da se implementacija odvija u horizontalnoj ravnini.



Slika 2.11 Fanuc LR Mate 200iC



Slika 2.12 Shematski prikaz šest-osnog upravljačkog sklopa (robota) – LR Mate 200*i*C5L

Roboti su opremljeni senzorima momenta i sile, kamerama i odgovarajućom programskom podrškom primjenom koje je moguće ispitivati različite scenarije. Oba robota komuniciraju s jednim računalom. Pokretanje i zaustavljanje robota vrši se međusobnom komunikacijom preko digitalnih ulaza/izlaza. Postoji pet zajedničkih digitalnih ulaza/izlaza preko kojih mogu roboti komunicirati.

Kada je spreman za rad prvi robot šalje informaciju preko digitalnog izlaza drugom robotu. Kada drugi robot pimi taj signal na digitalni ulaz, vraća prvom robotu signal da je primio njegovu poruku. Tada može krenuti u rad drugi robot, a kada stigne odgovor prvom robotu, kreće i prvi robot.

Osnovne tehničke karakteristike robota Fanuc LR Mate 200iC dane su u tbalici 1.

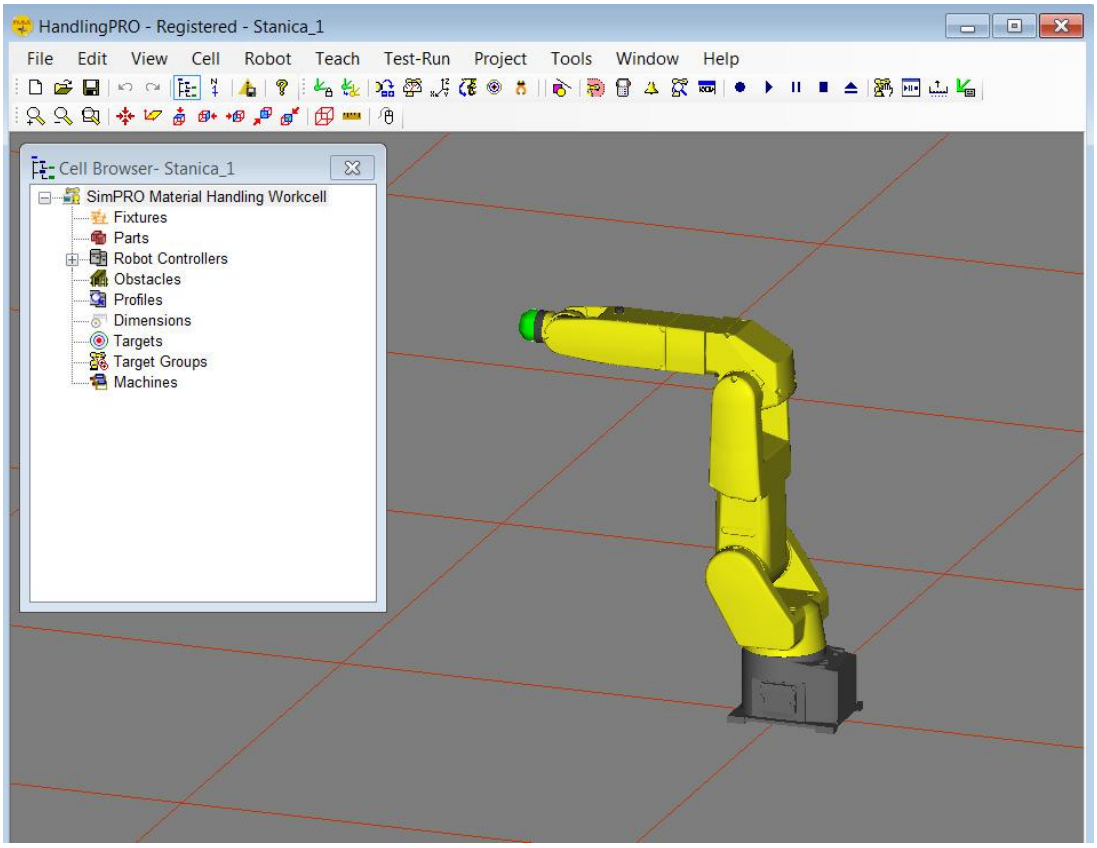
Tablica 1 Osnovne tehničke karakteristike robota Fanuc LR Mate 200iC

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/s)
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)

Programiranje Fanuc robota izvršeno je preko simulacijskog paketa Roboguide (slika 2.31). Roboguide je simulacijski paket koji služi za *offline* programiranje Fanuc robota te donosi dodatne funkcionalnosti u odnosu na rad sa upravljačkom konzolom:

- izrada te testiranje programa na osobnom računalu bez potrebe za zaustavljanjem rada robota
- rad većeg broja ljudi bez potrebe za velikim brojem realnih robota
- simulacija te optimiranje pozicija robota u radnom prostoru u svrhu smanjenja trajanja pojedinog ciklusa
- pisanje programa u proceduralnom programskom jeziku KAREL

Unutar Rooguide-a nalazi se virtualni sustav koji sadrži virtualne mehaničke jedinice robota kao i njihove upravljačke jedinice te upravljačke konzole.



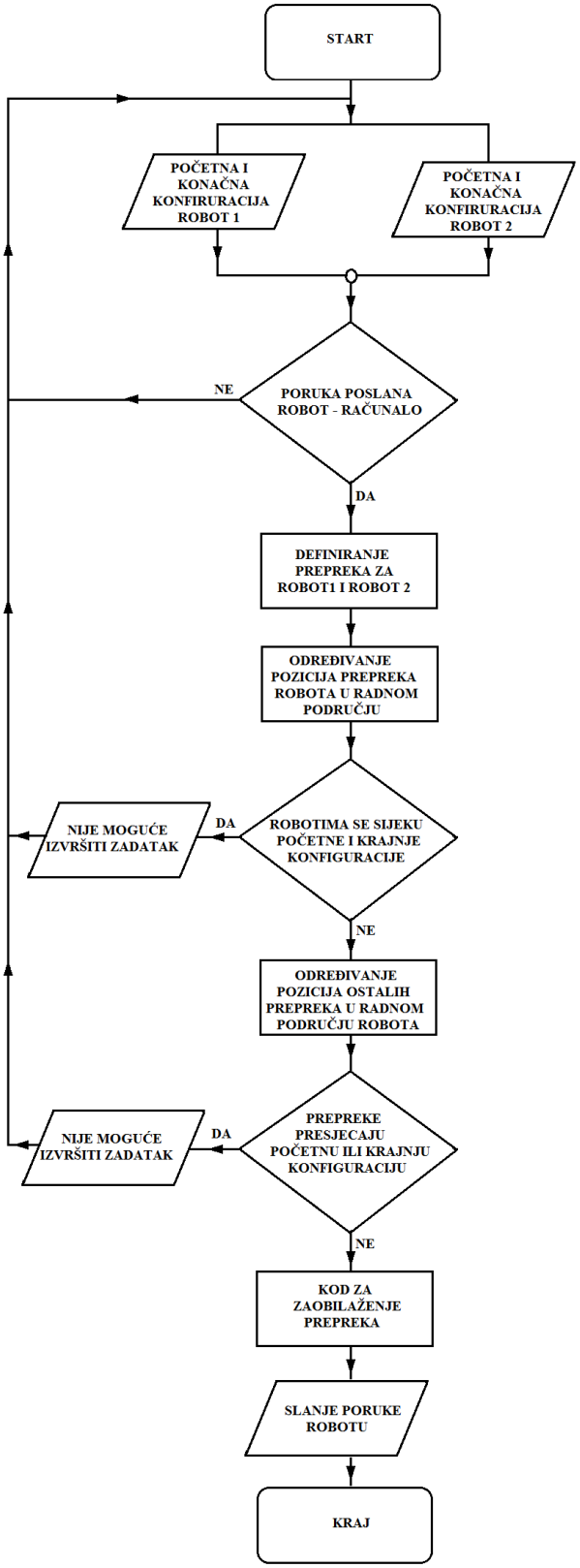
Slika 2.13 Sučelje programskog paketa Roboguide

Proceduralni jezik koji služi za programiranje Fanuc robotskih struktura je Karel. Jezik upravljačke konzole suviše je simplificiran za izradu kompliciranih programskih struktura. Sintaksa programskog jezika vrlo je slična programskom jeziku Pascal.

Nakon izrade programa pritiskom na tipku *Build* pojavljuje se obavijest ukoliko nije bilo grešaka u sintaksi. U pregledniku robotske stanice nalazi se nova datoteka sa ekstenzijom .pc. Ovo je datoteka koju je potrebno eksportirati u upravljačku jedinicu realnog robota. Pritiskom na tipku *Export* te zatim izborom željenog robota u robotskom susjedstvu (*Robot Neighborhood*) te potvrdom na tipku *To Robot* označena .pc datoteka kopira se na upravljačku jedinicu realnog robota.

Pokretanje programa iz upravljačke jedinice robota moguće je pozivom KAREL programa iz TPP.

3 METODA RJEŠAVANJA PROBLEMA

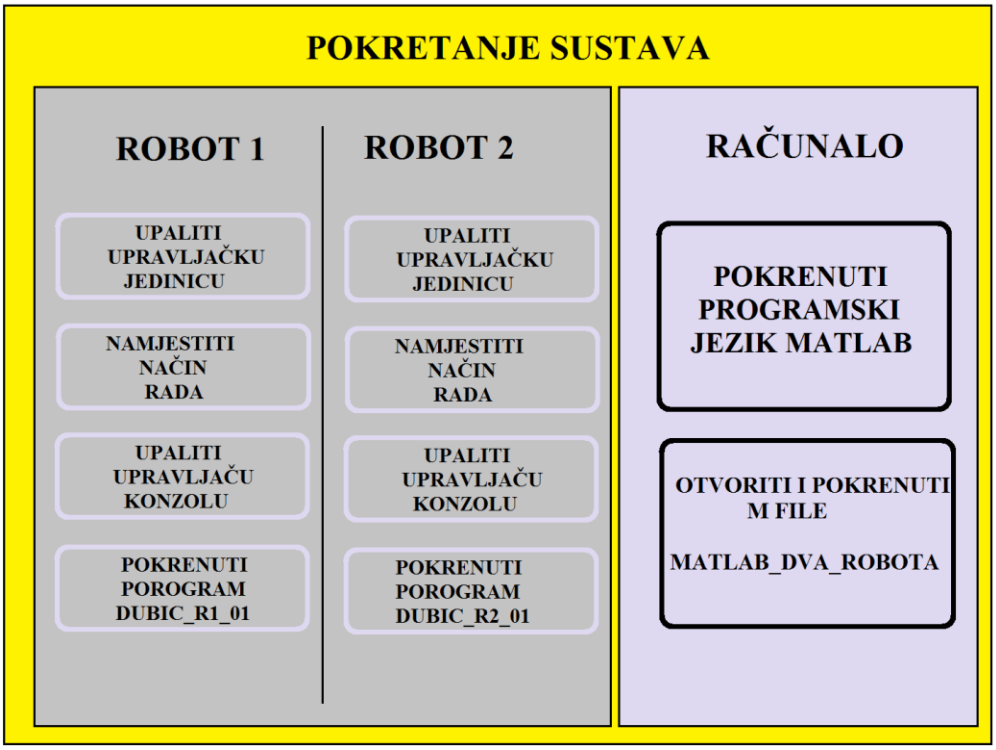


Slika 3.1 Dijagram toka

Kao što je navedeno u poglavlju 2.4 u prvom dijelu diplomskog rada izrađen je pojednostavljeni simulacijski model te algoritam za rješavanje problema planiranja istovremenog kretanja dvaju robota koji dijele radni prostor u programskom jeziku Matlab. Zatim je izrađen program u simulacijskom paketu Roboguide gdje su napravljena testiranja za razne pozicije robota na virtualnom sustavu koji sadrži virtualne mehaničke jedinice robota. Nakon toga sljedilo je povezivanje upravljačke jedinice robota i računala preko TCP/IP protokola. Time su rješenja pronađena u simulacijskom okružju implementirana na stvarne robote Fanuc LR Mate 200iC.

3.1 Start - pokretanje sustava

Prvi element u dijagramu toka je START. On označava pokretanje sustava - programskog jezika Matlab i dva stvarna robota Fanuc LR Mate 200iC.

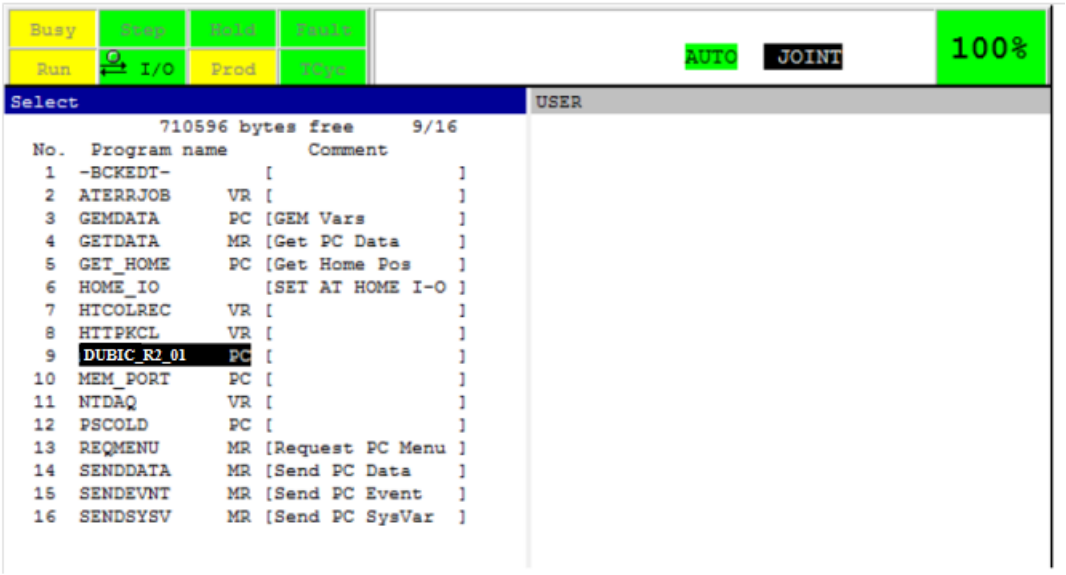


Slika 3.2 Redosljed pokretanja sustava

Kod pokretanja sustava prvo moramo upaliti upravljačke jedinice robota i pokrenuti program, a zatim računalo i programski jezik Matlab zbog toga što povezivanje istih je takvo da prvo roboti traže povezivanje od matlaba, a nakon toga matlab s robotima.

Pokretanje robota vrši se u nekoliko koraka:

1. upaliti upravljačku jedinicu i provjeriti da li je gljiva na upravljačkoj jedinici izvučena
2. odabrati način rada: T1 - maksimalna brzina do 250 mm/s
   T2 - maksimalna brzina
   AUTO - maksimalna brzina, automatski način rada
3. upaliti upravljačku konzolu i provjeriti da li je gljiva na upravljačkoj konzoli izvučena
4. prekid postojećeg programa - izvršava se tipkom Fctn te izborom Abort all
5. poništenje grešaka - kombinacija tipki SHIFT + Reset
6. program mora biti označen - lijeva tipka miša te tipka ENTER nakon pritiska na tipku *Select* upravljače konzole



Slika 3.3 Upravljačka konzola spremna za izvođenje programa

Pritiskom na tipku FWD na upravljačkoj konzoli pokreće se program. Program traži uspostavljanje veze s računalom. Nakon što se pokrenu programi na oba robota, potrebno je pokrenuti program u Matlabu. Čim se pokrene program u Matlabu, vrši se povezivanje upravljačke jedinice robota i računala.

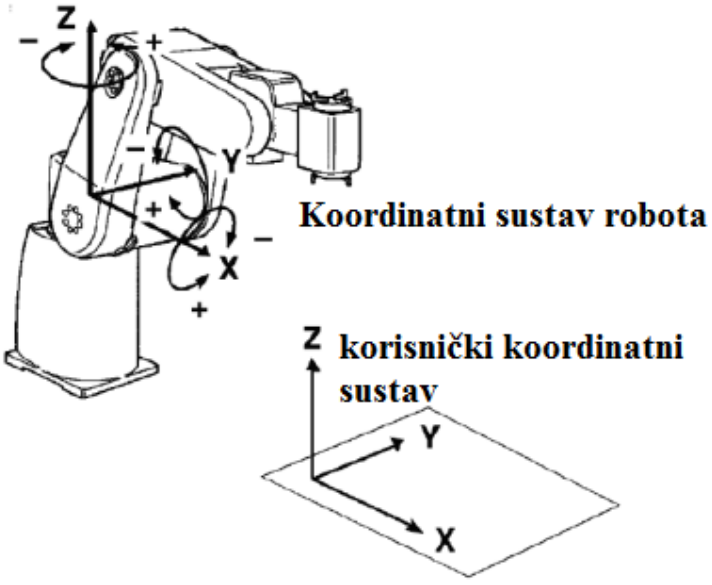
Na ekranu upravljačke konzole se ispisuje da je izvršeno povezivanje te da robot treba postaviti u početni položaj. Kada se robot postavi u početni položaj, pritiskom na tipku 1 na upravljačkoj konzoli,

izvršava se algoritam za rješavanje problema planiranja putanje za oba robota. Nakon što se dobiju rezultati u Matlabu, vraćaju se robotima i roboti izvšuju svoja gibanja.
Za prekid programa treba pritisnuti tipku 2 na upravljačkoj konzoli.

Da bi roboti bili spremni za izvršavanje programa, potrebno je u programu definirati globalni koordinatni sustav, koordinatni sustav robota, koordinatni sustav alata, način komuniciranja robotske upravljačke jedinice i računala, način komuniciranja između dva robota prilikom kretanja i zaustavljanja istih.

3.1.1 Definiranje koordinatnih sustava za robote Fanuc LR Mate 200iC

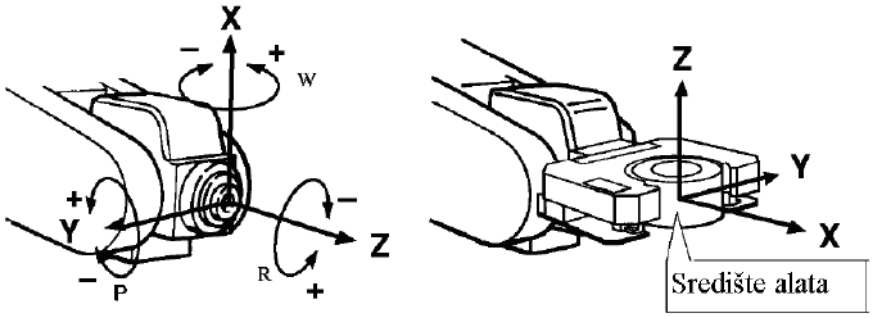
Koordinatni sustav robota
Po standardnim postavkama koordinatni sustav robota preddefiniran je u određenoj točci tijela robota. Ovaj koordinatni sustav je kartezijski, pravokutni, desnokretni koordinatni sustav. Svaki korisnički koordinatni sustav definira se u odnosu na ovaj osnovni koordinatni sustav 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 3.4.



Slika 3.4 Koordinatni sustav robota, korisnički koordinatni sustav

Koordinatni sustav alata

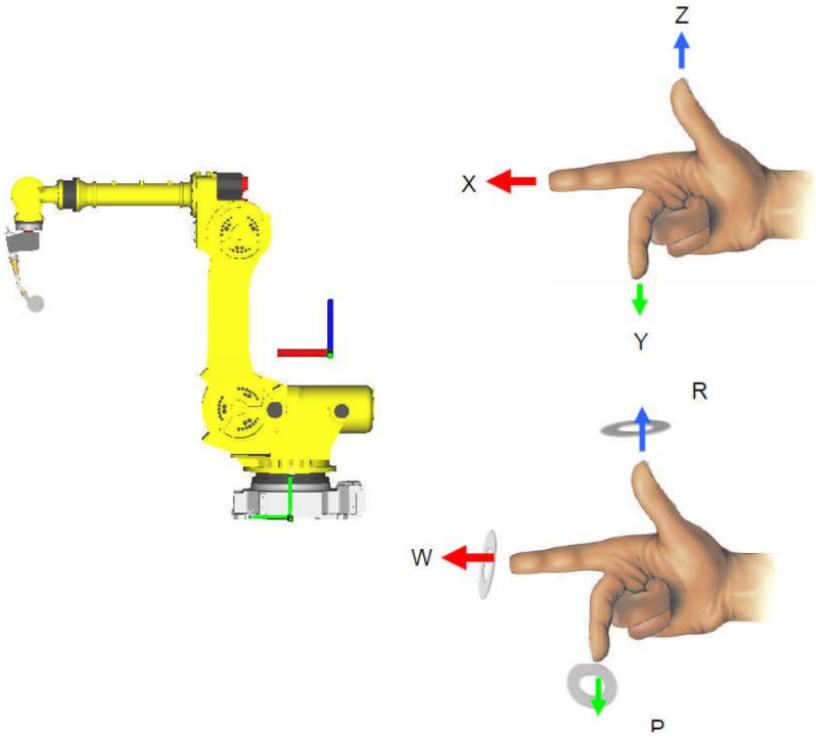
Ovisno o alatu kojeg robot koristi (npr. različite hvataljke) mora se definirati koordinatni sustav alata kako bi vrh alata uvijek izvršavao željeno gibanje. Analogno preddefiniranom koordinatnom sustavu svijeta robota postoji preddefiniran koordinatni sustav prihvata alata. On se nalazi kako je prikazano na slici u osi šestog zgloba. Prema ovoj točci 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) ***b)***

Slika 3.5 a) Točka prihvata alata ***b)*** Koordinatni sustav središta alata

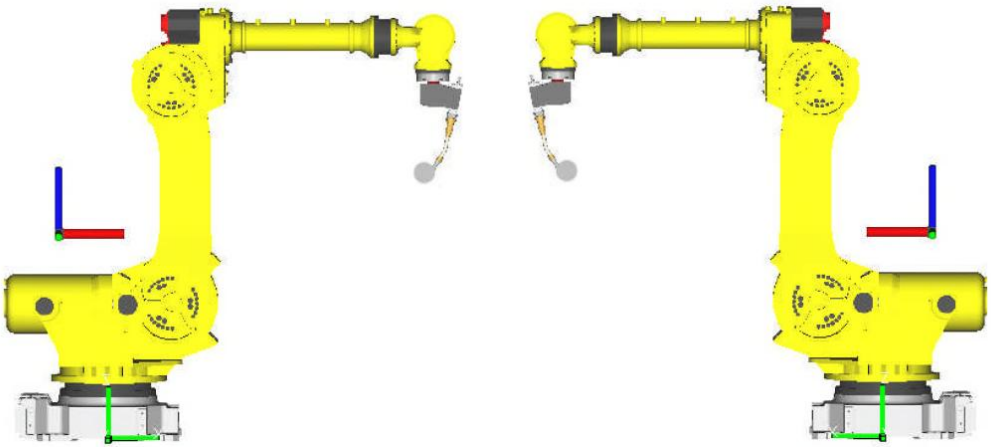
Pravokutni koordinatni sustav možemo u prostoru jednostavno predočiti pravilom desne ruke.



Slika 3.6 Koordinatni sustav - pravilo desne ruke

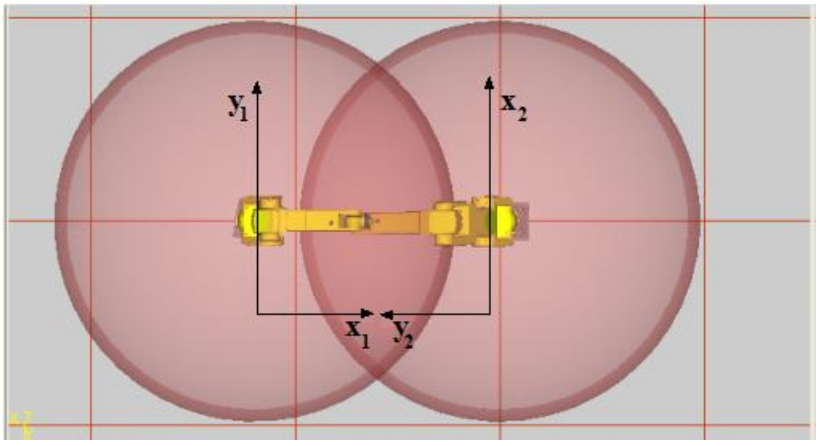
Dva robota djele isti radni prostor te je definiran globalni koordinatni sustav unutar kojeg se nalazi radno područje prvog (lijevog) i drugog (desnog) robota (slika 3.7). Udaljenost između robota predstavlja širinu radnoga područja i udaljenost između ishodišta koordinatnog sustava lijevog i desnog robota.

Kako su koordinatni sustavi oba robota kartezijski, pravokutni, desnokretni, koordinatni sustav desnog robota moramo zarotirati za 90° oko osi *z* kako bi dobili poklapanje koordinatnog sustava robota s globalnim koorinatnim sustavom.



Slika 3.7 Međusobni položaj lijevog i desnog robota

Koordinatni sustav lijevog robota x_1-y_1 poklapa se s globalnim koordinatnim sustavom *X*-*Y*, dok je koordinatni sustav desnog robota zarotiran za 90° i njegova x_2 os se poklapa s *X* osi globalnog koordinatnog sustava, ali je drugog smijera, dok se y_2 os poklapa s globalnom *Y* os (slika 3.8).



Slika 3.8 Koordinatni sustavi robota

Koordinatne sustave alata potrebno je zakrenuti zbog kamera koje se nalaze na zadnjem segmentu kako nam ne bi smetale prilikom mimoilaženja robota s preprekama.

Koordinatni sustav lijevog robota:

x_1= 175 mm

y_1= -600 mm

z_1= 50 mm

w_1= 0 °

p_1= 0 °

r_1= 0 °

Koordinatni sustav alata lijevog robota:

x_1= 0 mm

y_1= 0 mm

z_1= 228 mm

w_1= 180 °

p_1= 0 °

r_1= -180 °

Koordinatni sustav desnog robota:

x_2= 175 mm

y_2= 600 mm

z_2= -25 mm

w_2= 0 °

p_2= 0 °

r_2= -90 °

Koordinatni sustav alata desnog robota:

x_1= 0 mm

y_1= 0 mm

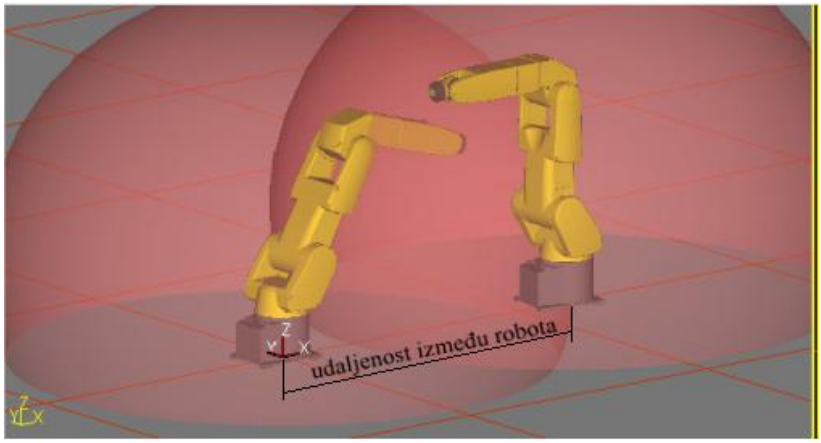
z_1= 315 mm

w_1= 180 °

p_1= 0 °

r_1= -90 °

Udaljenost između robota = širina radnog područja = 900 mm



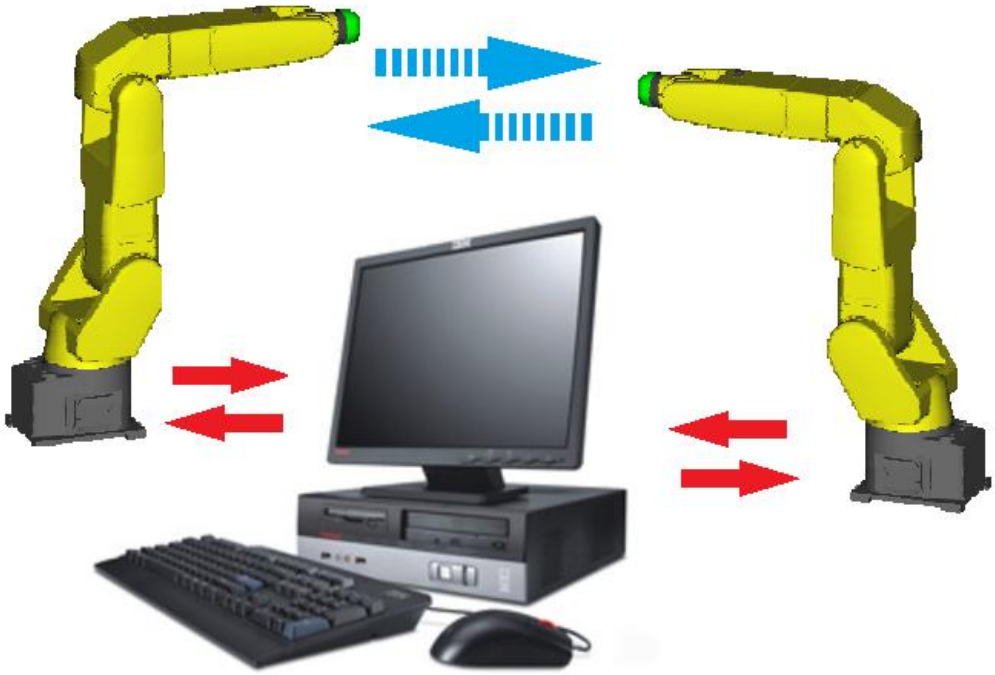
Slika 3.9 Udaljenost između robota

Udaljenost između robota nam prestavlja u algoritmu za pronalaženje putanje robota širinu radnog područja odnosno maksimalnu vrijednost *x* koordinate u globalnom koorinatnom sustavu.

Prilikom računanja i pretvorbe iz lokalnog u globalni koordinatni sustav, vrijesnoti *x* i *y* koordinata za lijevi robot ostaju jednake u globalnom koordinatnom sustavu, dok se *x* koordinata desnog robota pretvara u *y* koordinatu globalnog kordinatnog sustava, a *y* koordinata u *x* koordinatu globalnog koordinatnog sustava. Pošto se desni robot nalazi na kraju globalnog koordinatnog sustava gledajući po osi *x*, da bi dobili globalnu vrijednost *x* kooridnate za desni robot, potrebno je od udaljenosti između robota oduzeti vrijednost *x* kooridnate desnog robota.

3.1.2 Povezivanje robota i računala

Komunikacija između upravljačke jedinice robota i računala izvršena je preko TCP/IP protokola.



Slika 3.10 Komunikacija između upravljačke jedinice robota i računala

Upravljačka jedinica šalje pozicije početne i krajnje konfiguracije preko IP adrese.
IP adresa desnog robota: 192.168.123.26
IP adresa lijevog robota: 192.168.123.25
IP adresa računala: 192.168.123.40

Napravljena su dva komunikacijska kanala. Prvi komunikacijski kanal služi za slanje podataka sa robota na računalo. Robot u tom slučaju ima ulogu servera, dok računalo ima ulogu klijenta i prima podatke. Drugi komunikacijski kanal služi za slanje odgovora s računala na upravljačku jedinicu robota. Računalo je server, a robot je klijent i prima podatke od računala.
Poruka koja se šalje na računalo ima oblik:

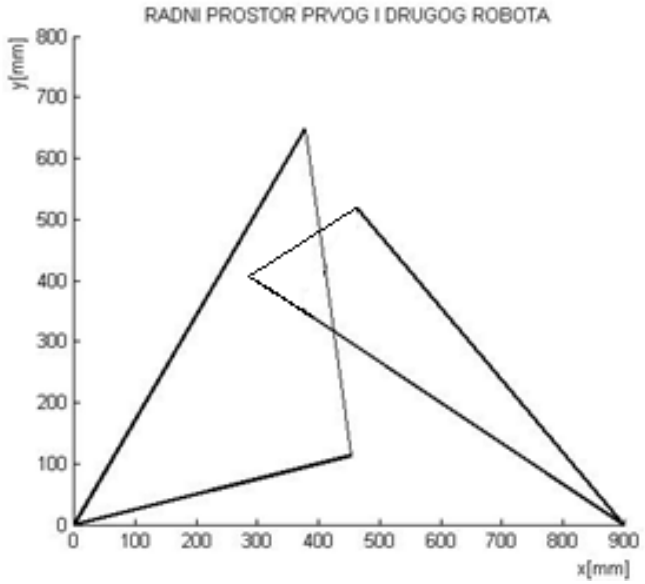
$$[prepreke\_sijeku \;\; broj\_pozicija \;\; x_1 \;\; y_1 \;\; x_2 \;\; y_2] \tag{4}$$

gdje je:

prepreke_sijeku - broj prepreka koje sijeku putanju
broj_pozicija - broj točaka kroz koje prolazi robot
x_1 - x koordinata početne konfiguracije
y_1 - y koordinata početne konfiguracije
x_2 - x koordinata krajnje konfiguracije
y_2 - y koordinata krajnje konfiguracije

3.2 Definiranje prepreka za robote Fanuc LR Mate 200iC

Ulaz u simulacijsko okruženje su konfiguracije robota u početnom i konačnom trenutku vremena (slika 3.11). Potrebno je ispitati u kakvom se međusobnom odnosu nalaze lijevi i desni robot.

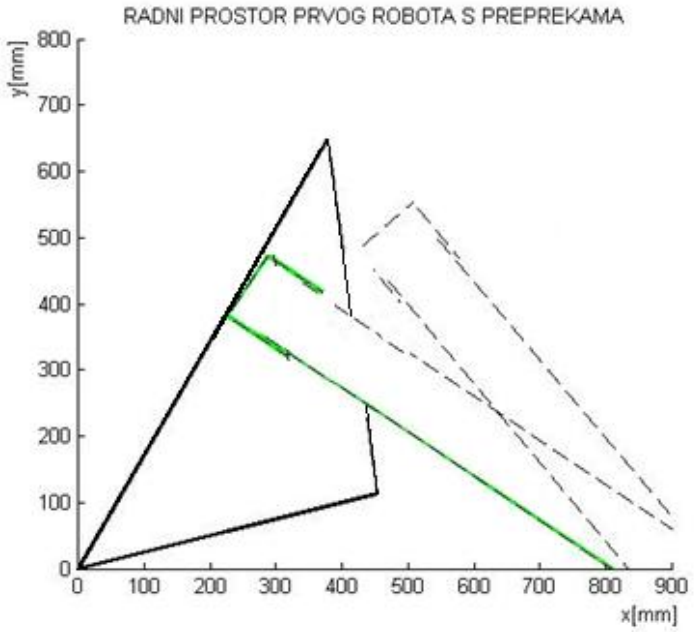


Slika 3.11 Početna i krajnja konfiguracija oba robota

Na slici je vidljivo kako su u simulacijskom modelu oba robota prikazani kao štapovi.
U ovom slučaju program bi izračunao da ne dolazi do sjecanja pravaca početne konfiguracije lijevog robota i krajnje konfiguracije desnog robota. No kada bi se u stvarnosti našla dva robota u ovakvoj situaciji došlo bi do kolizije, tj. sjecanja konfiguracija. Razlog tome je taj što robot u stvarnosti nije pravac, već ima segmente određene širine.

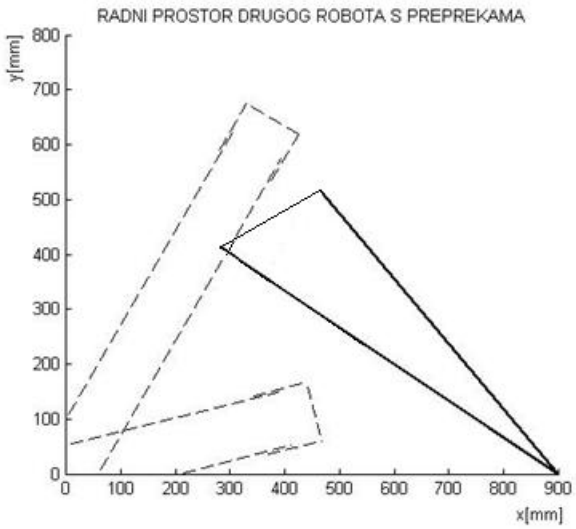
Nizom ispitivanja uvedena su dva faktora koji predstavljaju robot kao prepreku za drugi robot, a to su širina njegovih segmenata i veličina hvataljke (slika 3.12 i 3.13). S obzirom na ta dva parametra roboti predstavljaju dinamičku prepreku jedan drugome.

Desni robot kao dinamička prepreka lijevom robotu na slici je prikazan isprekidanom linijom (slika 3.12). Vidljivo je da je vrh početne i krajnje konfiguracije širi od segmenata robota zbog toga što hvataljka ima veći radijus (uz hvataljku se nalaze i kablovi od kamere, crijeva za zrak hvataljke) od širine segmenata robota. Podebljane linije desnog robota označuju da se taj dio robota nalazi u radnom području lijevog robota te će biti uzete u obzir u kodu za zaobilaženje prepreka.



Slika 3.12 Desni robot kao prepreka lijevom robotu

Ovim opisivanjem konfiguracije robota dobivamo rezultat koji se poklapa sa stvarnim vrijednostima, a to je da se robotske konfiguracije sijeku. Time smo ugradili dimenzije segmenata robota u simulacijski model čime smo dobili rješenja za realne uvjete.
Na isti način kako je desni robot definiran kao prepreka za lijevi, tako je i lijevi robot definiran kao dinamička prepreka za desni robot (slika 3.13).



Slika 3.13 Lijevi robot kao prepreka desnom robotu

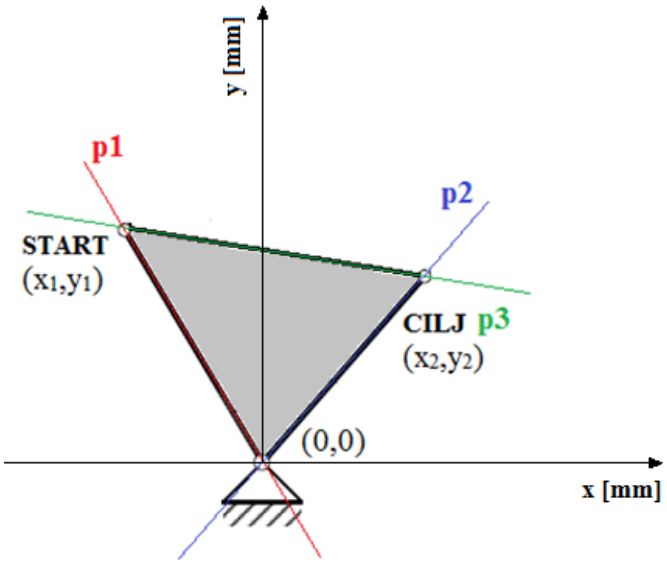
I iz ove slike je vidljivo da se početna konfiguracija lijevog robota siječe s krajnjom konfiguracijom desnog robota.

3.3 Metoda za određivanje pozicija prepreka u radnom prostoru

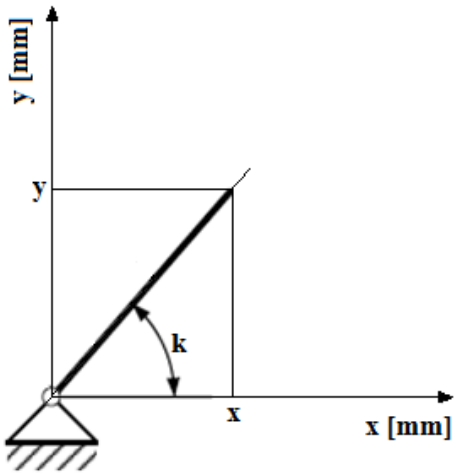
Najveći problem kod planiranja istovremenog kretanja dvaju robota s preklapajućim radnim prostorom je pronaći rješenje njihovih putanja u što kraćem vremenu uz što veću točnost rješavanja problema kako s velikom sigurnošću ne bi došlo do kolizije između njih. Ta dva parametra usmjerila su sva razmišljanja na rješavanje problema matematičkim putem koji će za svaku moguću poziciju robota uvijek dati sigurno rješenje.

Pošto su poznate početna i krajnja konfiguracija robota koje imaju određenu poziciju u radnom prostoru definiranu *x* i *y* koordinatama, kroz te točke mogu se provući pravci. Povlačenjem pravaca

kroz ishodište robota i točaka starta (početne konfiguracije) i cilja (krajnje konfiguracije) dobivena je površina koja predstavlja radni prostor robota u kojem je moguće izvršiti najkraću moguću putanju od starta do cilja (slika 3.14).



Slika 3.14 Radno područje robota omeđeno pravcima



Slika 3.15 Konfiguracija robota

Jednadžba pravca glasi:

$$y=k\cdot x+b \tag{5}$$

gdje su:

- *x* i *y* vrijednosti koordinata robota (ishodište robota, start ili cilj),
- *k* je nagib pravca,
- *b* je odsječak na *y*-os.

Radni prostor robota (putanja od starta do cilja) opisan je s tri pravca čije jednadžbe pravaca glase:

-za pravac p_1:

$$y_{p_1} = k_1 \cdot x_{p_1} + b_1 \tag{6}$$

-za pravac p_2:

$$y_{p_2} = k_2 \cdot x_{p_2} + b_2 \tag{7}$$

-za pravac p_3:

$$y_{p_3} = k_3 \cdot x_{p_3} + b_3 \tag{8}$$

Nagib pravca *k* izračunava se pomoću jednadžbe:

$$k = \tan\alpha = \frac{nasuprotna\ stranica}{prilježeća\ stranica} \tag{9}$$

Iz toga slijede jednadžbe za nagibe pravaca:

$$k_1 = \frac{y_1 - 0}{x_1 - 0} \tag{10}$$

$$k_2 = \frac{y_2 - 0}{x_2 - 0} \tag{11}$$

$$k_3 = \frac{y_1 - y_2}{x_1 - x_2} \tag{12}$$

Poznavanjem točaka starta i cilja, uz izračunate vrijednosti nagiba pravaca iz predhodnog koraka, mogu se dobiti vrijednosti odsječaka *b* iz jednadžbi:

$$b_1 = y_{p_1} - k_1 \cdot x_{p_1} = y_1 - k_1 \cdot x_1 \tag{13}$$

$$b_2 = y_{p_2} - k_2 \cdot x_{p_2} = y_2 - k_2 \cdot x_2 \tag{14}$$

$$b_3 = y_{p_3} - k_3 \cdot x_{p_3} = y_1 - k_1 \cdot x_1 = y_2 - k_2 \cdot x_2 \tag{15}$$

Sada se može za bilo koju vrijednost *x* koordinate na pravcima p_1, p_2 i p_3 pronaći odgovarajuća vrijednost *y* koordine jednadžbama:

$$y_1 = k_1 \cdot x_1 + b_1 \tag{16}$$

$$y_2 = k_2 \cdot x_2 + b_2 \tag{17}$$

$$y_3 = k_3 \cdot x_3 + b_3 \tag{18}$$

Dobivene jednadžbe predstavljaju granične vrijednosti koje se koriste kod postavljanja uvjeta za izračuvavanje da li se drugi robot, odnosno bilo koja druga prepreka nalazi unutra radnog prostora robota.

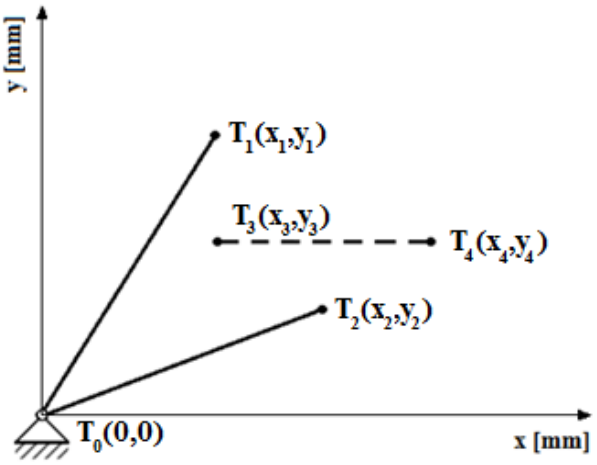
Ovako matematički opisan radni prostor omogućuje da se pronađe i najmanja prepreka u prostoru koja može biti definirana čak kao jedna točka. Time je ispunjen uvijet da se jednostavnim matematičkim izrazom, velikom brzinom, s velikom sigurnošću i točnošću pronađu prepreke izvan ili unutar radnog prostora.

3.4 Granični uvjeti

Kod kretanja dvaju robota s preklapajućim radnim prostorom može se reći da jedan robot drugome predstavljaju prepreke koje oni moraju zaobilaziti. Postoji niz mogućnosti u kakvim se međusobnim odnosima mogu naći roboti, tj. robot i prepreke.

Niz tih mogućnosti podijeljen je u pet grupa, a to su:

1. prepreka sijeće pravce početne i krajnje konfiguracije
2. prepreka sijeće pravac početne konfiguracije
3. prepreka sijeće pravac krajnje konfiguracije
4. prepreka se nalazi između pravaca početne i krajnje konfiguracije
5. prepreka se nalazi izvan pravaca početne i krajnje konfiguracije



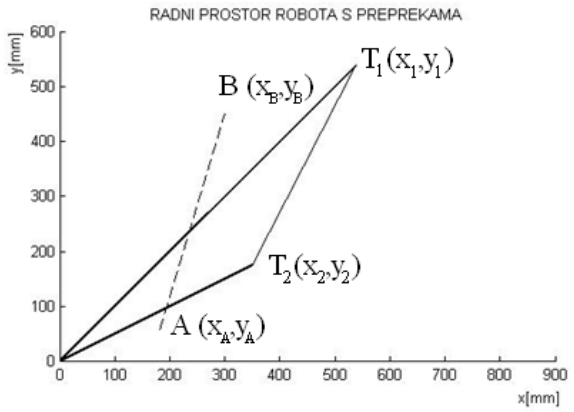
Slika 3.16 Karakteristične točke

Svi granični uvjeti proizlaze iz međusobnog položaja prepreke koju čine točka $T_3(x_3,y_3)$ (početak prepreke) i točka $T_4(x_4,y_4)$ (kraj prepreke) u odnosu na radno područje robota kojeg sačinjavaju pravci p_1, p_2 i p_3 s karakterističnim točkama $T_1(x_1,y_1)$ (počrtna konfiguracija) i $T_2(x_2,y_2)$ (krajnja konfiguracija) i $T_0(0,0)$ (ishodište robota).

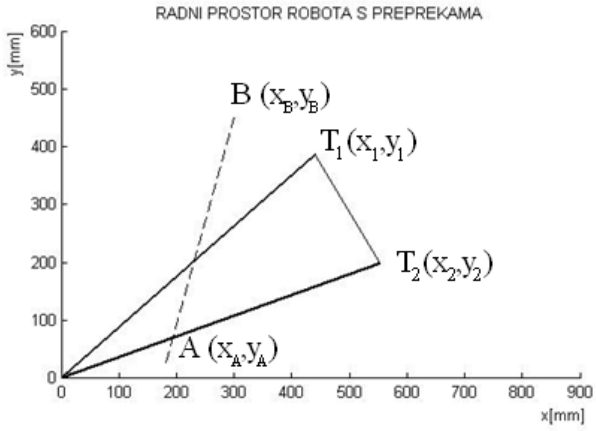
U nastavku teksta slijedi prikaz svih pozicija i objašnjenje graničnih uvjeta koji se koriste u programskom jeziku Matlab za rješavanje problema planiranja istovremenog kretanja dvaju robota.

3.4.1 Prepreka sijeće pravce početne i krajnje konfiguracije

Jedna od pet grupa mogućnosti gdje se javlja presjecanje prepreke i konfiguracije robota je ta da prepreka presijeca i početnu i krajnju konfiguraciju. Postoji dvadeset rzličitih pozicija prepreke u odnosu na robot gdje dolazi do presjecanja pravca početne i krajnje konfiguracije. Prepreka sijeće oba pravca ako je dio prepreke ispod krajnje konfiguracije, a dio prepreke iznad početne konfiguracije.
Oznaka $y_{pA\rightarrow 1}$ označuje y vrijednosti pravaca povučenog iz točke A kroz točku T_1.



Slika 3.17 Granični uvjet 1



Slika 3.18 Granični uvjet 2

Granični uvjet 1:

$x_1 > x_2$

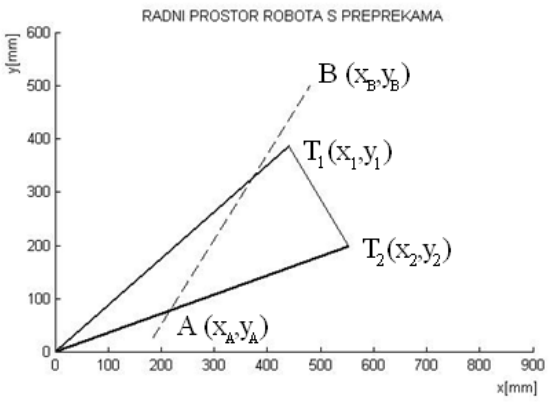
$x_A < x_2$

$y_B > y_{pA\rightarrow 1}$

Granični uvjet 2:

$x_1 < x_2$

$x_B < x_1$



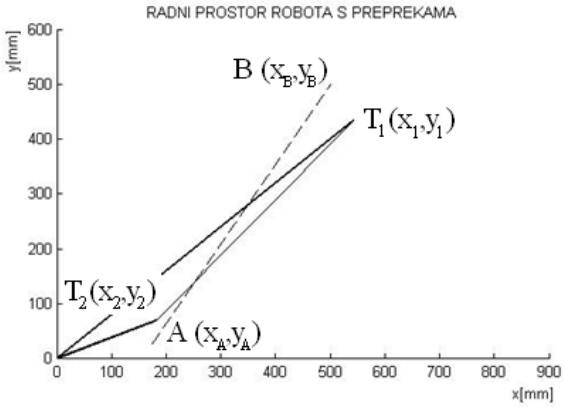
Slika 3.19 Granični uvjet 3

Granični uvjet 3:

$x_1 < x_2$

$x_B > x_1$

$y_A > y_{pB \to 1}$



Slika 3.20 Granični uvjet 4

Granični uvjet 4:

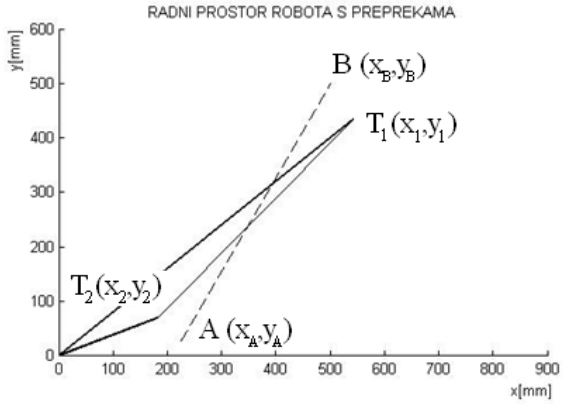
$x_1 > x_2$

$x_A < x_2$

$y_B < y_{pA \to 2}$

$y_B > y_{pA \to 1}$

$y_A < y_{pB \to 2}$



Slika 3.21 Granični uvjet 5

Granični uvjet 5:

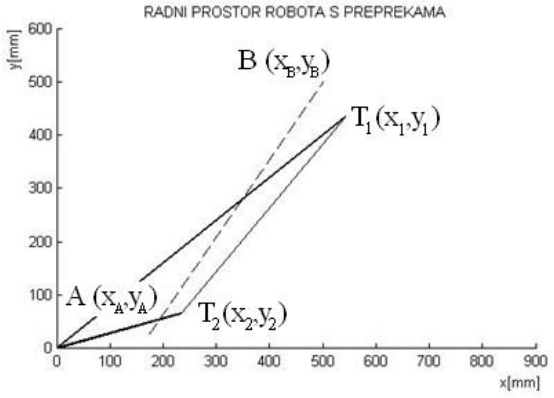
$x_1 > x_2$

$x_A > x_2$

$y_B > y_{pA \to 2}$

$y_B > y_{pA \to 1}$

$y_A < y_{pB \to 2}$



Slika 3.22 Granični uvjet 6

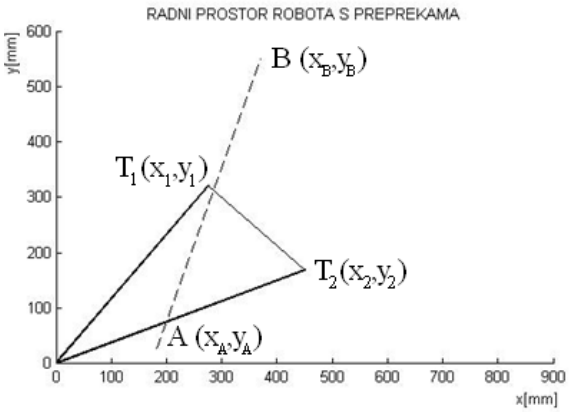
Granični uvjet 6:

$x_1 > x_2$

$y_A > y_{pB \to 2}$

$y_B < y_{pA \to 1}$

$y_B > y_{pA \to 2}$



Slika 3.23 Granični uvjet 7

Granični uvjet 7:

$x_1 < x_2$

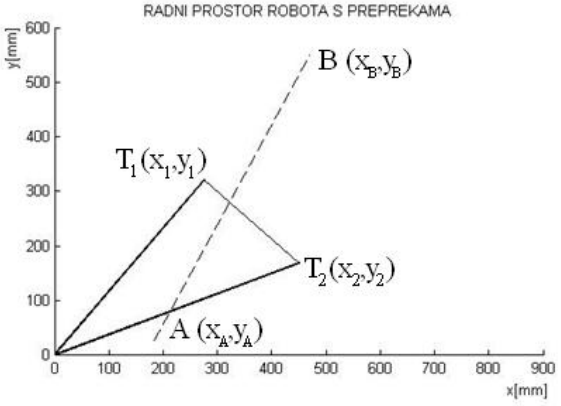
$x_A < x_1$

$y_B < y_{pA \rightarrow 1}$

$y_B > y_{pA \rightarrow 2}$

$x_B < x_2$

$y_A < y_{pB \rightarrow 2}$



Slika 3.24 Granični uvjet 8

Granični uvjet 8:

$x_1 < x_2$

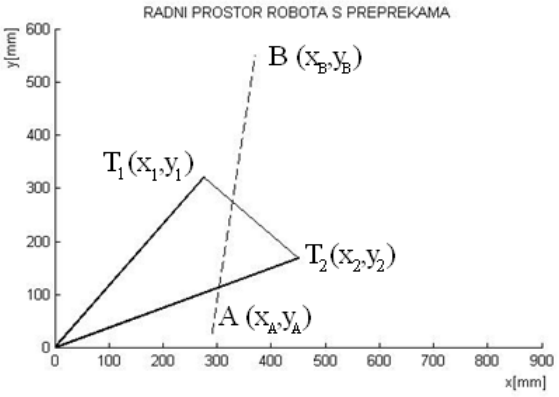
$x_A < x_1$

$y_B < y_{pA \rightarrow 1}$

$y_B > y_{pA \rightarrow 2}$

$x_B > x_2$

$y_A > y_{pB \rightarrow 2}$



Slika 3.25 Granični uvjet 9

Granični uvjet 9:

$x_1 < x_2$

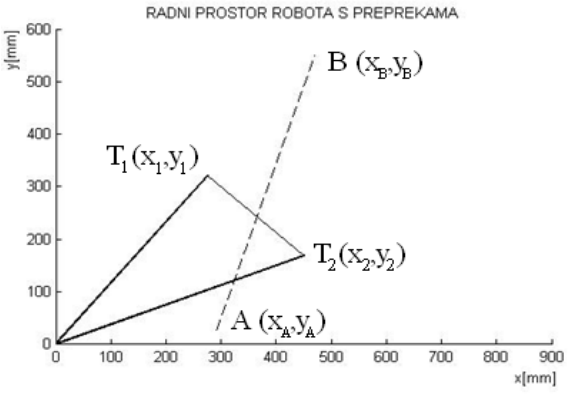
$x_A > x_1$

$y_B > y_{pA \rightarrow 1}$

$y_B > y_{pA \rightarrow 2}$

$x_B < x_2$

$y_A < y_{pB \rightarrow 2}$



Slika 3.26 Granični uvjet 10

Granični uvjet 10:

$x_1 < x_2$

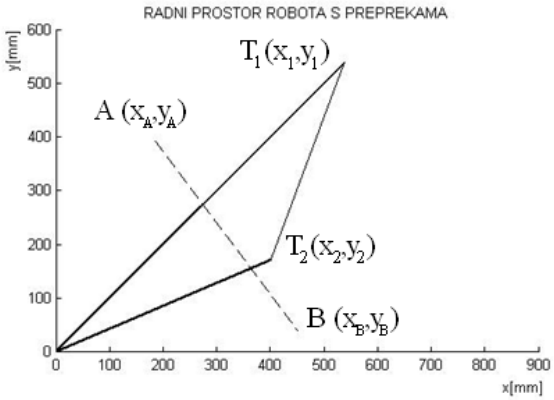
$x_A > x_1$

$y_B < y_{pA \rightarrow 1}$

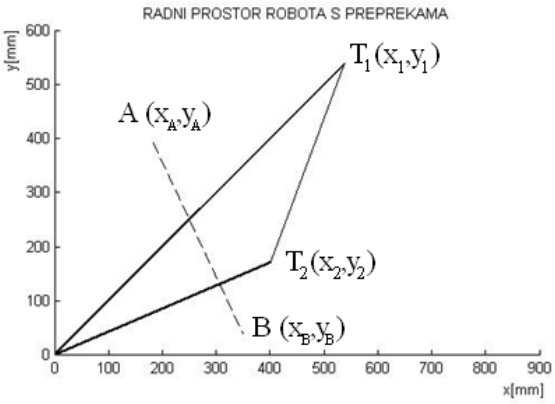
$y_B > y_{pA \rightarrow 2}$

$x_B > x_2$

$y_A > y_{pB \rightarrow 2}$



Slika 3.27 Granični uvjet 11



Slika 3.28 Granični uvjet 12

Granični uvjet 11:

$x_1 > x_2$

$x_B > x_2$

$y_A < y_{pB \to 2}$

$y_B < y_{pA \to 2}$

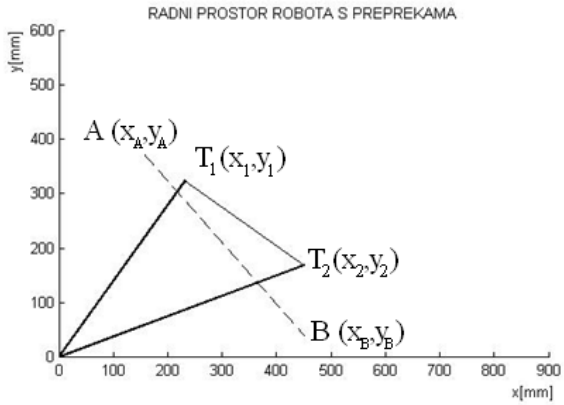
Granični uvjet 12:

$x_1 > x_2$

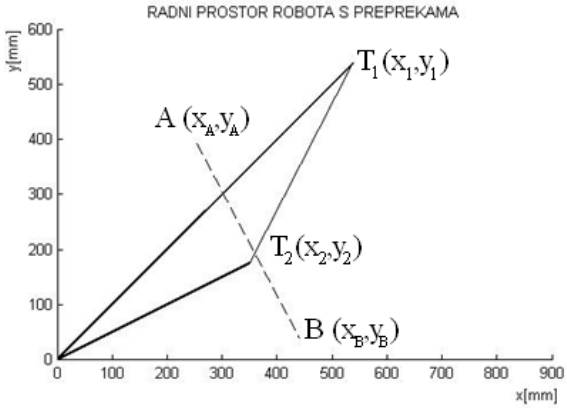
$x_B < x_2$

$y_A > y_{pB \to 2}$

$y_B < y_{pA \to 2}$



Slika 3.29 Granični uvjet 13



Slika 3.30 Granični uvjet 14

Granični uvjet 13:

$x_1 < x_2$

$x_A > x_1$

$y_B < y_{pA \to 2}$

Granični uvjet 14:

$x_1 > x_2$

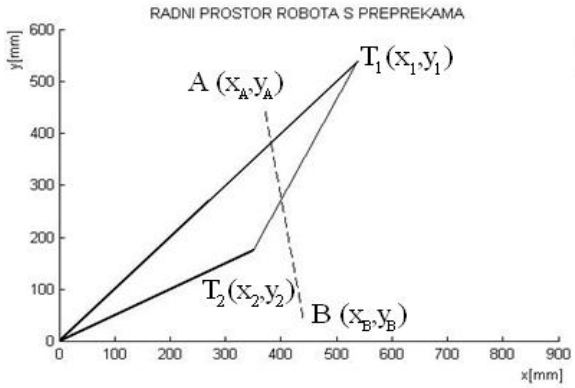
$x_B < x_1$

$y_A > y_{pB \to 2}$

$y_A > y_{pB \to 1}$

$x_A < x_2$

$y_B > y_{pA \to 2}$



Slika 3.31 Granični uvjet 15

Granični uvjet 15:

$x_1 > x_2$

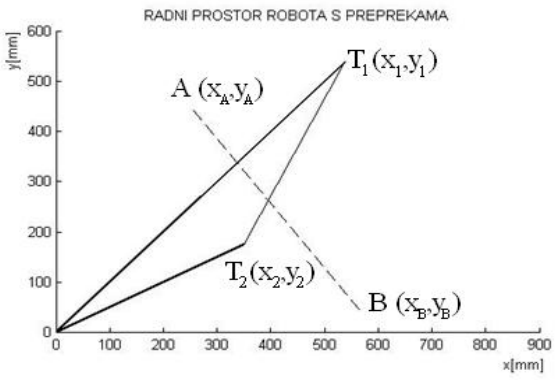
$x_B < x_1$

$y_A > y_{pB \rightarrow 2}$

$y_A > y_{pB \rightarrow 1}$

$x_A > x_2$

$y_B > y_{pA \rightarrow 2}$



Slika 3.32 Granični uvjet 16

Granični uvjet 16:

$x_1 > x_2$

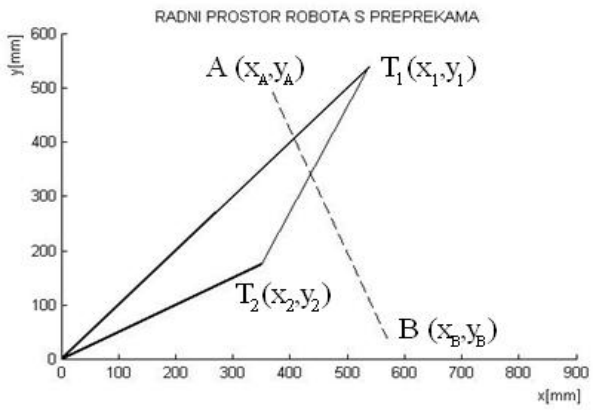
$x_B > x_1$

$y_A > y_{pB \rightarrow 2}$

$y_A < y_{pB \rightarrow 1}$

$x_A < x_2$

$y_B > y_{pA \rightarrow 2}$



Slika 3.33 Granični uvjet 17

Granični uvjet 17:

$x_1 > x_2$

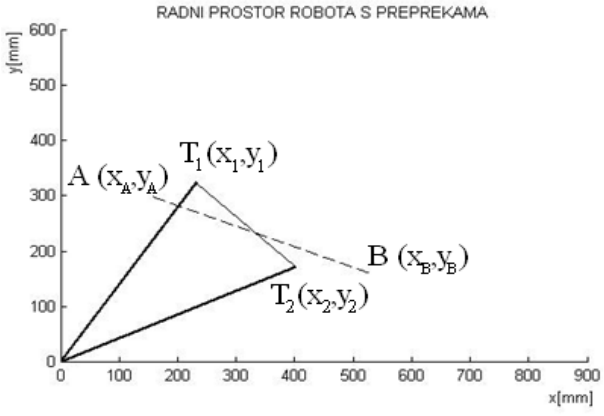
$x_B > x_1$

$y_A > y_{pB \rightarrow 2}$

$y_A < y_{pB \rightarrow 1}$

$x_A > x_2$

$y_B < y_{pA \rightarrow 2}$



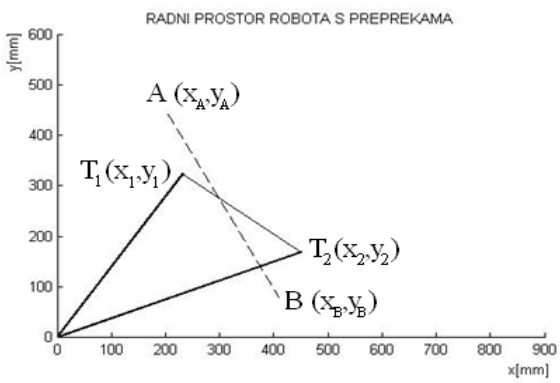
Slika 3.34 Granični uvjet 18

Granični uvjet 18:

$x_1 < x_2$

$y_A < y_{pB \rightarrow 1}$

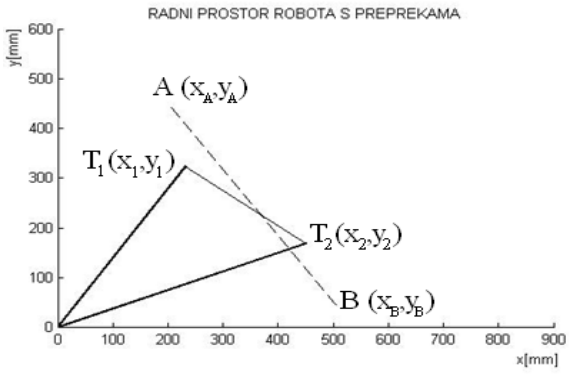
$y_B > y_{pA \rightarrow 2}$



Slika 3.35 Granični uvjet 19

Granični uvjet 19:

$x_1 < x_2$
$x_B < x_2$
$y_A > y_{pB \to 1}$
$y_A > y_{pB \to 2}$
$y_B > y_{pA \to 1}$
$y_B < y_{pA \to 2}$



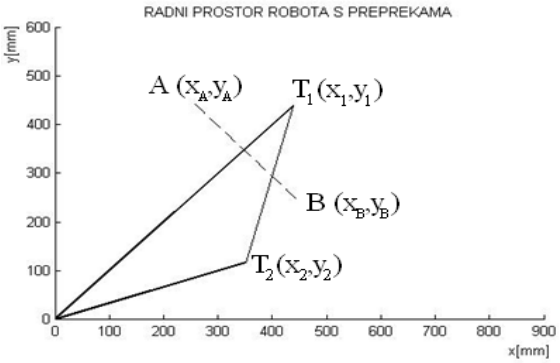
Slika 3.36 Granični uvjet 20

Granični uvjet 20:

$x_1 < x_2$
$x_B > x_2$
$y_A > y_{pB \to 1}$
$y_A < y_{pB \to 2}$
$y_B > y_{pA \to 1}$
$y_B < y_{pA \to 2}$

3.4.2 Prepreka sijeće početnu konfiguraciju

U ovom slučaju dio prepreke se nalazi iznad početne konfiguracije, a dio prepreke ispod početne i iznad krajnje konfiguracije.



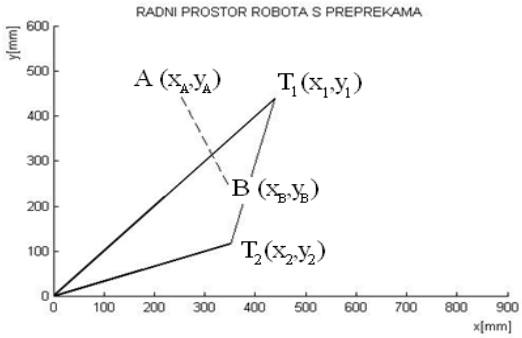
Slika 3.37 Granični uvjet 21

Granični uvjet 21:

$x_B > x_1$

$y_A < y_{pB \to 1}$

$y_B < y_{pA \to 1}$



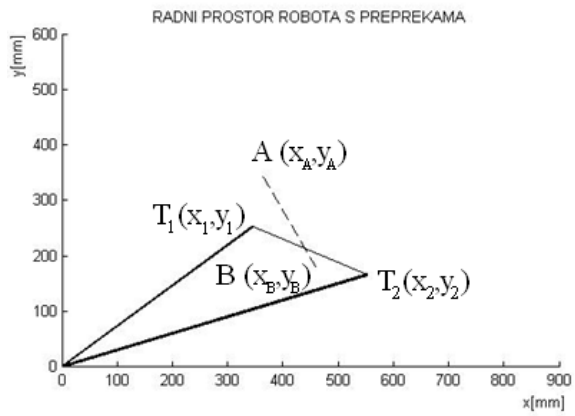
Slika 3.38 Granični uvjet 22

Granični uvjet 22:

$x_B < x_1$

$y_A > y_{pB \to 1}$

$y_B < y_{pA \to 1}$



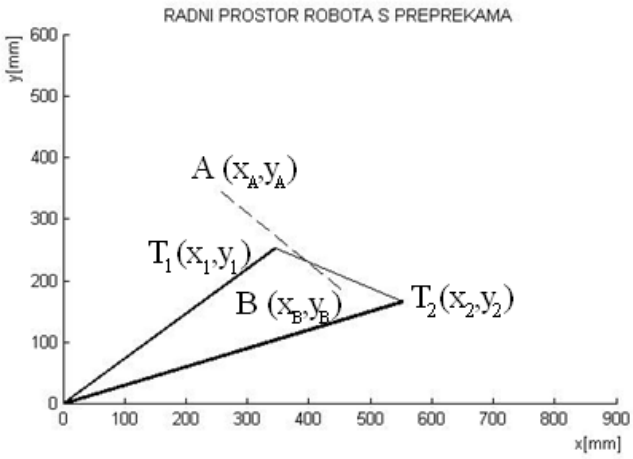
Slika 3.39 Granični uvjet 23

Granični uvjet 23:

$x_A > x_1$

$y_A > y_{pB \rightarrow 1}$

$y_B < y_{pA \rightarrow 1}$



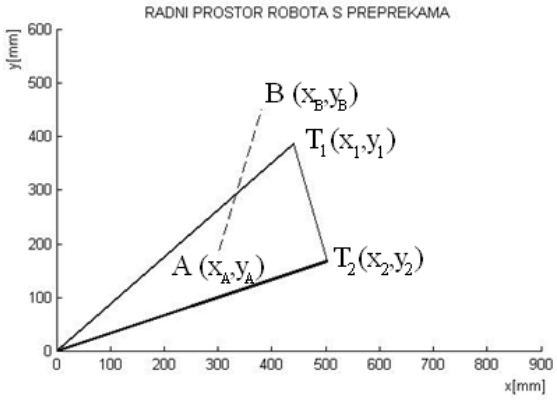
Slika 3.40 Granični uvjet 24

Granični uvjet 24:

$x_A < x_1$

$y_A > y_{pB \rightarrow 1}$

$y_B > y_{pA \rightarrow 1}$



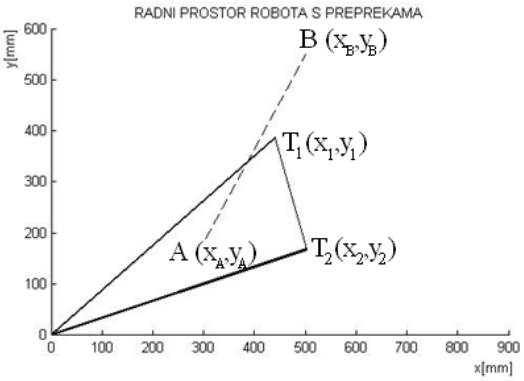
Slika 3.41 Granični uvjet 25

Granični uvjet 25:

$x_B < x_1$

$y_A < y_{pB \rightarrow 1}$

$y_B > y_{pA \rightarrow 1}$



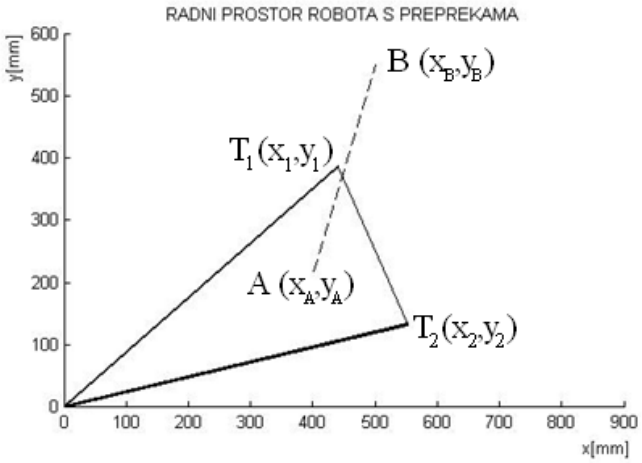
Slika 3.42 Granični uvjet 26

Granični uvjet 26:

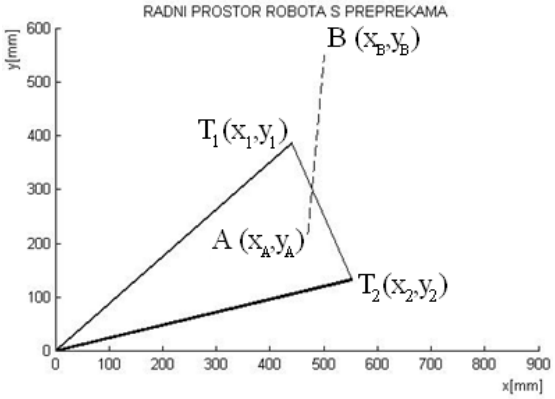
$x_B > x_1$

$y_A > y_{pB \rightarrow 1}$

$y_B > y_{pA \rightarrow 1}$



Slika 3.43 Granični uvjet 27



Slika 3.44 Granični uvjet 28

Granični uvjet 27:

$x_A < x_1$

$y_A < y_{pB \rightarrow 1}$

$y_B < y_{pA \rightarrow 1}$

$x_1 < x_2$

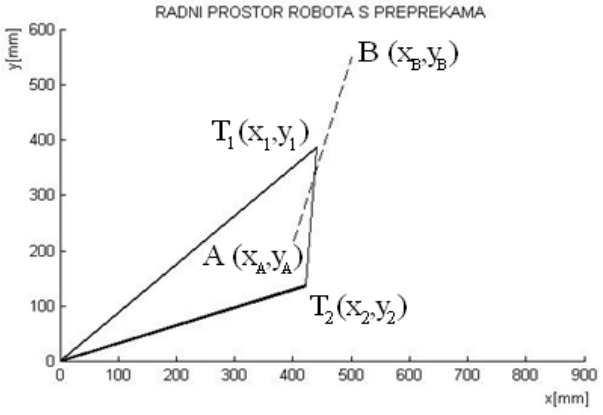
$y_A < y_{p3}$

Granični uvjet 28:

$x_A > x_1$

$y_A < y_{pB \rightarrow 1}$

$y_B > y_{pA \rightarrow 1}$



Slika 3.45 Granični uvjet 29

Granični uvjet 29:

$x_A > x_1$

$y_A < y_{pB \rightarrow 1}$

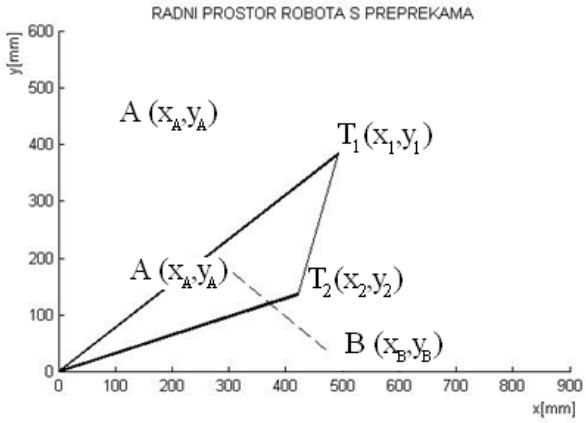
$y_B < y_{pA \rightarrow 1}$

$x_1 < x_2$

$y_A > y_{p3}$

3.4.3 Prepreka sijeće krajnju konfiguraciju

U ovom slučaju dio prepreke se nalazi ispod krajnje konfiguracije, a dio prepreke ispod početne i iznad krajnje konfiguracije.



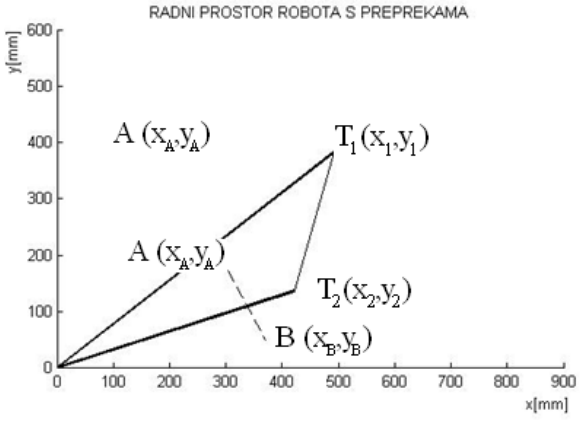
Slika 3.46 Granični uvjet 30

Granični uvjet 30:

$x_B > x_2$

$y_A < y_{pB \to 2}$

$y_B < y_{pA \to 2}$



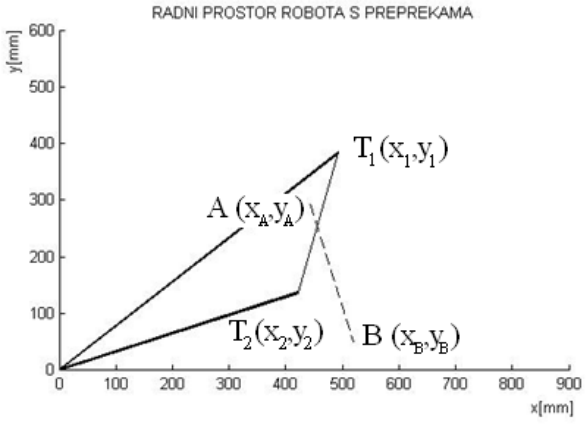
Slika 3.47 Granični uvjet 31

Granični uvjet 31:

$x_B < x_2$

$y_A > y_{pB \to 2}$

$y_B < y_{pA \to 2}$



Slika 3.48 Granični uvjet 32

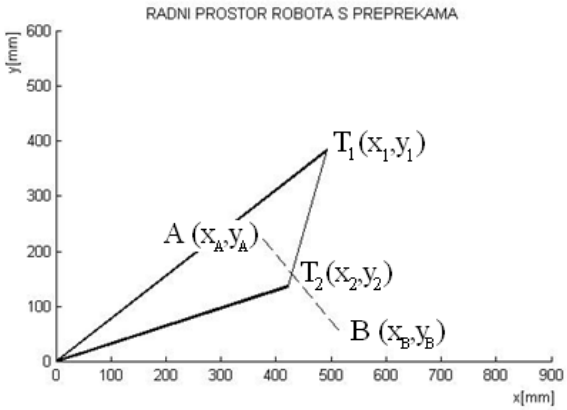
Granični uvjet 32:

$x_A > x_2$

$y_A > y_{pB \to 2}$

$y_B < y_{pA \to 2}$

$x_1 > x_2$



Slika 3.49 Granični uvjet 33

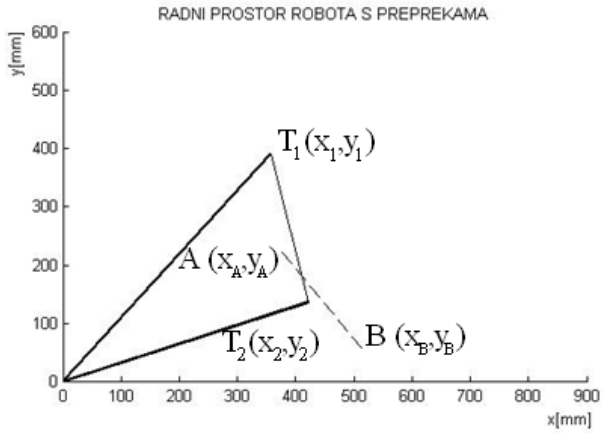
Granični uvjet 33:

$x_A < x_2$

$y_A > y_{pB \to 2}$

$y_B > y_{pA \to 2}$

$x_1 > x_2$



Slika 3.50 Granični uvjet 34

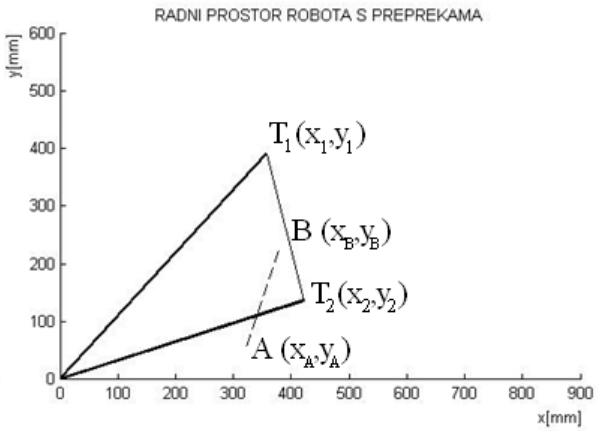
Granični uvjet 34:

$x_A < x_2$

$y_A > y_{pB \rightarrow 2}$

$y_B > y_{pA \rightarrow 2}$

$x_1 < x_2$



Slika 3.51 Granični uvjet 35

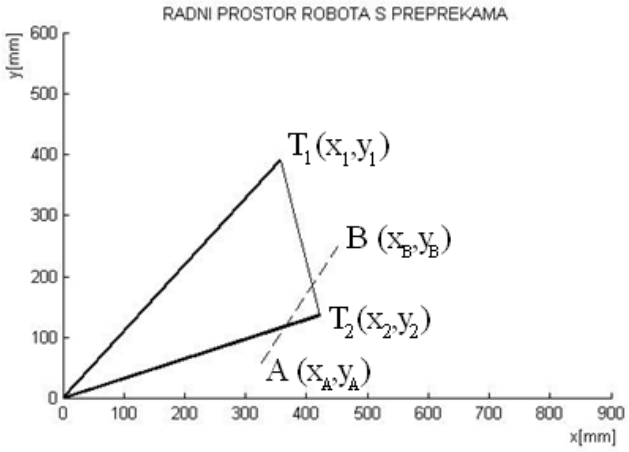
Granični uvjet 35:

$x_B < x_2$

$y_A < y_{pB \rightarrow 2}$

$y_B > y_{pA \rightarrow 2}$

$x_1 < x_2$



Slika 3.52 Granični uvjet 36

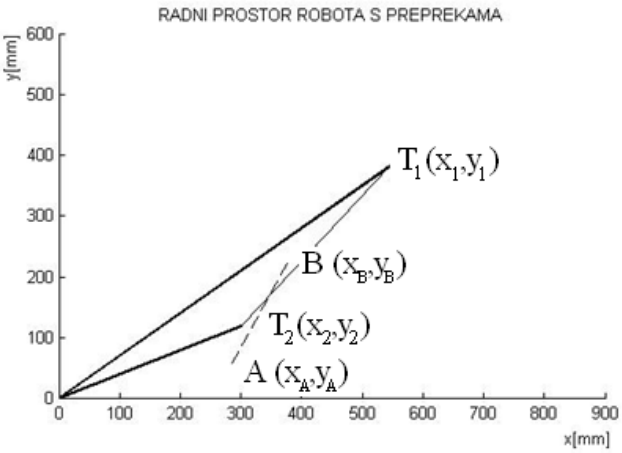
Granični uvjet 36:

$x_B > x_2$

$y_A > y_{pB \rightarrow 2}$

$y_B > y_{pA \rightarrow 2}$

$x_1 < x_2$



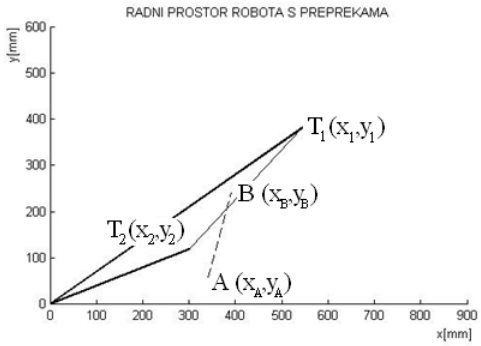
Slika 3.53 Granični uvjet 37

Granični uvjet 37:

$x_A < x_2$

$y_A < y_{pB \rightarrow 2}$

$y_B < y_{pA \rightarrow 2}$



Slika 3.54 Granični uvjet 38

Granični uvjet 38:

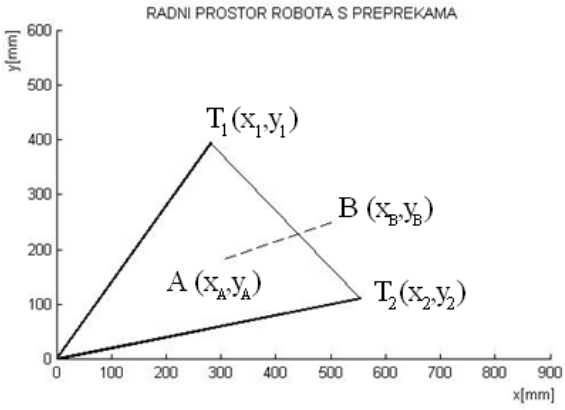
$x_B > x_2$

$y_A < y_{pB \rightarrow 2}$

$y_B > y_{pA \rightarrow 2}$

3.4.4 Prepreka se nalazi između početne i krajnje konfiguracije

U ovom slučaju prepreka se nalazi ispod početne konfiguracije i iznad krajnje konfiguracije.



Slika 3.55 Granični uvjet 39

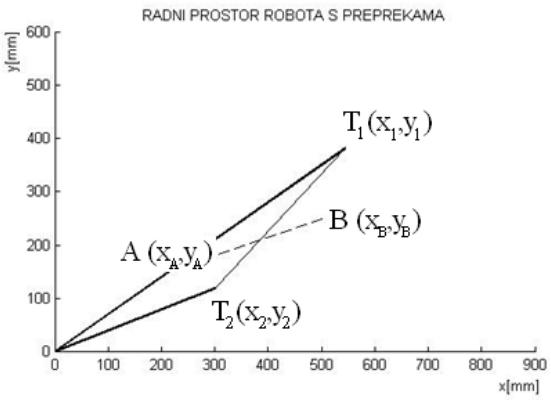
Granični uvjet 39:

$y_A < y_{p1}$

$y_A > y_{p2}$

$y_B < y_{p1}$

$y_B > y_{p2}$



Slika 3.56 Granični uvjet 40

Granični uvjet 40:

$y_A < y_{p1}$

$y_A > y_{p2}$

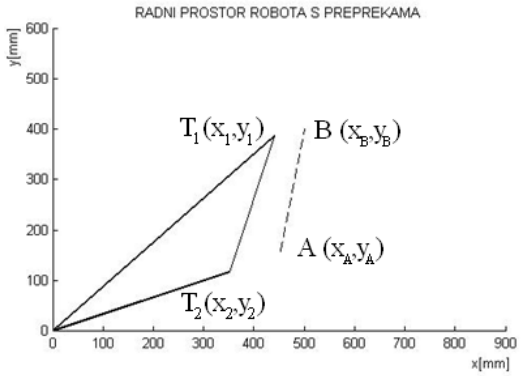
$y_B < y_{p1}$

$y_B > y_{p2}$

$x_1 < x_2$	$x_1 > x_2$
$y_A < y_{p3}$	$y_A > y_{p3}$
$y_B > y_{p3}$	$y_B < y_{p3}$

3.4.5 Prepreka se nalazi izvan početne i krajnje konfiguracije

U ovom slučaju prepreka se nalazi iznad početne konfiguracije ili ispod krajnje konfiguracije.



Slika 3.57 Granični uvjet 41

3.5 Definiranje točaka putanje

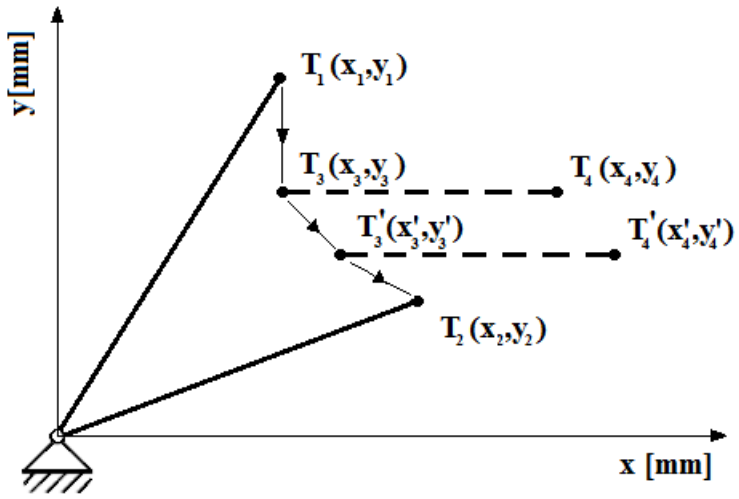
Ispunjavanjem pojedinih graničnih uvjeta dobije se precizno rješenje gdje se koja prepreka nalazi u radnom području.

Ukoliko neka prepreka siječe početnu ili krajnju konfiguraciju i nemogućnosti zaobilaženja iza prepreke zbog jednog rotacijskog zgloba, zadatak pronalaska krajnje putanje nije moguć i robot ne može izvršiti gibanje.

Ako se sve prepreke (uključujući i drugog robota) nalaze izvan radnog područja robota tada se gibanje robota može izvršiti najkraćim putem po pravcu od njegove početne do krajnje konfiguracije.

Ako unutar područja postoje prepreke tada je potreban kod za zaobilaženje prepreka.

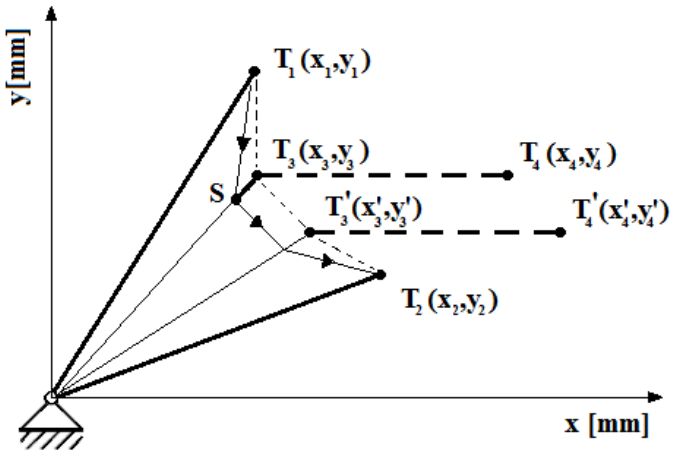
Detaljnom analizom došlo se do zaključka da krajnje točke prepreke predstavljaju točke po kojima ako se kreće robot, nikada neće doći do sudara s preprekom ili drugim robotom (slika 3.58).



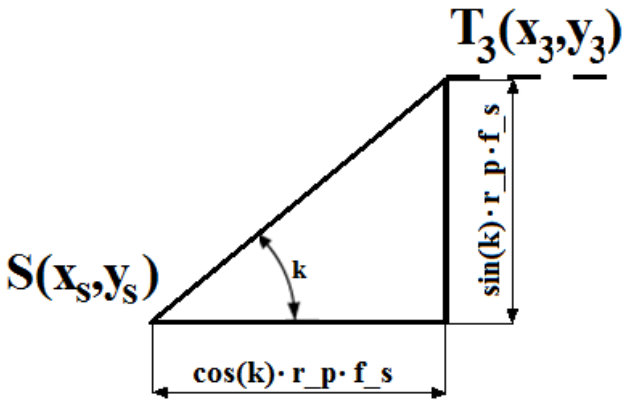
Slika 3.58 Definiranje točaka putanje

Na slici 3.58 tankom linijom je prikazana putanja robota od točke T_1 preko T_3, T'_3 pa do T_2. U ovom slučaju, robot je prikazan kao tanka linija, a u stvarnosti on ima svoje dimenzije - širinu segmenata. U tom slučaju, ako bi uzeli u obzir širinu segmenata, došlo bi do male kolizije između robota i prepreke kod samog mimoilaženja. Stoga je potrebno odmaknuti stvarnu putanju robota od krajnjih točaka prepreke.

Takozvano „skraćivanje“ putanje robota izveli smo tako da smo od stvarne točke putanje oduzeli radijus prihvatnice uvećan za faktor sigurnosti zaobilaženja kojeg možemo mijenjati ovisno kakvu sigurnost želimo.



Slika 3.59 Skraćena putanja robota



Slika 3.60 Uvećani prikaz skraćivanja putanje

Jednadžbe za skraćivanje putanje robota:

$$k = \frac{y_3 - 0}{x_3 - 0} \tag{19}$$

$$x_s = x_{T_3} - \cos(k) \cdot r_p \cdot f_s \tag{20}$$

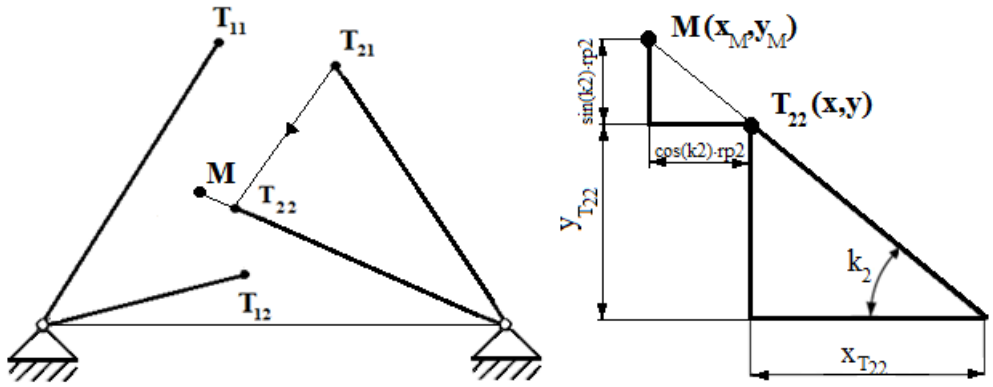
$$y_s = y_{T_3} - \sin(k) \cdot r_p \cdot f_s \tag{21}$$

gdje je:

r_p - radijus prihvatnice

f_s - faktor sigurnosti

Dodatni problem nam predstavlja zaobilaženje robota kada se jedan robot nalazi u radnom području drugog robota. Tada se javljaju i širina segmenata prvog i drugog robota. Analizom tog problema uspostavilo se da je potrebno još jedno dodatno odmicanje u odnosu na obične prepreke.

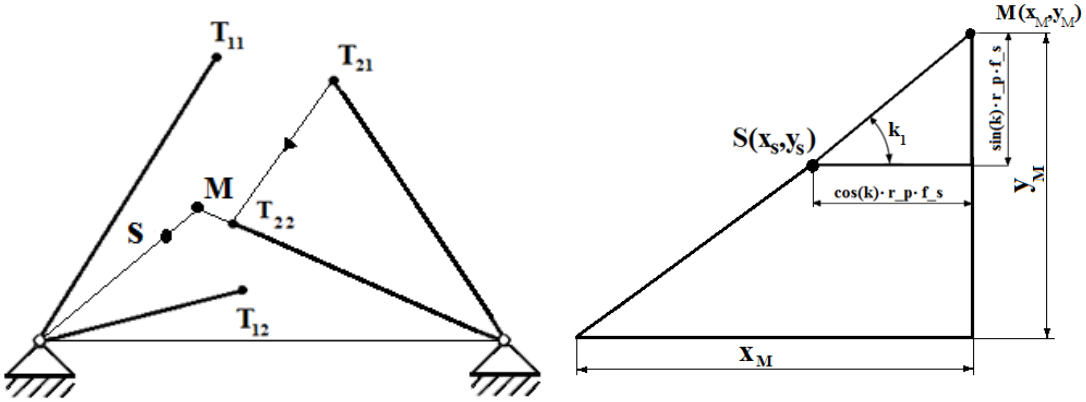


Slika 3.61 Skraćivanje putanje za radijus drugog robota

$$k_2 = \frac{y_{T_{22}} - 0}{udaljenost\ između\ robota - x_{T_{22}}} \tag{22}$$

$$x_M = x_{T_{22}} - \cos(k_2) \cdot r_{p2} \cdot f_s \tag{23}$$

$$y_M = y_{T_{22}} + \sin(k_2) \cdot r_{p2} \cdot f_s \tag{24}$$

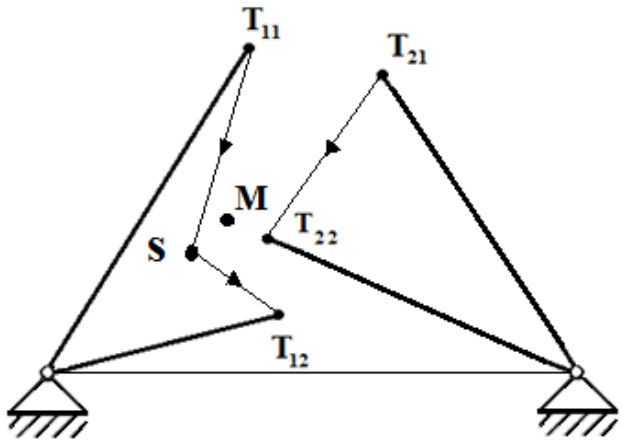


Slika 3.62 Skraćivanje putanje za radijus prvog robota

$$k_1 = \frac{y_M - 0}{x_M - 0} \tag{25}$$

$$x_S = x_M - \cos(k_1) \cdot r_{p1} \cdot f_s \tag{26}$$

$$y_S = y_M + \sin(k_1) \cdot r_{p1} \cdot f_s \tag{27}$$



Slika 3.63 Konačan izgled skraćene putanje

Na slici je prikazano pet karakterističnih točaka: *S, M, $T_{11,}$ $T_{12,}$ T_{21}* i T_{22}. Desni robot s krajnjom konfiguracijom nalazi se u radnom području lijevog robota. Time je krajnja konfiguracija desnog robota prepreka za lijevi robot koju on mora zaobići. Ako bi se robot odmaknuo u točku *M* tada bi se gibao po putanji T_{11} - *M* - T_{12} i na pravcu *M*-T_{12} zapeo bi za desni robot. Da bi se robot odmaknuo dovoljno od desnog robota, mora se odmaknuti za umnožak radijusa hvataljke drugog robota i faktora sigurnosti i za umnožak svoga radijusa hvataljke i faktora sigurnosti. Time smo dobili putanju kroz točku *S* koja je dovoljno odmaknuta od desnog robota (slika 3.63).

3.6 Slanje poruke računalo-robot

U poglavlju 3.1.2 opisano je kako upravljačka jedinica robota šalje poruku računalu. Isto tako nakon što se u Matlabu pronađu rješenja, tj. točke kroz koje se treba gibati robot, šalju se preko TCP-IP protokola nazad upravljačkoj jedinici robota.

Poruke koje se šalju mogu biti različitog oblika. Oblik poruke ovisi o tome da li se sijeku međusobno početne ili krajnje konfiguracije robota, da li postoje prepreke u radnom području robota, da li se jedan robot nalazi u radnom području drugog robota.

Poruka ima oblik:

$$[prepreke\_sijeku \;\; broj\_pozicija \;\; x_n \;\; y_n] \tag{28}$$

gdje je:

prepreke_sijeku - broj prepreka koje sijeku putanju

broj_pozicija - broj točaka kroz koje prolazi robot

x_n - x koordinata točke prolaska

y_n - y koordinata točke prolaska

Varijabla *prepreke_sijeku* može poprimiti četiri vrijednosti:

0 - prepreke ne presjecaju niti početnu niti krajnju konfiguraciju

1 - prepreke presjecaju početnu ili krajnju konfiguraciju i nije moguće izvršiti zadatak

2 - na krajnjoj konfiguraciji tog robota nalazi se drugi robot

3 - taj robot se nalazi na krajnjoj konfiguraciji drugog robota

Oblik poruke: [*0 0 0*] - znači da ne postoji nikakvog presjecanja prepreka te da se robot giba najkraćom putanjom od početne do krajnje konfiguracije.

Oblik poruke: [*1 0 0*] - znači da postoji presjecanja prepreka i početne ili krajnje konfiguracije, a iz toga proizlazi da nije moguće izvršiti zadatak.

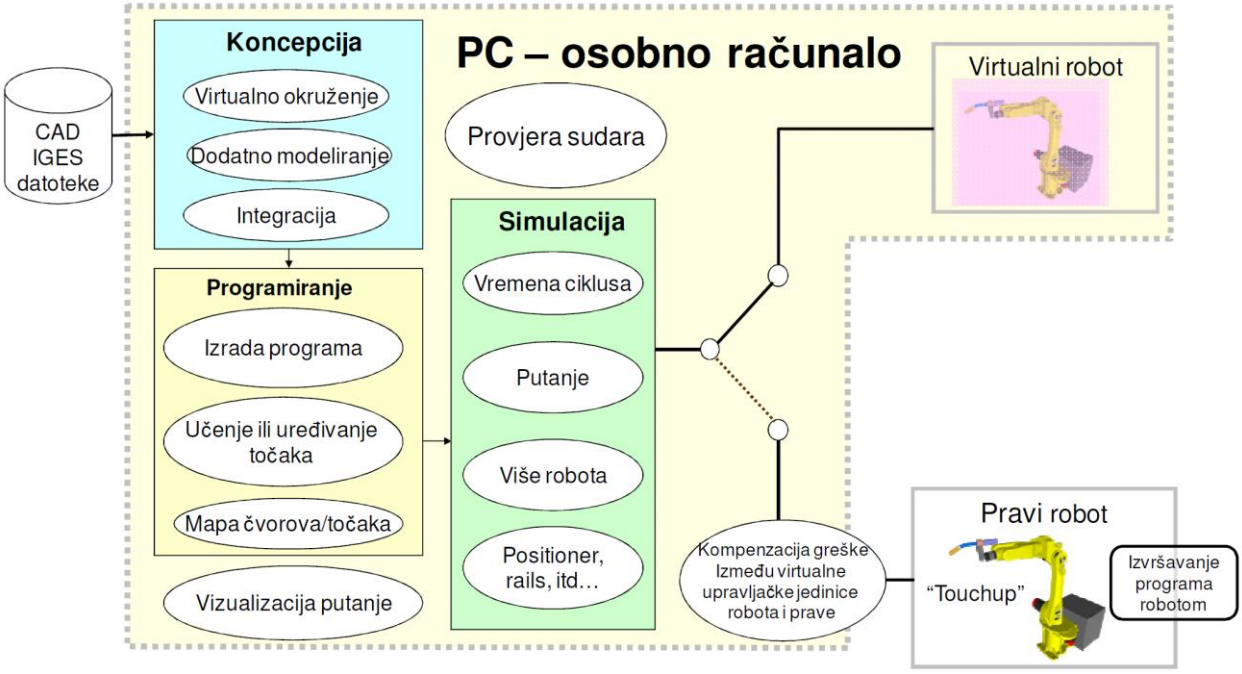
Oblik poruke: [*2 0 56.25 78.30*] - znači da se na krajnjoj konfiguraciji tog robota nalazi drugi robot i da se smije kretati do točke T_2(*56.25,78.30*). Kada robot dođe u tu točku, mora zaustaviti kretanje drugog robota i provjeriti da li se je drugi robot odmaknuo s njegove krajnje konfiguracije.

Oblik poruke: [*3 0 0*] - znači da se taj robot nalazi na kajnjoj konfiguraciji drugog robota i da očekuje da će ga drugi robot zaustaviti kad dođe do određene točke da provjeri da li se je odmaknuo s njegove krajnje konfiguracije.

Oblik poruke: [*0 1 452 386*] - znači da ne postoji nikakvog presjecanja prepreka te da postoji jedna pozicija u koju robot mora ići od početne do krajnje konfiguracije, a to je točka *S* (*452, 386*).

3.7 Simulacija u fizičkom simulacijskom okružju Roboguide

Simulacijsko okružje Roboguide fizičko je simulacijsko okružje koje pruža široke mogućnosti provedbe simulacija u realnim uvjetima. Osnovna namjena je brza prethodna verifikacija i validacija robotskog sustava uz minimalne utroške vremena i financijskih sredstava. Pri modeliranju i kasnijoj verifikaciji u obzir su uzeta kašnjenja signala, komunikacije, izvedbe odgovarajućih kretanja. U naravi, razlika između programiranja na samom robotu ili u ovom sučelju minimalna je, osim što je jednostavnija komunikacija s ulazno izlaznim jedinicama robota u odnosu na stvarne robote gdje se sve odvija putem privjeska za učenje. Stoga samo programiranje robota za bilo koju kompleksniju namjenu nije preporučljivo izvoditi putem privjeska za učenje već putem Roboguide sučelja, te naknadno pohraniti program na upravljačko računalo samoga robota.



Slika 3.64 Princip rada fanucovog *Roboguide* simulacijskog okružja

U Roboguide sučelju moguće je vrlo detaljno definirati ne samo robota s kojim se radi nego i cijelu radnu okolinu u kojoj se robot nalazi. Pri tome su fizičke prepreke (poput ograda, objekata ili drugih robota) u ovome virtualnom svijetu prepoznate kao prepreke i predstavljaju područje nedostupno za robota. Postupak rada je sljedeći, u prvom se koraku odabire tip robota koji se ugrađuje u novu radnu okolinu. Definiraju se softverske karakteristike robota, i sva pripadajuća oprema i odgovarajući softverski moduli (kamera, senzor momenta i sile, ostali senzori). Nakon toga definira se vrsta

komunikacije robota s perifernom opremom. Robot se potom dodaje u generičku okolinu gdje mu je potrebno definirati mjesto i orijentaciju u prostoru. Ukoliko je potrebno u okolinu ugraditi dodatne robote postupak se ponavlja.

4. ZAKLJUČAK

Ovaj rad se sastojao od dva dijela. U prvome dijelu diplomskog rada bilo je potrebno izraditi pojednostavljeni simulacijski model te algoritam za rješavanje problema putanje istovremenog kretanja dvaju robota koji dijele radni prostor. Kritičan parametar algoritma je vrijeme procesiranja odnosno pronalaženja prihvatljivog rješenja, te ga je bilo potrebno minimizirati. U drugome dijelu rada rješenja pronađena u simulacijskom okružju bilo je potrebno implementirati na dva stvarna robota Fanuc LR Mate 200iC.

Pojednostavljenjem modela robota na konfiguraciju s jednim rotacijskim zglobom i segmentom promijenjive duljine, uvelike je pridonjelo minimizaciji vremena potrebnog za rješavanje problema putanje robota. Brzina odziva je manja od 0.5 sekundi, što je dovoljno da se robot kontinuirano giba od njegove početne do krajnje konfiguracije.
No minimiziranjem vremena odziva pojavio se problem što robot zbog pojednostavljenog modela s jednim rotacijskim zglobom nije u mogućnosti doći iza neke prepreke ili drugog robota, pa se javljaju ograničenja u kretanju robota.

Uz pojednostavljeni model robota, napravljen je algoritam koji se sastoji od tridesetak manjih podprograma koji se pozivaju i izvršavaju ovisno o vrsti zadatka kojeg robot izvršava. Time smo još više ubrzali vrijeme procesiranja odnosno pronalaženja prihvatljivog rješenja. Po broju podprograma vidljivo je kao zadatk planiranja putanje nije nimalo jednostavan već vrlo komplesan koji ima nebrojena rješenja do kojih nije jednostavno doći.

Implementacijom simulacijskog okruženja na stvarne robote došli smo do niza problema koji su se morali riješiti da bi izveli zadatak planiranja kretenja dvaju robota. Njihova kofiguracija zadala je mnoštvo problema kod definiranja međusobnih odnosa dvaju robota, kao i način zaobilaženja.
Na robotima se nalaze kamre, hvataljka, crijeva za zrak, kablovi za kamere i senzori. Javile su se mnoge novosti koje u simulacijskom okruženju nismo uzeli u obzir. Stoga je bilo potrebno dodavati nove parametre kako bi uzeli u obzir sve „nesavršenosti“ konfiguracije koje se nisu mogle uvidjeti u simulacijskom okružju.

Uz navdene prednosti i nedostatke možemo reći da postoji još puno mogućnosti i prostora za razradu problema planiranja kretanja dvaju robota koji djele isti radni prostor.

5. LITERATURA

[1] K. A. De Jong, "Evolving Intelligent Agents: A 50 Year Quest," *IEEE Computational Intelligence Magazine*, vol. 3, no. 1, pp. 12-17, February 2008.

[2] J.-C. Latombe, *Robot Motion Planning*. Boston, USA: Kluwer Academic Publishers, 1991.

[3] (2011, listopad) IFR International Federation of Robotics. [Online]. http://www.ifr.org/

[4] P. Ćurković, ” Evolutivni algoritam za upravljanje višeagentskim robotskim sustavom“, Zagreb, Studeni 2010.

[5] R30iA Manual CRO

[6] B-82724EN-1_01_R-30iA_LRMate_HandlingTool

[7] R-30iA_KAREL_Reference_Manual_[Ver.7.30][MARRCRLRF04071E_REV.B]

[8] R-J3iC_Internet_Options_Setup&Operations_Manual_[7.20][MAROCINOP08051E_REV.B]

Literatura [5] osnovni je vodić kroz upoznavanje i osnovni rad sa Fanuc robotima. Priručnik [6] kompletna je dokumentacija o upravljačkoj jedinici R-30iA Mate te o programiranju uz pomoć upravljačke konzole.

Priručnik [7] odnosi se na rad u programskom jeziku Karel. U njemu je prikazana kompletna dokumentacija programskog jezika Karel, svih programskih struktura kao i mnogo primjera programa. Za izradu Karel programa koji koriste korisničko komuniciranje pomoću utora (*user socket messaging*) objašnjenja te primjeri se nalaze u [8].

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

PLANIRANJE ISTOVREMENOG KRETANJA DVAJU ROBOTA S PREKLAPAJUĆIM RADNIM PROSTOROM

PRILOG

KRUNOSLAV DUBIĆ

Zagreb, 2011.

PRILOG

1.1 Karel

```
PROGRAM DUB_R1_01
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE

VAR
    STATUS,entry,port,portC: INTEGER
    file_var_s,file_var_c: FILE
    nb,i,sjec,toc:INTEGER
    r_f:BOOLEAN
    s,niz,robot,sje,k: STRING[25]
    s1:STRING[120]
    rand,x1,y1,z1,z2,w1,r,p1,r1,x_po,y_po,x_ci,y_ci:REAL
    P00,P01,P02,P03,P04,P05,P06,P07: XYZWPR
    config_var1:CONFIG
    indx IN DRAM:INTEGER


 --*** vanjske rutine za uspostavljanje TCP/ip konekcije
  ROUTINE OPEN_FILE_ (FILE_ : FILE; TAG_ : STRING) FROM LIB_FILE
  ROUTINE CLOSE_FILE_(FILE_ : FILE; TAG_ : STRING) FROM LIB_FILE

BEGIN

    $GROUP[1].$UFRAME = $MNUFRAME[1,4];
    $GROUP[1].$UTOOL =  $MNUTOOL[1,5]
    $GROUP[1].$MOTYPE=LINEAR;
    $GROUP[1].$SPEED=400
    $GROUP[1].$TERMTYPE=FINE;

       --*********************************************************
       --o kojem je robotu riječ
       GET_VAR(entry,'*SYSTEM*', '$HOSTNAME',s,STATUS)
       IF s='ROBOT1' THEN;robot='R1_';port=10257;ENDIF;
       IF s='ROBOT1' THEN;robot='R1_';portC=10258;ENDIF;
       IF s='ROBOT2' THEN;robot='R2_';port=10267;ENDIF;
       IF s='ROBOT2' THEN;robot='R2_';portC=10268;ENDIF;
       --*********************************************************

   --zaustavi sve tagove prije promjene!!!
    -- postavljanje vrijenosti za svaki tak preko kojega će se vršiti komunikacija sa drugim
uređajem
    -- pomoću TCP/IP protokola     'C7:'    i 'S7:'
    -- moguće je definirati do 8 tagova za klijente 'C1:'-'C8:'   i 8 tagova za serverske
konekcije  'S1:'-'S8:'

    -- setiranje atributa za tak S7:
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$OPER',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STATE',0,STATUS) ; DELAY 20
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$COMMENT','SOUND',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$PROTOCOL','SM',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$REPERRS','FALSE',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$TIMEOUT',9999,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$PWRD_TIMEOUT',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$SERVER_PORT',port,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STRT_PATH','192.168.123.39',STATUS)--ip
adresa uređaja sa kojim se komunicira
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STRT_REMOTE','192.168.123.39',STATUS) ;--
-ll-
```

```
      -- setiranje atributa za tak C7:
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$OPER',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STATE',0,STATUS) ; DELAY 20
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$COMMENT','SOUND',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$PROTOCOL','SM',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$REPERRS','FALSE',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$TIMEOUT',9999,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$PWRD_TIMEOUT',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$SERVER_PORT',portC,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$PATH','192.168.123.39',STATUS) ;--ip adresa
uređaja sa kojim se komunicira
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STRT_PATH','192.168.123.39',STATUS) ;  --
-ll-
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$remote','192.168.123.39',STATUS) ;
        --       -ll-
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STRT_REMOTE','192.168.123.39',STATUS) ;--
-ll-

    --inicijaliziraj ponovo tagove 7 kada si napravio promjenu

    DELAY 10 ;

      SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$OPER',3,STATUS);
      SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STATE',3,STATUS) ;


    DELAY 10 ;

      SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$OPER',3,STATUS);
      SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STATE',3,STATUS) ;

 DELAY 10;

    DOUT[121]=OFF;
    DOUT[122]=OFF;

    CLOSE_FILE_(file_var_C,'C7:') ;  --zatvaranje konekcije ako je ostala otvorena od prije

DELAY 10

    OPEN_FILE_(file_var_C,'C7:');  --otvaranje konekcije

----------------------------------------
--stisni jedan da se pokrene program
   WRITE (CR,'POSTAVITE ROBOT U POCETNI POLOZAJ!',CR,CR)
   WRITE ('Pritisni 1 za POKRETANJE programa',CR)
   WRITE ('Pritisni 2 za IZLAZ iz programa',CR)

WHILE TRUE DO
  DELAY 10

  IF TPIN[49]=ON THEN;

   WRITE (CR,'IZVRŠAVANJE programa..',CR)

--inicijalizacija varijabli
    s1=''
    x1=0
    y1=0
    z1=20
    z2=5
    w1=0
    p1=0
    r1=0
    DOUT[121]=OFF;
```

```
    DOUT[122]=OFF;

-----------------------------------------
 RDO[1]=ON  -- otvori hvataljku
-----------------------------------------

-- startna pozicija robota (hvataljka otvoena)
   P01=CURPOS(0,0)     --spremanje trenutne pozicije robota u varijablu p1

-- startna pozicija robota (hvataljka otvoena)
  CNV_STR_CONF('nut000', config_var1, STATUS)
     P02= POS(P01.x,P01.y,z1,w1,p1,r1,config_var1)
-----------------------------------------

-- startna pozicija robota (hvataljka zatvorena)
  CNV_STR_CONF('nut000', config_var1, STATUS)
     P03= POS(P02.x,P02.y,z2,w1,p1,r1,config_var1)
-----------------------------------------

-- ciljna pozicija robota (hvataljka otvoena)
  CNV_STR_CONF('nut000', config_var1, STATUS)
  --   x_ci=500;y_ci=400;
   x_ci=480;y_ci=480;
     P05= POS(x_ci,y_ci,z1,w1,p1,r1, config_var1)
-----------------------------------------

-- ciljna pozicija robota (hvataljka zatvorena)
  CNV_STR_CONF('nut000', config_var1, STATUS)

     P06= POS(x_ci,y_ci,z2,w1,p1,r1, config_var1)
-----------------------------------------

-- ciljna pozicija robota (hvataljka zatvorena)
  CNV_STR_CONF('nut000', config_var1, STATUS)

     P07= POS(x_ci,y_ci,z1,w1,p1,r1, config_var1)

-----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-----------------------------------------
      DOUT[121]=ON;
      wait for DIN[121]=ON;


 DELAY 10
  OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
       WRITE file_var_S (0) --broj pravaca koji sjeku putanju
       WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

       CNV_REAL_STR(p02.x,2,2,s1) --točka starta
       WRITE file_var_S (s1)

       CNV_REAL_STR(p02.y,2,2,s1)
       WRITE file_var_S (s1)

       CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
       WRITE file_var_S (s1)

       CNV_REAL_STR(p05.y,2,2,s1)
       WRITE file_var_S (s1)

       WRITE file_var_S (' m')          --poruka namijenjena matlabu
-----------------------------------------
```

```
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------

provjera1::


 DELAY 10
 WHILE TRUE DO

       BYTES_AHEAD(file_var_C,nb,STATUS)   --čitanje koliko ima bajtova u međuspremniku

       DELAY 10 -- čekanje

       IF nb>3 THEN  -- ako ima više od 2 bajta u međuspremniku čitaj što je u međuspremniku
u varijablu niz

--čitanje da li koji pravac sijeće putanju
        READ file_var_C (sjec)
        WRITE('Broj tocaka koje sijeku:',sjec::7::0,CR)
----------------------------------------

--čitanje koliko ima pozicija
        READ file_var_C (toc)
        WRITE('Broj tocaka(prolazi):',toc::7::0,CR)

--ako niti jedan pravac ne sjece putanju
DELAY 10

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

     IF sjec=0 THEN

--       wait for DIN[121]=ON;
--       DOUT[121]=ON;

--ako postoji neka pozicija, kreni u tu poziciju
     DELAY 10
       IF toc>0 THEN
           P04= POS(P02.x,P02.y,z1,w1,p1,r1, config_var1)

          FOR i=1 TO toc DO
            READ file_var_C (x1)
            WRITE('x:',x1::7::2,CR)
              P04.x=x1
            READ file_var_C (y1)
            WRITE('y:',y1::7::2,CR)
              P04.Y=y1

            MOVE TO P04

          ENDFOR
       ENDIF
-- odlaganje predmeta

    MOVE TO P05
    MOVE TO P06
       DELAY 250
----------------------------------------
  RDO[1]=ON  -- otvori hvataljku
----------------------------------------
       DELAY 250
    MOVE TO P05
    WRITE('PROGRAM IZVRSEN!',CR)
```

```
    DOUT[121]=OFF;
    DOUT[122]=OFF;
  ENDIF

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

    IF sjec=2 THEN


--ako postoji neka pozicija, kreni u tu poziciju
     DELAY 10
       IF toc>0 THEN
           P04= POS(P02.x,P02.y,z1,w1,p1,r1, config_var1)

      FOR i=1 TO toc DO
        READ file_var_C (x1)
        WRITE('x:',x1::7::2,CR)
          P04.x=x1
        READ file_var_C (y1)
        WRITE('y:',y1::7::2,CR)
          P04.Y=y1

        MOVE TO P04

      ENDFOR
  ENDIF
-- odlaganje predmeta
        READ file_var_C (x1)
        WRITE('x:',x1::7::2,CR)
          P07.x=x1
        READ file_var_C (y1)
        WRITE('y:',y1::7::2,CR)
          P07.y=y1
    MOVE TO P07

    DOUT[122]=ON;
    wait for DIN[122]=ON;

    P01=CURPOS(0,0)

 DELAY 10
  OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
        WRITE file_var_S (0) --broj pravaca koji sjeku putanju
        WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

        CNV_REAL_STR(p01.x,2,2,s1) --točka starta
        WRITE file_var_S (s1)

        CNV_REAL_STR(p01.y,2,2,s1)
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.y,2,2,s1)
        WRITE file_var_S (s1)

        WRITE file_var_S (' m')          --poruka namijenjena matlabu
----------------------------------------
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------
```

```
DOUT[122]=OFF;

GOTO provjera1

ENDIF
----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

     IF sjec=3 THEN


--ako postoji neka pozicija, kreni u tu poziciju
    DELAY 10
      IF toc>0 THEN
         P04= POS(P02.x,P02.y,z1,w1,p1,r1, config_var1)

         FOR i=1 TO toc DO
            READ file_var_C (x1)
            WRITE('x:',x1::7::2,CR)
              P04.x=x1
            READ file_var_C (y1)
            WRITE('y:',y1::7::2,CR)
             P04.y=y1

            MOVE TO P04 NOWAIT
          WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P04.x-P00.x)<0.2) AND (ABS(P04.y-P00.y)<0.2)   THEN
            GOTO van4
          ENDIF
             DELAY 20
        ENDWHILE
      van4::

         ENDFOR
      ENDIF
  -- odlaganje predmeta

    MOVE TO P05 NOWAIT
         WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P05.x-P00.x)<0.2) AND (ABS(P05.y-P00.y)<0.2)   THEN
            GOTO van5
          ENDIF
             DELAY 20
        ENDWHILE
      van5::

    MOVE TO P06     NOWAIT
         WHILE TRUE DO
```

```
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P06.x-P00.x)<0.2) AND (ABS(P06.y-P00.y)<0.2)   THEN
            GOTO van6
          ENDIF
             DELAY 20
        ENDWHILE
      van6::
        DELAY 250
----------------------------------------
  RDO[1]=ON  -- otvori hvataljku
----------------------------------------
         DELAY 250
     MOVE TO P05 NOWAIT
        WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P05.x-P00.x)<0.2) AND (ABS(P05.y-P00.y)<0.2)   THEN
            GOTO van7
          ENDIF
             DELAY 20
        ENDWHILE
      van7::
       P01=CURPOS(0,0)

       GOTO provjera2

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------
provjera::

 DELAY 10
  OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
        WRITE file_var_S (0) --broj pravaca koji sjeku putanju
        WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

        CNV_REAL_STR(p01.x,2,2,s1) --točka starta
        WRITE file_var_S (s1)

        CNV_REAL_STR(p01.y,2,2,s1)
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.y,2,2,s1)
        WRITE file_var_S (s1)

        WRITE file_var_S (' m')          --poruka namijenjena matlabu
----------------------------------------
```

```
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------
DOUT[122]=OFF

GOTO provjera1

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

provjera2::
 DOUT[122]=OFF
  DELAY 10

    wait for DIN[122]=ON;
    DOUT[122]=ON;
 OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
        WRITE file_var_S (0) --broj pravaca koji sjeku putanju
        WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

        CNV_REAL_STR(p01.x,2,2,s1) --točka starta
        WRITE file_var_S (s1)

        CNV_REAL_STR(p01.y,2,2,s1)
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.y,2,2,s1)
        WRITE file_var_S (s1)

        WRITE file_var_S (' m')           --poruka namijenjena matlabu
----------------------------------------
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------
     WRITE('PROGRAM IZVRSEN!',CR)

GOTO van

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

--ako neki pravac sijeće putanju
 ELSE
        WRITE('Nije moguce izvesti putanju!',CR)

 ENDIF

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

GOTO van

 ENDIF

DELAY 20

  ENDWHILE
----------------------------------------
```

```
van::

 --stisni jedan da se pokrene program
   WRITE (CR,'POSTAVITE ROBOT U POCETNI POLOZAJ!',CR,CR)
   WRITE ('Pritisni 1 za POKRETANJE programa',CR)
   WRITE ('Pritisni 2 za IZLAZ iz programa',CR)

ENDIF

----------------------------------------
-- ako se pritisne 1 oilazi se iz programa
  IF TPIN[50]=ON THEN;
    WRITE (CR,'KRAJ programa',CR)
    GOTO izlaz
  ENDIF

DELAY 20

ENDWHILE

  izlaz::
----------------------------------------

  CLOSE_FILE_(file_var_C,'C7:') ;  --zatvaranje konekcije ako je ostala otvorena od prije


END DUB_R1_01
```

```
PROGRAM DUB_R2_04
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE

VAR
    STATUS,entry,port,portC: INTEGER
    file_var_s,file_var_c: FILE
    nb,i,sjec,toc:INTEGER
    r_f:BOOLEAN
    s,niz,robot,sje,k: STRING[25]
    s1:STRING[120]
    rand,x1,y1,z1,z2,w1,r,p1,r1,x_po,y_po,x_ci,y_ci:REAL
    P00,P01,P02,P03,P04,P05,P06,P07: XYZWPR
    config_var1:CONFIG



 --*** vanjske rutine za uspostavljanje TCP/ip konekcije
  ROUTINE OPEN_FILE_ (FILE_ : FILE; TAG_ : STRING) FROM LIB_FILE
  ROUTINE CLOSE_FILE_(FILE_ : FILE; TAG_ : STRING) FROM LIB_FILE

BEGIN

    $GROUP[1].$UFRAME = $MNUFRAME[1,4];
    $GROUP[1].$UTOOL =  $MNUTOOL[1,5]
    $GROUP[1].$MOTYPE=LINEAR;
    $GROUP[1].$SPEED=400
    $GROUP[1].$TERMTYPE=FINE;

       --*********************************************************
       --o kojem je robotu riječ
       GET_VAR(entry,'*SYSTEM*', '$HOSTNAME',s,STATUS)
       IF s='ROBOT1' THEN;robot='R1_';port=10257;ENDIF;
       IF s='ROBOT1' THEN;robot='R1_';portC=10258;ENDIF;
       IF s='ROBOT2' THEN;robot='R2_';port=10267;ENDIF;
       IF s='ROBOT2' THEN;robot='R2_';portC=10268;ENDIF;
       --*********************************************************

   --zaustavi sve tagove prije promjene!!!
    -- postavljanje vrijenosti za svaki tak preko kojega će se vršiti komunikacija sa drugim
uređajem
    -- pomoću TCP/IP protokola     'C7:'    i 'S7:'
    -- moguće je definirati do 8 tagova za klijente 'C1:'-'C8:'   i 8 tagova za serverske
konekcije  'S1:'-'S8:'

    -- setiranje atributa za tak S7:
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$OPER',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STATE',0,STATUS) ; DELAY 20
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$COMMENT','SOUND',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$PROTOCOL','SM',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$REPERRS','FALSE',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$TIMEOUT',9999,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$PWRD_TIMEOUT',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$SERVER_PORT',port,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STRT_PATH','192.168.123.39',STATUS)--ip
adresa uređaja sa kojim se komunicira
       SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STRT_REMOTE','192.168.123.39',STATUS) ;--
-ll-

      -- setiranje atributa za tak C7:
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$OPER',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STATE',0,STATUS) ; DELAY 20
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$COMMENT','SOUND',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$PROTOCOL','SM',STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$REPERRS','FALSE',STATUS) ;
```

```
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$TIMEOUT',9999,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$PWRD_TIMEOUT',0,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$SERVER_PORT',portC,STATUS) ;
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$PATH','192.168.123.39',STATUS) ;--ip adresa
uređaja sa kojim se komunicira
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STRT_PATH','192.168.123.39',STATUS) ;  --
-ll-
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$remote','192.168.123.39',STATUS) ;
        --        -ll-
       SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STRT_REMOTE','192.168.123.39',STATUS) ;--
-ll-

    --inicijaliziraj ponovo tagove 7 kada si napravio promjenu

    DELAY 10 ;

      SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$OPER',3,STATUS);
      SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[7].$STATE',3,STATUS) ;

    DELAY 10 ;

      SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$OPER',3,STATUS);
      SET_VAR(entry,'*SYSTEM*','$HOSTC_CFG[7].$STATE',3,STATUS) ;

 DELAY 10;

    DOUT[121]=OFF;
    DOUT[122]=OFF;

    CLOSE_FILE_(file_var_C,'C7:') ;  --zatvaranje konekcije ako je ostala otvorena od prije

DELAY 10

    OPEN_FILE_(file_var_C,'C7:');  --otvaranje konekcije

-----------------------------------------
--stisni jedan da se pokrene program
   WRITE (CR,'POSTAVITE ROBOT U POCETNI POLOZAJ!',CR,CR)
   WRITE ('Pritisni 1 za POKRETANJE programa',CR)
   WRITE ('Pritisni 2 za IZLAZ iz programa',CR)

   WHILE TRUE DO

 DELAY 10

  IF TPIN[49]=ON THEN;

   WRITE (CR,'IZVRŠAVANJE programa..',CR)

--inicijalizacija varijabli
    s1=''
    x1=0
    y1=0
    z1=20
    z2=5
    w1=0
    p1=0
    r1=0
    DOUT[121]=OFF;
    DOUT[122]=OFF;

-------------------------
  RDO[1]=ON  -- otvori hvataljku
-------------------------
```

```
-- startna pozicija robota (hvataljka otvoena)
   P01=CURPOS(0,0)     --spremanje trenutne pozicije robota u varijablu p1

-- startna pozicija robota (hvataljka otvoena)
  CNV_STR_CONF('nut000', config_var1, STATUS)
     P02= POS(P01.x,P01.y,z1,w1,p1,r1,config_var1)
----------------------------------------

-- startna pozicija robota (hvataljka zatvorena)
  CNV_STR_CONF('nut000', config_var1, STATUS)
     P03= POS(P02.x,P02.y,z2,w1,p1,r1,config_var1)
----------------------------------------

-- ciljna pozicija robota (hvataljka otvoena)
  CNV_STR_CONF('nut000', config_var1, STATUS)
 --    x_ci=650;y_ci=450;
      x_ci=300;y_ci=480;
     P05= POS(x_ci,y_ci,z1,w1,p1,r1, config_var1)
----------------------------------------

-- ciljna pozicija robota (hvataljka zatvorena)
  CNV_STR_CONF('nut000', config_var1, STATUS)

     P06= POS(x_ci,y_ci,z2,w1,p1,r1, config_var1)
----------------------------------------

-- ciljna pozicija robota (hvataljka zatvorena)
  CNV_STR_CONF('nut000', config_var1, STATUS)

     P07= POS(x_ci,y_ci,z1,w1,p1,r1, config_var1)
----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------
      wait for DIN[121]=ON;
      DOUT[121]=ON;


 DELAY 10
  OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
       WRITE file_var_S (0) --broj pravaca koji sjeku putanju
       WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

       CNV_REAL_STR(p02.x,2,2,s1) --točka starta
       WRITE file_var_S (s1)

       CNV_REAL_STR(p02.y,2,2,s1)
       WRITE file_var_S (s1)

       CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
       WRITE file_var_S (s1)

       CNV_REAL_STR(p05.y,2,2,s1)
       WRITE file_var_S (s1)

       WRITE file_var_S (' m')          --poruka namijenjena matlabu
----------------------------------------
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------

provjera1::


 DELAY 10
```

```
 WHILE TRUE DO

       BYTES_AHEAD(file_var_C,nb,STATUS)  --čitanje koliko ima bajtova u međuspremniku

       DELAY 10 -- čekanje

       IF nb>3 THEN  -- ako ima više od 2 bajta u međuspremniku čitaj što je u međuspremniku
u varijablu niz

--čitanje da li koji pravac sijeće putanju
        READ file_var_C (sjec)
        WRITE('Broj tocaka koje sijeku:',sjec::7::0,CR)
----------------------------------------

--čitanje koliko ima pozicija
        READ file_var_C (toc)
        WRITE('Broj tocaka(prolazi):',toc::7::0,CR)

--ako niti jedan pravac ne sjece putanju
DELAY 10

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

   IF sjec=0 THEN


--ako postoji neka pozicija, kreni u tu poziciju
    DELAY 10
      IF toc>0 THEN
         P04= POS(P02.x,P02.y,z1,w1,p1,r1, config_var1)

         FOR i=1 TO toc DO
            READ file_var_C (x1)
            WRITE('x:',x1::7::2,CR)
              P04.x=x1
            READ file_var_C (y1)
            WRITE('y:',y1::7::2,CR)
             P04.Y=y1

            MOVE TO P04

         ENDFOR
      ENDIF
  -- odlaganje predmeta

    MOVE TO P05
    MOVE TO P06
        DELAY 250
----------------------------------------
  RDO[1]=ON  -- otvori hvataljku
----------------------------------------
         DELAY 250
     MOVE TO P05
     WRITE('PROGRAM IZVRSEN!',CR)
     DOUT[121]=OFF;
     DOUT[122]=OFF;
  ENDIF

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

     IF sjec=3 THEN
```

```
--ako postoji neka pozicija, kreni u tu poziciju
    DELAY 10
      IF toc>0 THEN
         P04= POS(P02.x,P02.y,z1,w1,p1,r1, config_var1)

         FOR i=1 TO toc DO
            READ file_var_C (x1)
            WRITE('x:',x1::7::2,CR)
              P04.x=x1
            READ file_var_C (y1)
            WRITE('y:',y1::7::2,CR)
             P04.y=y1

            MOVE TO P04 NOWAIT
          WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P04.x-P00.x)<0.2) AND (ABS(P04.y-P00.y)<0.2)   THEN
            GOTO van4
          ENDIF
             DELAY 20
        ENDWHILE
      van4::

         ENDFOR
      ENDIF
  -- odlaganje predmeta

    MOVE TO P05 NOWAIT
         WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P05.x-P00.x)<0.2) AND (ABS(P05.y-P00.y)<0.2)   THEN
            GOTO van5
          ENDIF
             DELAY 20
        ENDWHILE
      van5::

    MOVE TO P06     NOWAIT
         WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P06.x-P00.x)<0.2) AND (ABS(P06.y-P00.y)<0.2)   THEN
            GOTO van6
```

```
          ENDIF
             DELAY 20
        ENDWHILE
      van6::
        DELAY 250
----------------------------------------
  RDO[1]=ON  -- otvori hvataljku
----------------------------------------
         DELAY 250
     MOVE TO P05 NOWAIT
        WHILE TRUE DO
          IF DIN[122]=ON THEN
            DOUT[122]=ON
            CANCEL
            DELAY 20
            P01=CURPOS(0,0)
            GOTO provjera
          ENDIF
          P00=CURPOS(0,0)
          IF (ABS(P05.x-P00.x)<0.2) AND (ABS(P05.y-P00.y)<0.2)   THEN
            GOTO van7
          ENDIF
             DELAY 20
        ENDWHILE
      van7::
       P01=CURPOS(0,0)

       GOTO provjera2
  ENDIF
----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

    IF sjec=2 THEN


--ako postoji neka pozicija, kreni u tu poziciju
     DELAY 10
       IF toc>0 THEN
           P04= POS(P02.x,P02.y,z1,w1,p1,r1, config_var1)

      FOR i=1 TO toc DO
        READ file_var_C (x1)
        WRITE('x:',x1::7::2,CR)
          P04.x=x1
        READ file_var_C (y1)
        WRITE('y:',y1::7::2,CR)
          P04.Y=y1

        MOVE TO P04

      ENDFOR
  ENDIF
-- odlaganje predmeta
        READ file_var_C (x1)
        WRITE('x:',x1::7::2,CR)
          P07.x=x1
        READ file_var_C (y1)
        WRITE('y:',y1::7::2,CR)
          P07.y=y1
    MOVE TO P07

    DOUT[122]=ON;
    wait for DIN[122]=ON;
```

```
    P01=CURPOS(0,0)

 DELAY 10
  OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
        WRITE file_var_S (0) --broj pravaca koji sjeku putanju
        WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

        CNV_REAL_STR(p01.x,2,2,s1) --točka starta
        WRITE file_var_S (s1)

        CNV_REAL_STR(p01.y,2,2,s1)
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.y,2,2,s1)
        WRITE file_var_S (s1)

        WRITE file_var_S (' m')          --poruka namijenjena matlabu
----------------------------------------
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------
DOUT[122]=OFF;

GOTO provjera1
----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------
provjera::

 DELAY 10
  OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
        WRITE file_var_S (0) --broj pravaca koji sjeku putanju
        WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

        CNV_REAL_STR(p01.x,2,2,s1) --točka starta
        WRITE file_var_S (s1)

        CNV_REAL_STR(p01.y,2,2,s1)
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.y,2,2,s1)
        WRITE file_var_S (s1)

        WRITE file_var_S (' m')          --poruka namijenjena matlabu
----------------------------------------
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------
DOUT[122]=OFF

GOTO provjera1
----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------

provjera2::
 DOUT[122]=OFF
```

```
  DELAY 10

    wait for DIN[122]=ON;
    DOUT[122]=ON;
 OPEN_FILE_(file_var_S,'S7:');

 DELAY 10
        WRITE file_var_S (0) --broj pravaca koji sjeku putanju
        WRITE file_var_S (0) --broj pozicija kroz koje robot prolazi

        CNV_REAL_STR(p01.x,2,2,s1) --točka starta
        WRITE file_var_S (s1)

        CNV_REAL_STR(p01.y,2,2,s1)
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.x,2,2,s1)  --točka cilja
        WRITE file_var_S (s1)

        CNV_REAL_STR(p05.y,2,2,s1)
        WRITE file_var_S (s1)

        WRITE file_var_S (' m')          --poruka namijenjena matlabu
----------------------------------------
  CLOSE_FILE_(file_var_S,'S7:');
----------------------------------------
     WRITE('PROGRAM IZVRSEN!',CR)

GOTO van

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------
--ako neki pravac sijeće putanju
 ELSE
        WRITE('Nije moguce izvesti putanju!',CR)

 ENDIF

----------------------------------------
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------------------



GOTO van

       ENDIF

DELAY 20

  ENDWHILE
----------------------------------------

van::

--stisni jedan da se pokrene program
   WRITE (CR,'POSTAVITE ROBOT U POCETNI POLOZAJ!',CR,CR)
   WRITE ('Pritisni 1 za POKRETANJE programa',CR)
   WRITE ('Pritisni 2 za IZLAZ iz programa',CR)


  ENDIF

----------------------------------------
```

```
-- ako se pritisne 1 oilazi se iz programa
 IF TPIN[50]=ON THEN;
    WRITE (CR,'KRAJ programa',CR)
    GOTO izlaz
 ENDIF

DELAY 20

ENDWHILE

  izlaz::
----------------------------------------

  CLOSE_FILE_(file_var_C,'C7:') ;  --zatvaranje konekcije ako je ostala otvorena od prije


END DUB_R2_01
```

1.2 Matlab

```
%MATLAB_1
%glavni program
%povezivanje matlaba i ROBOTA1 i ROBOTA2
%primanje poruke od oba robota
%pozivanje programa za izračun najboljeg pojedinca/putanje
%slanje poruke robotima


format bank  %ispisiva sve na dvije decimale

global server_socket1;
global output_socket1;
global server_socket2;
global output_socket2;

%deklariranje varijabli za prvi robot
ip1='192.168.123.25'; % IP adresa robota '192.168.123.25'
port_c1=10257;   %podaci robot->matlab
port_s1=10258;   %podaci matlab->robot
%-------------------------------------------

%deklariranje varijabli za drugi robot
ip2='192.168.123.26'; % IP adresa robota '192.168.123.26'
port_c2=10267;   %podaci robot->matlab
port_s2=10268;   %podaci matlab->robot
%-------------------------------------------
retry1=30;
retry2=100;
%-------------------------------------------

% Veličina radnog prostora robota

duz_prostora=900;      %dužina radnog prostora robota
sir_prostora=1200;     %širina radnog prostora robota
%-------------------------------------------

% Veličina hvataljke PRVOG robota (promjer hvataljke)
d_h_1=130;
% Veličina hvataljke DRUGOG robota (promjer hvataljke)
d_h_2=130;
%-------------------------------------------
% Širina PRVOG robota
d_s_1=120;
% Širina DRUGOG robota
d_s_2=120;

f_h=1.6; %faktor sigurnosti za odmicanje robota
%-------------------------------------------

r_h_1=d_h_1/2;
r_h_2=d_h_2/2;
r_s_1=d_s_1/2;
```

```
r_s_2=d_s_2/2;

%-------------------------------------------

%povezivanje MATLAB_1 i ROBOT_1

server_open_1_R1(port_s1,retry1)
%-------------------------------------------------

pause(0.4)

%povezivanje MATLAB_1 i ROBOT_2

server_open_1_R2(port_s2,retry1)
%-------------------------------------------------

pause(0.5)
con1=0;con2=0;

while true

if con1~=1
    nova_poz1=0;a1=0;b1=0;c1=0;d1=0;e1=0;g1=3;h1=0;j1=0;

    [con1]=client_open_1_R1(ip1, port_c1,retry2)
end

if con2~=1
    nova_poz2=0;a2=0;b2=0;c2=0;d2=0;e2=0;g2=3;h2=0;j2=0;

    [con2]=client_open_1_R2(ip2, port_c2,retry2)
end

%-------------------------------------------------

% ako je uspjela konekcija
if con1==1 & con2==1

            pause(0.2)
            %primanje poruke od ROBOTA_1
            [message_M_d1,message_c1,h1]=client_receive_1_R1(ip1,port_c1,retry1)

            pause(0.2)

            %primanje poruke od ROBOTA_2
            [message_M_d2,message_c2,h2]=client_receive_1_R2(ip2,port_c2,retry2)

    tic;

  pause(0.2)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% R O B O T   1
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%ako poruka iz PRVOG ROBOTA nije prazna
     if h1>0

         j1=message_c1(1,h1);

%ako je u poruci zadnje slovo m
 if j1=='m'
%-------------------------------------------------
    [a1,b1]=size(message_M_d1); %brojevi
    c1=(b1-2)/2;  %broj točaka

    br_pozicija=message_M_d1(1,1);
    prep_sijeku=message_M_d1(1,2);
    t_1_x_poc_1=message_M_d1(1,3);
    t_1_y_poc_1=message_M_d1(1,4);
    t_1_x_kraj_1=message_M_d1(1,5);
    t_1_y_kraj_1=message_M_d1(1,6);

%-------------------------------------------------
%ako se slučajno preklapaju pozicije
    if   t_1_x_poc_1== t_1_x_kraj_1

        t_1_x_poc_1=t_1_x_poc_1+0.00001;

    end
%-------------------------------------------------
    end
 end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% R O B O T   2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%ako poruka iz DRUGOG ROBOTA nije prazna
     if h2>0

         j2=message_c2(1,h2);

%ako je u poruci zadnje slovo m
 if j2=='m'
%-------------------------------------------------
    [a2,b2]=size(message_M_d2); %brojevi
    c2=(b2-2)/2;  %broj točaka

    br_pozicija=message_M_d2(1,1);
    prep_sijeku=message_M_d2(1,2);
    t_1_y_poc_2=message_M_d2(1,3);
    t_1_x_poc_2=duz_prostora-message_M_d2(1,4);
    t_1_y_kraj_2=message_M_d2(1,5);
    t_1_x_kraj_2=duz_prostora-message_M_d2(1,6);

%-------------------------------------------------
```

```
%ako se slučajno preklapaju pozicije
    if   t_1_x_poc_2== t_1_x_kraj_2

        t_1_x_poc_2=t_1_x_poc_2+0.00001;

    end
%-------------------------------------------------

 end
     end


%ako su primljene poruke, izračunaj putanje

if h1>0 & h2>0 & j1=='m' & j2=='m'


[nove_pozicije1,nove_pozicije2,prep_sijeku1,prep_sijeku2,toc_1_kraj_1,toc_1_kraj_2]
=prog_robot(t_1_x_poc_1,t_1_y_poc_1,t_1_x_kraj_1,t_1_y_kraj_1,t_1_x_poc_2,t_1_y_poc
_2,t_1_x_kraj_2,t_1_y_kraj_2,duz_prostora,sir_prostora,r_h_1,r_h_2,r_s_1,r_s_2,f_h)
;



 %ROBOT 1
  [d1,e1]=size(nove_pozicije1);
  f1=e1-2; %broj pozicija

 nova_poz1(1,1)=prep_sijeku1;
 nova_poz1(1,2)=f1;

if (f1~=0 & (prep_sijeku1==0 | prep_sijeku1==3))

        for i=2:1:(e1-1)
            nova_poz1(1,g1)=nove_pozicije1(1,i);
            nova_poz1(1,g1+1)=nove_pozicije1(2,i);
            g1=g1+2;
        end

elseif (f1==0 & prep_sijeku1==2)
    nova_poz1(1,g1)=toc_1_kraj_1(1);
    nova_poz1(1,g1+1)=toc_1_kraj_1(2);
    g1=g1+2;

elseif (f1~=0 & prep_sijeku1==2)
        for i=2:1:(e1-1)
            nova_poz1(1,g1)=nove_pozicije1(1,i);
            nova_poz1(1,g1+1)=nove_pozicije1(2,i);
            g1=g1+2;
        end
    nova_poz1(1,g1)=toc_1_kraj_1(1);
    nova_poz1(1,g1+1)=toc_1_kraj_1(2);
    g1=g1+2;
end

nova_poz1(1,g1)=0;
```

```
nova_poz1
%-------------------------------------------------


 %ROBOT 2
  [d2,e2]=size(nove_pozicije2);
  f2=e2-2; %broj pozicija

 nova_poz2(1,1)=prep_sijeku2;
 nova_poz2(1,2)=f2;

if (f2~=0 & (prep_sijeku2==0 | prep_sijeku2==3))

        for i=2:1:(e2-1)
            nova_poz2(1,g2)=nove_pozicije2(2,i);
            nova_poz2(1,g2+1)=duz_prostora-nove_pozicije2(1,i);
            g2=g2+2;
        end

elseif (f2==0 & prep_sijeku2==2)
    nova_poz2(1,g2)=toc_1_kraj_2(2);
    nova_poz2(1,g2+1)=toc_1_kraj_2(1);
    g2=g2+2;

elseif (f2~=0 & prep_sijeku2==2)

        for i=2:1:(e2-1)
            nova_poz2(1,g2)=nove_pozicije2(2,i);
            nova_poz2(1,g2+1)=duz_prostora-nove_pozicije2(1,i);
            g2=g2+2;
        end
    nova_poz2(1,g2)=toc_1_kraj_2(1);
    nova_poz2(1,g2+1)=toc_1_kraj_2(2);
    g2=g2+2;
end

nova_poz2(1,g2)=0;

nova_poz2
%-------------------------------------------------

%slanje poruke ROBOTU_1

     pozicija_M1=num2str(nova_poz1);
     server_M_send2_R1(pozicija_M1);
%-------------------------------------------------
%slanje poruke ROBOTU_2

     pozicija_M2=num2str(nova_poz2);
     server_M_send2_R2(pozicija_M2);
%-------------------------------------------------

 con1=0;
 con2=0;
```

```
end


t=toc

     end %zatvaranje konekcija

end  %zatvaranje while
```

```
% CLIENT connect to a server and read a message
%
% Usage - message = client(host, port, number_of_retries)
function [con1]=f(host1, port1, number_of_retries1)

    import java.net.Socket
    import java.io.*

    global input_socket1

    retry1        = 0;
    input_socket1 = [];
    message1      = [];
    con1=0;
    while true

        retry1 = retry1 + 1;
        if ((number_of_retries1 > 0) && (retry1 > number_of_retries1))
            fprintf(1, 'Too many retries\n');
            break;
        end

        try
            fprintf(1, 'Retry %d connecting to %s:%d\n', ...
                    retry1, host1, port1);

            % throws if unable to connect
            input_socket1 = Socket(host1, port1);

            fprintf(1, 'Connected at 192.168.123.25\n')
            con1=1;
            break
            pause(10000000)

            break

        catch
            if ~isempty(input_socket1)
                input_socket1.close;
            end

            % pause before retrying
            pause(0.05);
```

```
        end
    end
end
```

```
% CLIENT connect to a server and read a message
%
% Usage - message = client(host, port, number_of_retries)
function [con2]=f(host1, port1, number_of_retries1)

    import java.net.Socket
    import java.io.*

    global input_socket2

    retry1        = 0;
    input_socket2 = [];
    message1      = [];
    con2=0;
    while true

        retry1 = retry1 + 1;
        if ((number_of_retries1 > 0) && (retry1 > number_of_retries1))
            fprintf(1, 'Too many retries\n');
            break;
        end

        try
            fprintf(1, 'Retry %d connecting to %s:%d\n', ...
                    retry1, host1, port1);

            % throws if unable to connect
            input_socket2 = Socket(host1, port1);

            fprintf(1, 'Connected at 192.168.123.26\n')
            con2=1;
            break
            pause(10000000)

            break

        catch
            if ~isempty(input_socket2)
                input_socket2.close;
            end

            % pause before retrying
            pause(0.05);
        end
    end
end
```

```
% CLIENT connect to a server and read a message
%
% Usage - message = client(host, port, number_of_retries)
function [message_c_d,message_c,e] = f(host1, port1, number_of_retries1)

    import java.net.Socket
    import java.io.*

    global input_socket1
    global message_c

    d=0;
    e=0;
    message_c='00';
    message_cc='00';
    message_c_d=0;

            % get a buffered data input stream from the socket
            input_stream1   = input_socket1.getInputStream;
            d_input_stream1 = DataInputStream(input_stream1);

            % read data from the socket - wait a short time first
            bytes_available = input_stream1.available;
            %fprintf(1, 'Reading %d bytes\n', bytes_available);

            message_c = zeros(1, bytes_available, 'uint8');
            for ii = 1:bytes_available
                message_c(ii) = d_input_stream1.readByte;
            end

            message_c = char(message_c);
            %fprintf(message_c, '\n');

            [d,e]=size(message_c); %string

            if e>0
                for i=1:1:e-1
                    message_cc(1,i)=message_c(i);
                end

             message_c_d=str2num(message_cc);

            end
end
```

```
% CLIENT connect to a server and read a message
%
% Usage - message = client(host, port, number_of_retries)
function [message_c_d,message_c,e] = f(host1, port1, number_of_retries1)

    import java.net.Socket
    import java.io.*

    global input_socket2
    global message_c

    d=0;
    e=0;
    message_c='00';
    message_cc='00';
    message_c_d=0;

            % get a buffered data input stream from the socket
            input_stream2   = input_socket2.getInputStream;
            d_input_stream2 = DataInputStream(input_stream2);

            % read data from the socket - wait a short time first
            bytes_available = input_stream2.available;
            %fprintf(1, 'Reading %d bytes\n', bytes_available);

            message_c = zeros(1, bytes_available, 'uint8');
            for ii = 1:bytes_available
                message_c(ii) = d_input_stream2.readByte;
            end

            message_c = char(message_c);
            %fprintf(message_c, '\n');

            [d,e]=size(message_c); %string

            if e>0
                for i=1:1:e-1
                    message_cc(1,i)=message_c(i);
                end

             message_c_d=str2num(message_cc);

            end
end
```

```
% SERVER Write a message over the specified port
%
% Usage - server(message, output_port, number_of_retries)
function server_M_send2_R1(message)
    import java.net.ServerSocket
    import java.io.*

     global  output_socket1 output_stream1

           output_stream1   = output_socket1.getOutputStream;
            d_output_stream1 = DataOutputStream(output_stream1);
            fprintf(1, 'Writing %d bytes\n', length(message))
            d_output_stream1.writeBytes(char(message));
            d_output_stream1.flush;
            pause (0.2)

end
```

```
% SERVER Write a message over the specified port
%
% Usage - server(message, output_port, number_of_retries)
function server_M_send2_R2(message)
    import java.net.ServerSocket
    import java.io.*

     global  output_socket2 output_stream2

           output_stream2   = output_socket2.getOutputStream;
            d_output_stream2 = DataOutputStream(output_stream2);
            fprintf(1, 'Writing %d bytes\n', length(message))
            d_output_stream2.writeBytes(char(message));
            d_output_stream2.flush;
            pause (0.2)

end
```

```
% SERVER Connect to a specified port
%
% Usage - server(output_port, number_of_retries)
function server_open_1_R1(output_port, number_of_retries)
    import java.net.ServerSocket
    import java.io.*
    global  output_stream1 server_socket1 output_socket1

    server_socket1=[];
    output_socket1=[];
```

```
    retry             = 0;

    while true

        retry = retry + 1;

        try
            if ((number_of_retries > 0) && (retry > number_of_retries))
                fprintf(1, 'Too many retries\n');
                break;
            end

            fprintf(1, ['S2 Try %d waiting for client to connect  ' ...
                        ' on port : %d\n'], retry, output_port);

            server_socket1 = ServerSocket(output_port); %ovo ih spoji
            server_socket1.setSoTimeout(500);
            output_socket1 = server_socket1.accept;

            fprintf(1, 'Client connected\n');

            output_stream1   = output_socket1.getOutputStream;
            d_output_stream1 = DataOutputStream(output_stream1);

            break;

            pause(500000000);

            server_socket1.close;
            output_socket1.close;
            break;

        catch
            if ~isempty(server_socket1)
                server_socket1.close
            end

            if ~isempty(output_socket1)
                output_socket1.close
            end

            % pause before retrying
            pause(0.5);
        end
    end
end
```

```
% SERVER Connect to a specified port
%
% Usage - server(output_port, number_of_retries)
function server_open_1_R2(output_port, number_of_retries)
    import java.net.ServerSocket
```

```
    import java.io.*
    global  output_stream2 server_socket2 output_socket2

    server_socket2=[];
    output_socket2=[];

    retry             = 0;

    while true

        retry = retry + 1;

        try
            if ((number_of_retries > 0) && (retry > number_of_retries))
                fprintf(1, 'Too many retries\n');
                break;
            end

            fprintf(1, ['S2 Try %d waiting for client to connect  ' ...
                        ' on port : %d\n'], retry, output_port);

            server_socket2 = ServerSocket(output_port); %ovo ih spoji
            server_socket2.setSoTimeout(500);
            output_socket2 = server_socket2.accept;

            fprintf(1, 'Client connected\n');

            output_stream2   = output_socket2.getOutputStream;
            d_output_stream2 = DataOutputStream(output_stream2);

            break;

            pause(500000000);

            server_socket2.close;
            output_socket2.close;
            break;

        catch
            if ~isempty(server_socket2)
                server_socket2.close
            end

            if ~isempty(output_socket2)
                output_socket2.close
            end

            % pause before retrying
            pause(0.5);
        end
    end
end
```

```
%crtanje putanje oba robota s preprekama

function [crtaj_konf] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz,sir,xpr,ypr,naj_pojed_1,naj
_pojed_2)

a=0;
b=0;
a1=0;
b=10;
a2=0;
b=20;
%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće PRVI robot

px11=[0 toc_1_poc_1(1)];       %x koord početne i krajnje točke pravca p1
py11=[(sir/2) toc_1_poc_1(2)]; %y koord početne i krajnje točke pravca p1

px12=[0 toc_1_kraj_1(1)];       %x koord početne i krajnje točke pravca p2
py12=[(sir/2) toc_1_kraj_1(2)]; %y koord početne i krajnje točke pravca p2

px13=[px11(2) px12(2)];
py13=[py11(2) py12(2)];

%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće DRUGI robot

px21=[duz toc_1_poc_2(1)];       %x koord početne i krajnje točke pravca p1
py21=[(sir/2) toc_1_poc_2(2)];   %y koord početne i krajnje točke pravca p1

px22=[duz toc_1_kraj_2(1)];      %x koord početne i krajnje točke pravca p2
py22=[(sir/2) toc_1_kraj_2(2)];  %y koord početne i krajnje točke pravca p2

px23=[px21(2) px22(2)];
py23=[py21(2) py22(2)];

%-------------------------------------------
%crtanje radnog prosrtora prvog i drugog robota

figure(15)
axis equal
title 'PUTANJA PRVOG i DRUGOG ROBOTA'
hold on
axis([0 duz 0 sir])

line(px11,py11,'Color','b','LineWidth',1);  %pravac p1 ROBOT 1
line(px12,py12,'Color','b','LineWidth',1);  %pravac p2 ROBOT 1

line(px21,py21,'Color','r','LineWidth',1);  %pravac p1 ROBOT 2
line(px22,py22,'Color','r','LineWidth',1);  %pravac p2 ROBOT 2

%-------------------------------------------
%definiranje prepreka u matričnom obliku
```

```
 mat_prepreka=[xpr;ypr];


 %-------------------------------------------
 %crtanje prepreka

  [a,b]=size(mat_prepreka);

      for i=1:2:b

        prx(1,1)=mat_prepreka(1,i);
        pry(1,1)=mat_prepreka(2,i);

        prx(1,2)=mat_prepreka(1,i+1);
        pry(1,2)=mat_prepreka(2,i+1);

        line(prx,pry,'Color','k','LineWidth',1);
      end

%-------------------------------------------
%crtanje najboljeg pojedinca ROBOT 1

  [a1,b1]=size(naj_pojed_1);

 for i=1:b1
    x1(i)= naj_pojed_1(1,i);
    y1(i)=naj_pojed_1(2,i);
end
t1=1:b1;
ts1=1:1/100:b1;
xs1=spline(t1,x1,ts1);
ys1=spline(t1,y1,ts1);
hold on
plot(xs1,ys1,'Color','g','LineWidth',2);
hold off

%-------------------------------------------
%crtanje najboljeg pojedinca ROBOT 2

  [a2,b2]=size(naj_pojed_2);


 for i=1:b2
    x2(i)= naj_pojed_2(1,i);
    y2(i)=naj_pojed_2(2,i);
end
t2=1:b2;
ts2=1:1/100:b2;
xs2=spline(t2,x2,ts2);
ys2=spline(t2,y2,ts2);
hold on
plot(xs2,ys2,'Color','g','LineWidth',2);
hold off
crtaj_konf=1;
```

```
%ROBOT 1

%funkcija koja ispituje da li se prepreka nalazi unurat, izvan ili u radnom
%prostoru robota

function
[prep_sijece_oba,prep_sijece_gore,prep_sijece_dole,prep_vani,prep_unutar,mat_prep_u
nutar] = f(toc_1_poc,toc_1_kraj,xpr,ypr,br_prepreka,duz,sir)

%-------------------------------------------
%zadavanje početnih uvjeta
a=0; e=0; f=0; g=0; h=1;
prep_sijece_oba=0;
prep_sijece_dole=0;
prep_sijece_gore=0;
prep_unutar=0;
prep_vani=0;
mat_prep_unutar=0;
%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće robot

px1=[0 toc_1_poc(1)];       %x koord početne i krajnje točke pravca p1
py1=[(sir/2) toc_1_poc(2)]; %y koord početne i krajnje točke pravca p1

px2=[0 toc_1_kraj(1)];       %x koord početne i krajnje točke pravca p2
py2=[(sir/2) toc_1_kraj(2)]; %y koord početne i krajnje točke pravca p2

px3=[px1(2) px2(2)];
py3=[py1(2) py2(2)];

%-------------------------------------------
%definiranje prepreka u matričnom obliku

mat_pr=[xpr;ypr];

%-------------------------------------------
%crtanje radnog prosrtora

figure(7)
title 'RADNI PROSTOR PRVOG ROBOTA S PREPREKAMA'
axis equal
hold on
axis([0 duz 0 sir])

line(px1,py1,'Color','b','LineWidth',1);  %pravac p1
line(px2,py2,'Color','b','LineWidth',1);  %pravac p2
line(px3,py3,'Color','c','LineWidth',1);  %pravac p3

%-------------------------------------------
%definiranje prve i druge točke prepreke

[c,d]=size(mat_pr);
```

```
for i=1:2:d
    a=a+2;
      if
(mat_pr(1,i)>mat_pr(1,a))|((mat_pr(1,i)==mat_pr(1,a))&(mat_pr(2,i)<mat_pr(2,a)))
         mat_prep(:,i)=mat_pr(:,a);
         mat_prep(:,a)=mat_pr(:,i);
        else
         mat_prep(:,i)=mat_pr(:,i);
         mat_prep(:,a)=mat_pr(:,a);
      end
end

mat_prep;

%---------------------------------------------
%crtanje prepreka

for i=1:2:d
    prx(1,1)=mat_prep(1,i);
    pry(1,1)=mat_prep(2,i);

    prx(1,2)=mat_prep(1,i+1);
    pry(1,2)=mat_prep(2,i+1);

line(prx,pry,'Color','k','LineWidth',1);
end
hold off

%---------------------------------------------
%k - koeficijenti smijera -> k=tg?

k1=((py1(2)-py1(1))+0.0000001)/((px1(2)-px1(1))+0.0000001);
k2=((py2(2)-py2(1))+0.0000001)/((px2(2)-px2(1))+0.0000001);
k3=((py1(2)-py2(2))+0.0000001)/((px1(2)-px2(2))+0.0000001);

%---------------------------------------------
%b - odsječci na y osi

b1=py1(2)-k1*px1(2);
b2=py2(2)-k2*px2(2);
b3=py1(2)-k3*px1(2);

%---------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:2:d
    e=e+1;
   y_prep_poc_rec_p1(e)=k1*mat_prep(1,i)+b1;
   y_prep_poc_rec_p2(e)=k2*mat_prep(1,i)+b2;
   y_prep_poc_rec_p3(e)=k3*mat_prep(1,i)+b3;
end

for i=2:2:d
    f=f+1;
   y_prep_kraj_rec_p1(f)=k1*mat_prep(1,i)+b1;
```

```
   y_prep_kraj_rec_p2(f)=k2*mat_prep(1,i)+b2;
   y_prep_kraj_rec_p3(f)=k3*mat_prep(1,i)+b3;
end

h01=0;
h02=0;


h011=0;
h012=0;
h021=0;
h022=0;
h11=0;
h12=0;
h21=0;
h22=0;

%---------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:2:d
     g=g+1;

%---------------------------------------------
%prepreka sijeće oba pravca

     if (mat_prep(2,i)>y_prep_poc_rec_p1(1,g) &
mat_prep(2,i+1)<y_prep_kraj_rec_p2(1,g))|(mat_prep(2,i+1)>y_prep_kraj_rec_p1(1,g) &
mat_prep(2,i)<y_prep_poc_rec_p2(1,g)& mat_prep(1,i+1)>mat_prep(1,i))


      if (mat_prep(2,i)< y_prep_poc_rec_p3(1,g) & mat_prep(2,i+1)<
y_prep_kraj_rec_p3(1,g)& px1(2)<px2(2))|(mat_prep(2,i)> y_prep_poc_rec_p3(1,g) &
mat_prep(2,i+1)> y_prep_kraj_rec_p3(1,g)& px1(2)>px2(2))|(mat_prep(1,i)<px1(2)&
mat_prep(1,i+1)<px1(2)&px1(2)==px2(2))
             disp('Prepreka sijeće oba pravca.');
             prep_sijece_oba=prep_sijece_oba+1;

     %---------------------------------------------
     %oba pravca, A je lijevo od pravca, B je desno od pravca

      elseif mat_prep(2,i)>y_prep_poc_rec_p1(1,g)

     %gornja tocka
       h011=h011+1;
       k5011(h011)=((py1(2)-mat_prep(2,i)))/((px1(2)-mat_prep(1,i)));
       b5011(h011)=mat_prep(2,i)-(k5011(h011)*mat_prep(1,i));
       y_prep_rec_p5011(h011)=k5011(h011)*mat_prep(1,i+1)+b5011(h011);

       k4011(h011)=((py1(2)-mat_prep(2,i+1)))/((px1(2)-mat_prep(1,i+1)));
       b4011(h011)=mat_prep(2,i+1)-(k4011(h011)*mat_prep(1,i+1));
       y_prep_rec_p4011(h011)=k4011(h011)*mat_prep(1,i)+b4011(h011);

    %donja tocka
       h012=h012+1;
```

```
        k5012(h012)=((py2(2)-mat_prep(2,i)))/((px2(2)-mat_prep(1,i)));
        b5012(h012)=mat_prep(2,i)-(k5012(h012)*mat_prep(1,i));
        y_prep_rec_p5012(h012)=k5012(h012)*mat_prep(1,i+1)+b5012(h012);

        k4012(h012)=((py2(2)-mat_prep(2,i+1)))/((px2(2)-mat_prep(1,i+1)));
        b4012(h012)=mat_prep(2,i+1)-(k4012(h012)*mat_prep(1,i+1));
        y_prep_rec_p4012(h012)=k4012(h012)*mat_prep(1,i)+b4012(h012);


     %uvjeti
           if (px1(2)>px2(2) & mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)<y_prep_rec_p4012(h012) & mat_prep(2,i+1)<
y_prep_rec_p5012(h012))|(px1(2)>px2(2) & mat_prep(1,i+1)<px2(2)&
mat_prep(2,i)>y_prep_rec_p4012(h012) & mat_prep(2,i+1)<
y_prep_rec_p5012(h012))|(px1(2)<px2(2) &
mat_prep(1,i)<px1(2)&mat_prep(2,i+1)<y_prep_rec_p5012(h012))

              disp('Prepreka sijeće oba pravca.');
             prep_sijece_oba=prep_sijece_oba+1;

           elseif (px1(2)>px2(2)& mat_prep(1,i+1)<px1(2)&
mat_prep(2,i)>y_prep_rec_p4012(h012)& mat_prep(2,i)>y_prep_rec_p4011(h011)&
mat_prep(1,i)<px2(2)& mat_prep(2,i+1)>y_prep_rec_p5012(h012))| (px1(2)>px2(2)&
mat_prep(1,i+1)<px1(2)& mat_prep(2,i)>y_prep_rec_p4012(h012)&
mat_prep(2,i)>y_prep_rec_p4011(h011)& mat_prep(1,i)>px2(2)&
mat_prep(2,i+1)<y_prep_rec_p5012(h012))|(px1(2)>px2(2)& mat_prep(1,i+1)>px1(2)&
mat_prep(2,i)>y_prep_rec_p4012(h012)& mat_prep(2,i)<y_prep_rec_p4011(h011)&
mat_prep(1,i)<px2(2)& mat_prep(2,i+1)>y_prep_rec_p5012(h012))|(px1(2)>px2(2)&
mat_prep(1,i+1)>px1(2)& mat_prep(2,i)>y_prep_rec_p4012(h012)&
mat_prep(2,i)<y_prep_rec_p4011(h011)& mat_prep(1,i)>px2(2)&
mat_prep(2,i+1)<y_prep_rec_p5012(h012)) |(px1(2)<px2(2) &
mat_prep(2,i)<y_prep_rec_p4011(h011)& mat_prep(2,i+1)>y_prep_rec_p5012(h012))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;

           elseif (px1(2)<px2(2)& mat_prep(1,i+1)<px2(2)&
mat_prep(2,i)>y_prep_rec_p4011(h011)& mat_prep(2,i)>y_prep_rec_p4012(h012)&
mat_prep(2,i+1)>y_prep_rec_p5011(h011) &
mat_prep(2,i+1)<y_prep_rec_p5012(h012))|(px1(2)<px2(2)& mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)>y_prep_rec_p4011(h011)& mat_prep(2,i)<y_prep_rec_p4012(h012)&
mat_prep(2,i+1)>y_prep_rec_p5011(h011) & mat_prep(2,i+1)<y_prep_rec_p5012(h012))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

           else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
           end


     %-----------------------------------------------
     %oba pravca, B je lijevo od pravca, A je desno od pravca

      elseif mat_prep(2,i+1)>y_prep_poc_rec_p1(1,g)

   %gornja tocka
```

```
       h021=h021+1;
       k5021(h021)=((py1(2)-mat_prep(2,i))+0.0000001)/((px1(2)-
mat_prep(1,i))+0.0000001);
       b5021(h021)=mat_prep(2,i)-(k5021(h021)*mat_prep(1,i));
       y_prep_rec_p5021(h021)=k5021(h021)*mat_prep(1,i+1)+b5021(h021);

       k4021(h021)=((py1(2)-mat_prep(2,i+1))+0.0000001)/((px1(2)-
mat_prep(1,i+1))+0.0000001);
       b4021(h021)=mat_prep(2,i+1)-(k4021(h021)*mat_prep(1,i+1));
       y_prep_rec_p4021(h021)=k4021(h021)*mat_prep(1,i)+b4021(h021);

    %donja tocka
       h022=h022+1;
       k5022(h022)=((py2(2)-mat_prep(2,i))+0.0000001)/((px2(2)-
mat_prep(1,i))+0.0000001);
       b5022(h022)=mat_prep(2,i)-(k5022(h022)*mat_prep(1,i));
       y_prep_rec_p5022(h022)=k5022(h022)*mat_prep(1,i+1)+b5022(h022);

       k4022(h022)=((py2(2)-mat_prep(2,i+1))+0.0000001)/((px2(2)-
mat_prep(1,i+1))+0.0000001);
       b4022(h022)=mat_prep(2,i+1)-(k4022(h022)*mat_prep(1,i+1));
       y_prep_rec_p4022(h022)=k4022(h022)*mat_prep(1,i)+b4022(h022);

     %uvjeti
           if (px1(2)>px2(2) & mat_prep(1,i)<px2(2)&
mat_prep(2,i+1)>y_prep_rec_p5021(h021))|(px1(2)<px2(2) &
mat_prep(1,i+1)<px1(2))|(px1(2)<px2(2) & mat_prep(1,i+1)>px1(2)&
mat_prep(2,i)>y_prep_rec_p4021(h021))
             disp('Prepreka sijeće oba pravca.');
             prep_sijece_oba=prep_sijece_oba+1;

           elseif (px1(2)>px2(2) & mat_prep(1,i)<px2(2)&
mat_prep(2,i+1)<y_prep_rec_p5022(h022)& mat_prep(2,i+1)>y_prep_rec_p5021(h021)&
mat_prep(2,i)<y_prep_rec_p4022(h022))|(px1(2)>px2(2) & mat_prep(1,i)>px2(2)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022)& mat_prep(2,i)<y_prep_rec_p4022(h022)&
((mat_prep(1,i)<px1(2)& mat_prep(2,i+1)>y_prep_rec_p5021(h021))
|(mat_prep(1,i)>px1(2)& mat_prep(2,i+1)<y_prep_rec_p5021(h021))))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;

           elseif (px1(2)>px2(2) & mat_prep(2,i)>y_prep_rec_p4022(h022)&
mat_prep(2,i+1)<y_prep_rec_p5021(h021)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022))|(px1(2)<px2(2)& mat_prep(1,i)<px1(2)&
mat_prep(2,i+1)<y_prep_rec_p5021(h021)& mat_prep(2,i+1)>y_prep_rec_p5022(h022)&
mat_prep(1,i+1)<px2(2)& mat_prep(2,i)<y_prep_rec_p4022(h022))|(px1(2)<px2(2)&
mat_prep(1,i)<px1(2)& mat_prep(2,i+1)<y_prep_rec_p5021(h021)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022)& mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)>y_prep_rec_p4022(h022))|(px1(2)<px2(2)& mat_prep(1,i)>px1(2)&
mat_prep(2,i+1)>y_prep_rec_p5021(h021)& mat_prep(2,i+1)>y_prep_rec_p5022(h022)&
mat_prep(1,i+1)<px2(2)& mat_prep(2,i)<y_prep_rec_p4022(h022))|(px1(2)<px2(2)&
mat_prep(1,i)>px1(2)& mat_prep(2,i+1)>y_prep_rec_p5021(h021)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022)& mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)>y_prep_rec_p4022(h022))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

           else
```

```
            disp('Prepreka se NE nalazi unutar prostora!');
            prep_vani=prep_vani+1;
          end
        end

%-------------------------------------------
%lijevi pravac, A je lijevo od pravca, B je desno od pravca

     elseif mat_prep(2,i)>=y_prep_poc_rec_p1(1,g) &
mat_prep(2,i+1)<=y_prep_kraj_rec_p1(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g))& px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))

       h11=h11+1;
       k511(h11)=((py1(2)-mat_prep(2,i)))/((px1(2)-mat_prep(1,i))+000000001);
       b511(h11)=mat_prep(2,i)-(k511(h11)*mat_prep(1,i));
       y_prep_rec_p511(h11)=k511(h11)*mat_prep(1,i+1)+b511(h11);

       k411(h11)=((py1(2)-mat_prep(2,i+1)))/((px1(2)-mat_prep(1,i+1)));
       b411(h11)=mat_prep(2,i+1)-(k411(h11)*mat_prep(1,i+1));
       y_prep_rec_p411(h11)=k411(h11)*mat_prep(1,i)+b411(h11);

            if (mat_prep(1,i+1)>px1(2)& mat_prep(2,i)<y_prep_rec_p411(h11)&
mat_prep(2,i+1)<y_prep_rec_p511(h11)) | (mat_prep(1,i+1)<px1(2) &
mat_prep(2,i)>y_prep_rec_p411(h11)& mat_prep(2,i+1)<y_prep_rec_p511(h11))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;

            elseif ((mat_prep(1,i)>px1(2)& mat_prep(2,i)>=y_prep_rec_p411(h11)&
mat_prep(2,i+1)<=y_prep_rec_p511(h11)) | (mat_prep(1,i)<=px1(2) &
mat_prep(2,i)>=y_prep_rec_p411(h11)& mat_prep(2,i+1)>=y_prep_rec_p511(h11))) &
(mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)& px1(2)<px2(2))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;

            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;

            end

%-------------------------------------------
%lijevi pravac, B je lijevo od pravca, A je desno od pravca

     elseif mat_prep(2,i+1)>=y_prep_kraj_rec_p1(1,g)&
mat_prep(2,i)<=y_prep_poc_rec_p1(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) & px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))
```

```
       h12=h12+1;
       k512(h12)=((py1(2)-mat_prep(2,i))+0.0000001)/((px1(2)-
mat_prep(1,i))+0.0000001);
       b512(h12)=mat_prep(2,i)-(k512(h12)*mat_prep(1,i));
       y_prep_rec_p512(h12)=k512(h12)*mat_prep(1,i+1)+b512(h12);

       k412(h12)=((py1(2)-mat_prep(2,i+1))+0.0000001)/((px1(2)-
mat_prep(1,i+1))+0.0000001);
       b412(h12)=mat_prep(2,i+1)-(k412(h12)*mat_prep(1,i+1));
       y_prep_rec_p412(h12)=k412(h12)*mat_prep(1,i)+b412(h12);

            if ((mat_prep(1,i+1)<px1(2) & mat_prep(2,i)<y_prep_rec_p412(h12)&
mat_prep(2,i+1)>y_prep_rec_p512(h12)) | (mat_prep(2,i)>y_prep_rec_p412(h12)&
mat_prep(2,i+1)>y_prep_rec_p512(h12))) &
(((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)<y_prep_poc_rec_p3(1,g) )&
px1(2)<px2(2))
|((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)>y_prep_poc_rec_p3(1,g)) &
px1(2)>px2(2)))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;
            elseif ((mat_prep(1,i)<px1(2) & mat_prep(2,i)<=y_prep_rec_p412(h12)&
mat_prep(2,i+1)<=y_prep_rec_p512(h12)) | (mat_prep(2,i)<=y_prep_rec_p412(h12)&
mat_prep(2,i+1)>=y_prep_rec_p512(h12))) & ((mat_prep(2,i)<y_prep_poc_rec_p3(1,g) &
px1(2)<px2(2)) |(mat_prep(2,i)>y_prep_poc_rec_p3(1,g) & px1(2)>px2(2)))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;
            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
            end

%-------------------------------------------
%desni pravac, A je lijevo od pravca, B je desno od pravca

  elseif mat_prep(2,i)>=y_prep_poc_rec_p2(1,g) &
mat_prep(2,i+1)<=y_prep_kraj_rec_p2(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) & px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))

       h21=h21+1;
       k521(h21)=((py2(2)-mat_prep(2,i))+0.0000001)/((px2(2)-
mat_prep(1,i))+0.0000001);
       b521(h21)=mat_prep(2,i)-(k521(h21)*mat_prep(1,i));
       y_prep_rec_p521(h21)=k521(h21)*mat_prep(1,i+1)+b521(h21);

       k421(h21)=((py2(2)-mat_prep(2,i+1))+0.0000001)/((px2(2)-
mat_prep(1,i+1))+0.0000001);
       b421(h21)=mat_prep(2,i+1)-(k421(h21)*mat_prep(1,i+1));
       y_prep_rec_p421(h21)=k421(h21)*mat_prep(1,i)+b421(h21);
```

```
            if (mat_prep(1,i+1)>px2(2)& mat_prep(2,i)<y_prep_rec_p421(h21)&
mat_prep(2,i+1)<y_prep_rec_p521(h21)) | (mat_prep(1,i+1)<px2(2) &
mat_prep(2,i)>y_prep_rec_p421(h21)& mat_prep(2,i+1)<y_prep_rec_p521(h21))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

            elseif ((mat_prep(1,i)>px2(2)& mat_prep(2,i)>=y_prep_rec_p421(h21)&
mat_prep(2,i+1)<=y_prep_rec_p521(h21)) | (mat_prep(1,i)<px2(2) &
mat_prep(2,i)>=y_prep_rec_p421(h21)& mat_prep(2,i+1)>=y_prep_rec_p521(h21))) &
((mat_prep(2,i)<y_prep_poc_rec_p3(1,g) & px1(2)<px2(2))
|(mat_prep(2,i)>y_prep_poc_rec_p3(1,g) & px1(2)>px2(2)))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;

            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
            end


%-----------------------------------------------
%desni pravac, B je lijevo od pravca, A je desno od pravca

     elseif mat_prep(2,i+1)>=y_prep_kraj_rec_p2(1,g) &
mat_prep(2,i)<=y_prep_poc_rec_p2(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) & px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))

       h22=h22+1;
       k522(h22)=((py2(2)-mat_prep(2,i))+0.0000001)/((px2(2)-
mat_prep(1,i))+0.0000001);
       b522(h22)=mat_prep(2,i)-(k522(h22)*mat_prep(1,i));
       y_prep_rec_p522(h22)=k522(h22)*mat_prep(1,i+1)+b522(h22);

       k422(h22)=((py2(2)-mat_prep(2,i+1))+0.0000001)/((px2(2)-
mat_prep(1,i+1))+0.0000001);
       b422(h22)=mat_prep(2,i+1)-(k422(h22)*mat_prep(1,i+1));
       y_prep_rec_p422(h22)=k422(h22)*mat_prep(1,i)+b422(h22);

            if ((mat_prep(1,i+1)<px2(2) & mat_prep(2,i)<y_prep_rec_p422(h22)&
mat_prep(2,i+1)>y_prep_rec_p522(h22)) | (mat_prep(2,i)>y_prep_rec_p422(h22)&
mat_prep(2,i+1)>y_prep_rec_p522(h22))) & ((mat_prep(2,i)<y_prep_poc_rec_p3(1,g) &
px1(2)<px2(2)) |(mat_prep(2,i)>y_prep_poc_rec_p3(1,g) & px1(2)>px2(2)))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

            elseif ((mat_prep(1,i)<px2(2) & mat_prep(2,i)<=y_prep_rec_p422(h22)&
mat_prep(2,i+1)<=y_prep_rec_p522(h22)) | (mat_prep(2,i)<=y_prep_rec_p422(h22)&
mat_prep(2,i+1)>=y_prep_rec_p522(h22))) &
((((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) &
```

```
px1(2)<px2(2))
|((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2))))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;

            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
            end

%---------------------------------------------
%prepreka se nalazi unutar područja
    elseif (((mat_prep(2,i)<y_prep_poc_rec_p1(1,g)) &
(mat_prep(2,i)>y_prep_poc_rec_p2(1,g)))  &
((mat_prep(2,i+1)<y_prep_kraj_rec_p1(1,g)) &
(mat_prep(2,i+1)>y_prep_kraj_rec_p2(1,g)))) &
(((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) &
px1(2)<px2(2))  | ((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) & px1(2)>px2(2)))
         disp('Prepreka je unutar područja.');
         prep_unutar=prep_unutar+1;
         mat_prep_unutar(1,h)=mat_prep(1,i);
         mat_prep_unutar(2,h)=mat_prep(2,i);
         mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
         mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
         h=h+2;

%---------------------------------------------
%prepreka se nalazi izvan područja

    else
         disp('Prepreka se NE nalazi unutar prostora!');
         prep_vani=prep_vani+1;
    end

end
```

```
%ROBOT 2

%funkcija koja ispituje da li se prepreka nalazi unurat, izvan ili u radnom
%prostoru robota

function
[prep_sijece_oba,prep_sijece_gore,prep_sijece_dole,prep_vani,prep_unutar,mat_prep_u
nutar] = f(toc_1_poc,toc_1_kraj,xpr,ypr,br_prepreka,duz,sir)

%-------------------------------------------
%zadavanje početnih uvjeta
a=0; e=0; f=0; g=0; h=1;
prep_sijece_oba=0;
prep_sijece_dole=0;
prep_sijece_gore=0;
prep_unutar=0;
prep_vani=0;
mat_prep_unutar=0;
%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće robot

px1=[0 toc_1_poc(1)];       %x koord početne i krajnje točke pravca p1
py1=[(sir/2) toc_1_poc(2)]; %y koord početne i krajnje točke pravca p1

px2=[0 toc_1_kraj(1)];       %x koord početne i krajnje točke pravca p2
py2=[(sir/2) toc_1_kraj(2)]; %y koord početne i krajnje točke pravca p2

px3=[px1(2) px2(2)];
py3=[py1(2) py2(2)];

%-------------------------------------------
%definiranje prepreka u matričnom obliku

mat_pr=[xpr;ypr];

%-------------------------------------------
%crtanje radnog prosrtora

figure(7)
title 'RADNI PROSTOR PRVOG ROBOTA S PREPREKAMA'
axis equal
hold on
axis([0 duz 0 sir])

line(px1,py1,'Color','b','LineWidth',1);  %pravac p1
line(px2,py2,'Color','b','LineWidth',1);  %pravac p2
line(px3,py3,'Color','c','LineWidth',1);  %pravac p3

%-------------------------------------------
%definiranje prve i druge točke prepreke

[c,d]=size(mat_pr);
```

```
for i=1:2:d
    a=a+2;
      if
(mat_pr(1,i)>mat_pr(1,a))|((mat_pr(1,i)==mat_pr(1,a))&(mat_pr(2,i)<mat_pr(2,a)))
         mat_prep(:,i)=mat_pr(:,a);
         mat_prep(:,a)=mat_pr(:,i);
        else
         mat_prep(:,i)=mat_pr(:,i);
         mat_prep(:,a)=mat_pr(:,a);
      end
end

mat_prep;

%---------------------------------------------
%crtanje prepreka

for i=1:2:d
    prx(1,1)=mat_prep(1,i);
    pry(1,1)=mat_prep(2,i);

    prx(1,2)=mat_prep(1,i+1);
    pry(1,2)=mat_prep(2,i+1);

line(prx,pry,'Color','k','LineWidth',1);
end
hold off

%---------------------------------------------
%k - koeficijenti smijera -> k=tg?

k1=((py1(2)-py1(1))+0.0000001)/((px1(2)-px1(1))+0.0000001);
k2=((py2(2)-py2(1))+0.0000001)/((px2(2)-px2(1))+0.0000001);
k3=((py1(2)-py2(2))+0.0000001)/((px1(2)-px2(2))+0.0000001);

%---------------------------------------------
%b - odsječci na y osi

b1=py1(2)-k1*px1(2);
b2=py2(2)-k2*px2(2);
b3=py1(2)-k3*px1(2);

%---------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:2:d
    e=e+1;
   y_prep_poc_rec_p1(e)=k1*mat_prep(1,i)+b1;
   y_prep_poc_rec_p2(e)=k2*mat_prep(1,i)+b2;
   y_prep_poc_rec_p3(e)=k3*mat_prep(1,i)+b3;
end

for i=2:2:d
    f=f+1;
   y_prep_kraj_rec_p1(f)=k1*mat_prep(1,i)+b1;
```

```
   y_prep_kraj_rec_p2(f)=k2*mat_prep(1,i)+b2;
   y_prep_kraj_rec_p3(f)=k3*mat_prep(1,i)+b3;
end

h01=0;
h02=0;


h011=0;
h012=0;
h021=0;
h022=0;
h11=0;
h12=0;
h21=0;
h22=0;

%---------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:2:d
     g=g+1;

%---------------------------------------------
%prepreka sijeće oba pravca

     if (mat_prep(2,i)>y_prep_poc_rec_p1(1,g) &
mat_prep(2,i+1)<y_prep_kraj_rec_p2(1,g))|(mat_prep(2,i+1)>y_prep_kraj_rec_p1(1,g) &
mat_prep(2,i)<y_prep_poc_rec_p2(1,g)& mat_prep(1,i+1)>mat_prep(1,i))


      if (mat_prep(2,i)< y_prep_poc_rec_p3(1,g) & mat_prep(2,i+1)<
y_prep_kraj_rec_p3(1,g)& px1(2)<px2(2))|(mat_prep(2,i)> y_prep_poc_rec_p3(1,g) &
mat_prep(2,i+1)> y_prep_kraj_rec_p3(1,g)& px1(2)>px2(2))|(mat_prep(1,i)<px1(2)&
mat_prep(1,i+1)<px1(2)&px1(2)==px2(2))
             disp('Prepreka sijeće oba pravca.');
             prep_sijece_oba=prep_sijece_oba+1;

     %---------------------------------------------
     %oba pravca, A je lijevo od pravca, B je desno od pravca

      elseif mat_prep(2,i)>y_prep_poc_rec_p1(1,g)

     %gornja tocka
       h011=h011+1;
       k5011(h011)=((py1(2)-mat_prep(2,i)))/((px1(2)-mat_prep(1,i)));
       b5011(h011)=mat_prep(2,i)-(k5011(h011)*mat_prep(1,i));
       y_prep_rec_p5011(h011)=k5011(h011)*mat_prep(1,i+1)+b5011(h011);

       k4011(h011)=((py1(2)-mat_prep(2,i+1)))/((px1(2)-mat_prep(1,i+1)));
       b4011(h011)=mat_prep(2,i+1)-(k4011(h011)*mat_prep(1,i+1));
       y_prep_rec_p4011(h011)=k4011(h011)*mat_prep(1,i)+b4011(h011);

    %donja tocka
       h012=h012+1;
```

```
        k5012(h012)=((py2(2)-mat_prep(2,i)))/((px2(2)-mat_prep(1,i)));
        b5012(h012)=mat_prep(2,i)-(k5012(h012)*mat_prep(1,i));
        y_prep_rec_p5012(h012)=k5012(h012)*mat_prep(1,i+1)+b5012(h012);

        k4012(h012)=((py2(2)-mat_prep(2,i+1)))/((px2(2)-mat_prep(1,i+1)));
        b4012(h012)=mat_prep(2,i+1)-(k4012(h012)*mat_prep(1,i+1));
        y_prep_rec_p4012(h012)=k4012(h012)*mat_prep(1,i)+b4012(h012);


     %uvjeti
           if (px1(2)>px2(2) & mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)<y_prep_rec_p4012(h012) & mat_prep(2,i+1)<
y_prep_rec_p5012(h012))|(px1(2)>px2(2) & mat_prep(1,i+1)<px2(2)&
mat_prep(2,i)>y_prep_rec_p4012(h012) & mat_prep(2,i+1)<
y_prep_rec_p5012(h012))|(px1(2)<px2(2) &
mat_prep(1,i)<px1(2)&mat_prep(2,i+1)<y_prep_rec_p5012(h012))

              disp('Prepreka sijeće oba pravca.');
             prep_sijece_oba=prep_sijece_oba+1;

           elseif (px1(2)>px2(2)& mat_prep(1,i+1)<px1(2)&
mat_prep(2,i)>y_prep_rec_p4012(h012)& mat_prep(2,i)>y_prep_rec_p4011(h011)&
mat_prep(1,i)<px2(2)& mat_prep(2,i+1)>y_prep_rec_p5012(h012))| (px1(2)>px2(2)&
mat_prep(1,i+1)<px1(2)& mat_prep(2,i)>y_prep_rec_p4012(h012)&
mat_prep(2,i)>y_prep_rec_p4011(h011)& mat_prep(1,i)>px2(2)&
mat_prep(2,i+1)<y_prep_rec_p5012(h012))|(px1(2)>px2(2)& mat_prep(1,i+1)>px1(2)&
mat_prep(2,i)>y_prep_rec_p4012(h012)& mat_prep(2,i)<y_prep_rec_p4011(h011)&
mat_prep(1,i)<px2(2)& mat_prep(2,i+1)>y_prep_rec_p5012(h012))|(px1(2)>px2(2)&
mat_prep(1,i+1)>px1(2)& mat_prep(2,i)>y_prep_rec_p4012(h012)&
mat_prep(2,i)<y_prep_rec_p4011(h011)& mat_prep(1,i)>px2(2)&
mat_prep(2,i+1)<y_prep_rec_p5012(h012)) |(px1(2)<px2(2) &
mat_prep(2,i)<y_prep_rec_p4011(h011)& mat_prep(2,i+1)>y_prep_rec_p5012(h012))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;

           elseif (px1(2)<px2(2)& mat_prep(1,i+1)<px2(2)&
mat_prep(2,i)>y_prep_rec_p4011(h011)& mat_prep(2,i)>y_prep_rec_p4012(h012)&
mat_prep(2,i+1)>y_prep_rec_p5011(h011) &
mat_prep(2,i+1)<y_prep_rec_p5012(h012))|(px1(2)<px2(2)& mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)>y_prep_rec_p4011(h011)& mat_prep(2,i)<y_prep_rec_p4012(h012)&
mat_prep(2,i+1)>y_prep_rec_p5011(h011) & mat_prep(2,i+1)<y_prep_rec_p5012(h012))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

           else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
           end


     %-----------------------------------------------
     %oba pravca, B je lijevo od pravca, A je desno od pravca

      elseif mat_prep(2,i+1)>y_prep_poc_rec_p1(1,g)

   %gornja tocka
```

```
       h021=h021+1;
       k5021(h021)=((py1(2)-mat_prep(2,i))+0.0000001)/((px1(2)-
mat_prep(1,i))+0.0000001);
       b5021(h021)=mat_prep(2,i)-(k5021(h021)*mat_prep(1,i));
       y_prep_rec_p5021(h021)=k5021(h021)*mat_prep(1,i+1)+b5021(h021);

       k4021(h021)=((py1(2)-mat_prep(2,i+1))+0.0000001)/((px1(2)-
mat_prep(1,i+1))+0.0000001);
       b4021(h021)=mat_prep(2,i+1)-(k4021(h021)*mat_prep(1,i+1));
       y_prep_rec_p4021(h021)=k4021(h021)*mat_prep(1,i)+b4021(h021);

    %donja tocka
       h022=h022+1;
       k5022(h022)=((py2(2)-mat_prep(2,i))+0.0000001)/((px2(2)-
mat_prep(1,i))+0.0000001);
       b5022(h022)=mat_prep(2,i)-(k5022(h022)*mat_prep(1,i));
       y_prep_rec_p5022(h022)=k5022(h022)*mat_prep(1,i+1)+b5022(h022);

       k4022(h022)=((py2(2)-mat_prep(2,i+1))+0.0000001)/((px2(2)-
mat_prep(1,i+1))+0.0000001);
       b4022(h022)=mat_prep(2,i+1)-(k4022(h022)*mat_prep(1,i+1));
       y_prep_rec_p4022(h022)=k4022(h022)*mat_prep(1,i)+b4022(h022);

     %uvjeti
           if (px1(2)>px2(2) & mat_prep(1,i)<px2(2)&
mat_prep(2,i+1)>y_prep_rec_p5021(h021))|(px1(2)<px2(2) &
mat_prep(1,i+1)<px1(2))|(px1(2)<px2(2) & mat_prep(1,i+1)>px1(2)&
mat_prep(2,i)>y_prep_rec_p4021(h021))
             disp('Prepreka sijeće oba pravca.');
             prep_sijece_oba=prep_sijece_oba+1;

           elseif (px1(2)>px2(2) & mat_prep(1,i)<px2(2)&
mat_prep(2,i+1)<y_prep_rec_p5022(h022)& mat_prep(2,i+1)>y_prep_rec_p5021(h021)&
mat_prep(2,i)<y_prep_rec_p4022(h022))|(px1(2)>px2(2) & mat_prep(1,i)>px2(2)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022)& mat_prep(2,i)<y_prep_rec_p4022(h022)&
((mat_prep(1,i)<px1(2)& mat_prep(2,i+1)>y_prep_rec_p5021(h021))
|(mat_prep(1,i)>px1(2)& mat_prep(2,i+1)<y_prep_rec_p5021(h021))))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;

           elseif (px1(2)>px2(2) & mat_prep(2,i)>y_prep_rec_p4022(h022)&
mat_prep(2,i+1)<y_prep_rec_p5021(h021)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022))|(px1(2)<px2(2)& mat_prep(1,i)<px1(2)&
mat_prep(2,i+1)<y_prep_rec_p5021(h021)& mat_prep(2,i+1)>y_prep_rec_p5022(h022)&
mat_prep(1,i+1)<px2(2)& mat_prep(2,i)<y_prep_rec_p4022(h022))|(px1(2)<px2(2)&
mat_prep(1,i)<px1(2)& mat_prep(2,i+1)<y_prep_rec_p5021(h021)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022)& mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)>y_prep_rec_p4022(h022))|(px1(2)<px2(2)& mat_prep(1,i)>px1(2)&
mat_prep(2,i+1)>y_prep_rec_p5021(h021)& mat_prep(2,i+1)>y_prep_rec_p5022(h022)&
mat_prep(1,i+1)<px2(2)& mat_prep(2,i)<y_prep_rec_p4022(h022))|(px1(2)<px2(2)&
mat_prep(1,i)>px1(2)& mat_prep(2,i+1)>y_prep_rec_p5021(h021)&
mat_prep(2,i+1)>y_prep_rec_p5022(h022)& mat_prep(1,i+1)>px2(2)&
mat_prep(2,i)>y_prep_rec_p4022(h022))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

           else
```

```
            disp('Prepreka se NE nalazi unutar prostora!');
            prep_vani=prep_vani+1;
          end
        end

%-------------------------------------------
%lijevi pravac, A je lijevo od pravca, B je desno od pravca

     elseif mat_prep(2,i)>=y_prep_poc_rec_p1(1,g) &
mat_prep(2,i+1)<=y_prep_kraj_rec_p1(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g))& px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))

       h11=h11+1;
       k511(h11)=((py1(2)-mat_prep(2,i)))/((px1(2)-mat_prep(1,i))+000000001);
       b511(h11)=mat_prep(2,i)-(k511(h11)*mat_prep(1,i));
       y_prep_rec_p511(h11)=k511(h11)*mat_prep(1,i+1)+b511(h11);

       k411(h11)=((py1(2)-mat_prep(2,i+1)))/((px1(2)-mat_prep(1,i+1)));
       b411(h11)=mat_prep(2,i+1)-(k411(h11)*mat_prep(1,i+1));
       y_prep_rec_p411(h11)=k411(h11)*mat_prep(1,i)+b411(h11);

            if (mat_prep(1,i+1)>px1(2)& mat_prep(2,i)<y_prep_rec_p411(h11)&
mat_prep(2,i+1)<y_prep_rec_p511(h11)) | (mat_prep(1,i+1)<px1(2) &
mat_prep(2,i)>y_prep_rec_p411(h11)& mat_prep(2,i+1)<y_prep_rec_p511(h11))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;

            elseif ((mat_prep(1,i)>px1(2)& mat_prep(2,i)>=y_prep_rec_p411(h11)&
mat_prep(2,i+1)<=y_prep_rec_p511(h11)) | (mat_prep(1,i)<=px1(2) &
mat_prep(2,i)>=y_prep_rec_p411(h11)& mat_prep(2,i+1)>=y_prep_rec_p511(h11))) &
(mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)& px1(2)<px2(2))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;

            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;

            end

%-------------------------------------------
%lijevi pravac, B je lijevo od pravca, A je desno od pravca

     elseif mat_prep(2,i+1)>=y_prep_kraj_rec_p1(1,g)&
mat_prep(2,i)<=y_prep_poc_rec_p1(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) & px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))
```

```
       h12=h12+1;
       k512(h12)=((py1(2)-mat_prep(2,i))+0.0000001)/((px1(2)-
mat_prep(1,i))+0.0000001);
       b512(h12)=mat_prep(2,i)-(k512(h12)*mat_prep(1,i));
       y_prep_rec_p512(h12)=k512(h12)*mat_prep(1,i+1)+b512(h12);

       k412(h12)=((py1(2)-mat_prep(2,i+1))+0.0000001)/((px1(2)-
mat_prep(1,i+1))+0.0000001);
       b412(h12)=mat_prep(2,i+1)-(k412(h12)*mat_prep(1,i+1));
       y_prep_rec_p412(h12)=k412(h12)*mat_prep(1,i)+b412(h12);

            if ((mat_prep(1,i+1)<px1(2) & mat_prep(2,i)<y_prep_rec_p412(h12)&
mat_prep(2,i+1)>y_prep_rec_p512(h12)) | (mat_prep(2,i)>y_prep_rec_p412(h12)&
mat_prep(2,i+1)>y_prep_rec_p512(h12))) &
(((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)<y_prep_poc_rec_p3(1,g) )&
px1(2)<px2(2))
|((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)>y_prep_poc_rec_p3(1,g)) &
px1(2)>px2(2)))
             disp('Prepreka sijeće gornji pravac.');
             prep_sijece_gore=prep_sijece_gore+1;
            elseif ((mat_prep(1,i)<px1(2) & mat_prep(2,i)<=y_prep_rec_p412(h12)&
mat_prep(2,i+1)<=y_prep_rec_p512(h12)) | (mat_prep(2,i)<=y_prep_rec_p412(h12)&
mat_prep(2,i+1)>=y_prep_rec_p512(h12))) & ((mat_prep(2,i)<y_prep_poc_rec_p3(1,g) &
px1(2)<px2(2)) |(mat_prep(2,i)>y_prep_poc_rec_p3(1,g) & px1(2)>px2(2)))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;
            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
            end

%-------------------------------------------
%desni pravac, A je lijevo od pravca, B je desno od pravca

  elseif mat_prep(2,i)>=y_prep_poc_rec_p2(1,g) &
mat_prep(2,i+1)<=y_prep_kraj_rec_p2(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) & px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))

       h21=h21+1;
       k521(h21)=((py2(2)-mat_prep(2,i))+0.0000001)/((px2(2)-
mat_prep(1,i))+0.0000001);
       b521(h21)=mat_prep(2,i)-(k521(h21)*mat_prep(1,i));
       y_prep_rec_p521(h21)=k521(h21)*mat_prep(1,i+1)+b521(h21);

       k421(h21)=((py2(2)-mat_prep(2,i+1))+0.0000001)/((px2(2)-
mat_prep(1,i+1))+0.0000001);
       b421(h21)=mat_prep(2,i+1)-(k421(h21)*mat_prep(1,i+1));
       y_prep_rec_p421(h21)=k421(h21)*mat_prep(1,i)+b421(h21);
```

```
            if (mat_prep(1,i+1)>px2(2)& mat_prep(2,i)<y_prep_rec_p421(h21)&
mat_prep(2,i+1)<y_prep_rec_p521(h21)) | (mat_prep(1,i+1)<px2(2) &
mat_prep(2,i)>y_prep_rec_p421(h21)& mat_prep(2,i+1)<y_prep_rec_p521(h21))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

            elseif ((mat_prep(1,i)>px2(2)& mat_prep(2,i)>=y_prep_rec_p421(h21)&
mat_prep(2,i+1)<=y_prep_rec_p521(h21)) | (mat_prep(1,i)<px2(2) &
mat_prep(2,i)>=y_prep_rec_p421(h21)& mat_prep(2,i+1)>=y_prep_rec_p521(h21))) &
((mat_prep(2,i)<y_prep_poc_rec_p3(1,g) & px1(2)<px2(2))
|(mat_prep(2,i)>y_prep_poc_rec_p3(1,g) & px1(2)>px2(2)))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;

            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
            end


%-----------------------------------------------
%desni pravac, B je lijevo od pravca, A je desno od pravca

     elseif mat_prep(2,i+1)>=y_prep_kraj_rec_p2(1,g) &
mat_prep(2,i)<=y_prep_poc_rec_p2(1,g) & (((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) & px1(2)<px2(2))  |
((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2)))

       h22=h22+1;
       k522(h22)=((py2(2)-mat_prep(2,i))+0.0000001)/((px2(2)-
mat_prep(1,i))+0.0000001);
       b522(h22)=mat_prep(2,i)-(k522(h22)*mat_prep(1,i));
       y_prep_rec_p522(h22)=k522(h22)*mat_prep(1,i+1)+b522(h22);

       k422(h22)=((py2(2)-mat_prep(2,i+1))+0.0000001)/((px2(2)-
mat_prep(1,i+1))+0.0000001);
       b422(h22)=mat_prep(2,i+1)-(k422(h22)*mat_prep(1,i+1));
       y_prep_rec_p422(h22)=k422(h22)*mat_prep(1,i)+b422(h22);

            if ((mat_prep(1,i+1)<px2(2) & mat_prep(2,i)<y_prep_rec_p422(h22)&
mat_prep(2,i+1)>y_prep_rec_p522(h22)) | (mat_prep(2,i)>y_prep_rec_p422(h22)&
mat_prep(2,i+1)>y_prep_rec_p522(h22))) & ((mat_prep(2,i)<y_prep_poc_rec_p3(1,g) &
px1(2)<px2(2)) |(mat_prep(2,i)>y_prep_poc_rec_p3(1,g) & px1(2)>px2(2)))
             disp('Prepreka sijeće donji pravac.');
             prep_sijece_dole=prep_sijece_dole+1;

            elseif ((mat_prep(1,i)<px2(2) & mat_prep(2,i)<=y_prep_rec_p422(h22)&
mat_prep(2,i+1)<=y_prep_rec_p522(h22)) | (mat_prep(2,i)<=y_prep_rec_p422(h22)&
mat_prep(2,i+1)>=y_prep_rec_p522(h22))) &
((((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) &
```

```
px1(2)<px2(2))
|((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)|mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) &
px1(2)>px2(2))))
             disp('Prepreka je unutar područja.');
             prep_unutar=prep_unutar+1;
             mat_prep_unutar(1,h)=mat_prep(1,i);
             mat_prep_unutar(2,h)=mat_prep(2,i);
             mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
             mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
             h=h+2;

            else
             disp('Prepreka se NE nalazi unutar prostora!');
             prep_vani=prep_vani+1;
            end

%---------------------------------------------
%prepreka se nalazi unutar područja
    elseif (((mat_prep(2,i)<y_prep_poc_rec_p1(1,g)) &
(mat_prep(2,i)>y_prep_poc_rec_p2(1,g)))  &
((mat_prep(2,i+1)<y_prep_kraj_rec_p1(1,g)) &
(mat_prep(2,i+1)>y_prep_kraj_rec_p2(1,g)))) &
(((mat_prep(2,i)<y_prep_poc_rec_p3(1,g)| mat_prep(2,i+1)<y_prep_kraj_rec_p3(1,g)) &
px1(2)<px2(2))  | ((mat_prep(2,i)>y_prep_poc_rec_p3(1,g)|
mat_prep(2,i+1)>y_prep_kraj_rec_p3(1,g)) & px1(2)>px2(2)))
         disp('Prepreka je unutar područja.');
         prep_unutar=prep_unutar+1;
         mat_prep_unutar(1,h)=mat_prep(1,i);
         mat_prep_unutar(2,h)=mat_prep(2,i);
         mat_prep_unutar(1,h+1)=mat_prep(1,i+1);
         mat_prep_unutar(2,h+1)=mat_prep(2,i+1);
         h=h+2;

%---------------------------------------------
%prepreka se nalazi izvan područja

    else
         disp('Prepreka se NE nalazi unutar prostora!');
         prep_vani=prep_vani+1;
    end

end
```

```
%ispitivanje da li se prepreka nalazi u donjem području
%ROBOT1

function [nova_prep_dole,novo_najbolji_pojedinac] =
f(nova_mat_prep,najbolji_pojedinac,duz,sir)
%-----------------------------------------
%zadavanje po?etnih uvjeta
e=0; f=0;,m=0;n=0;p=1; s=1;
prep_dole=0;
nova_prep_dole=0;
g=1;h=1;

[a,b] = size(nova_mat_prep);

p1=nova_mat_prep(1:a,1);
matrica_prep=nova_mat_prep(:,2:(b-1));
p2=nova_mat_prep(1:a,b);

%-----------------------------------------
%pronaći najbojeg pojedinca koji odgovara sljedećoj točci u koju ide robot

[e,f]=size(matrica_prep);

naj_pojed=[matrica_prep(1,1);matrica_prep(2,1)];

for i=2:1:f

      if (matrica_prep(1,i)<naj_pojed(1,1)) | (matrica_prep(1,i)==naj_pojed(1,1) &
matrica_prep(2,i)<naj_pojed(2,1))
        g=i;
          naj_pojed(1,1)=matrica_prep(1,i);
        naj_pojed(2,1)=matrica_prep(2,i);
      end
end
naj_pojed;


%-----------------------------------------
%uvrstiti najboljeg pojedinca u gen najboljih pojedinaca

[c,d]=size(najbolji_pojedinac);
a1=0;
for i=1:1:d

    if (najbolji_pojedinac(1,i)==p1(1,1))& (najbolji_pojedinac(2,i)==p1(2,1))
        a1=i;
    end

end

novo_najbolji_pojedinac(1:2,(1:a1))=najbolji_pojedinac(1:2,(1:a1));
novo_najbolji_pojedinac(1:2,(a1+1))=naj_pojed;
novo_najbolji_pojedinac(1:2,(a1+2):(d+1))=najbolji_pojedinac(1:2,((a1+1):d));
```

```
novo_najbolji_pojedinac;

if f>1

%-----------------------------------------------
%pronaći točke koje se nalaze u novoj putanji robota

for i=1:1:f

      if i~=g
       nova_mat_prepreke(1,h)=matrica_prep(1,i);
       nova_mat_prepreke(2,h)=matrica_prep(2,i);
       h=h+1;
      end
end


[j,r]=size(nova_mat_prepreke);

%-----------------------------------------------
%k - koeficijenti smijera -> k=tg?
k1=((p1(2,1)-(sir/2))+0.0000001)/((p1(1,1)-0)+0.0000001);
k2=((p2(2,1)-(sir/2))+0.0000001)/((p2(1,1)-0)+0.0000001);
k3=((p1(2,1)-naj_pojed(2,1))+0.00000011)/((p1(1,1)-naj_pojed(1,1))+0.0000001);
k4=((naj_pojed(2,1)-p2(2,1))+0.0000001)/((naj_pojed(1,1)-p2(1,1))+0.0000001);



%-----------------------------------------------
%b - odsječci na y osi
b1=p1(2,1)-k1*p1(1,1);
b2=p2(2,1)-k2*p2(1,1);
b3=naj_pojed(2,1)-k3*naj_pojed(1,1);
b4=naj_pojed(2,1)-k4*naj_pojed(1,1);

%-----------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:1:r

   y_prep_rec_p1(i)=k1*nova_mat_prepreke(1,i)+b1;
   y_prep_rec_p2(i)=k2*nova_mat_prepreke(1,i)+b2;
   y_prep_rec_p3(i)=k3*nova_mat_prepreke(1,i)+b3;
   y_prep_rec_p4(i)=k4*nova_mat_prepreke(1,i)+b4;
end

%-----------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:1:r

    if (nova_mat_prepreke(2,i)<=y_prep_rec_p4(1,i)) &
(nova_mat_prepreke(2,i)>y_prep_rec_p2(1,i))
```

```
       disp('Prepreka je u donjem području.');
       prep_dole(1,s)=nova_mat_prepreke(1,i);
       prep_dole(2,s)=nova_mat_prepreke(2,i);
       s=s+1;

    else
       disp('Prepreka se NE nalazi unutar prostora!');
   end
end


    if prep_dole>0

        nova_prep_dole(1:2,1)=naj_pojed;
        nova_prep_dole(:,2:s)=prep_dole;
        nova_prep_dole(1:2,(s+1))=p2;
        nova_prep_dole;
    end
end
```

```
%ispitivanje da li se prepreka nalazi u donjem području
%ROBOT2

function [nova_prep_dole,novo_najbolji_pojedinac] =
f(nova_mat_prep,najbolji_pojedinac,duz,sir)
%-----------------------------------------------
%zadavanje po?etnih uvjeta
e=0; f=0;,m=0;n=0;p=1; s=1;
prep_dole=0;
nova_prep_dole=0;
g=1;h=1;

[a,b] = size(nova_mat_prep);

p1=nova_mat_prep(1:a,1);
matrica_prep=nova_mat_prep(:,2:(b-1));
p2=nova_mat_prep(1:a,b);

%-----------------------------------------------
%pronaći najbojeg pojedinca koji odgovara sljedećoj točci u koju ide robot

[e,f]=size(matrica_prep);

naj_pojed=[matrica_prep(1,1);matrica_prep(2,1)];

for i=2:1:f

      if (matrica_prep(1,i)>naj_pojed(1,1)) | (matrica_prep(1,i)==naj_pojed(1,1) &
matrica_prep(2,i)<naj_pojed(2,1))
        g=i;
          naj_pojed(1,1)=matrica_prep(1,i);
        naj_pojed(2,1)=matrica_prep(2,i);
      end
```

```
end
naj_pojed;


%-------------------------------------------
%uvrstiti najboljeg pojedinca u gen najboljih pojedinaca

[c,d]=size(najbolji_pojedinac);
a1=0;
for i=1:1:d

    if (najbolji_pojedinac(1,i)==p1(1,1))& (najbolji_pojedinac(2,i)==p1(2,1))
        a1=i;
    end

end

novo_najbolji_pojedinac(1:2,(1:a1))=najbolji_pojedinac(1:2,(1:a1));
novo_najbolji_pojedinac(1:2,(a1+1))=naj_pojed;
novo_najbolji_pojedinac(1:2,(a1+2):(d+1))=najbolji_pojedinac(1:2,((a1+1):d));

novo_najbolji_pojedinac;

if f>1

%-------------------------------------------
%pronaći točke koje se nalaze u novoj putanji robota

for i=1:1:f

      if i~=g
       nova_mat_prepreke(1,h)=matrica_prep(1,i);
       nova_mat_prepreke(2,h)=matrica_prep(2,i);
       h=h+1;
      end
end
nova_mat_prepreke;

[j,r]=size(nova_mat_prepreke);

%-------------------------------------------
%k - koeficijenti smijera -> k=tg?
k1=(((p1(2,1)-(sir/2))+0.0000001)/((p1(1,1)-duz)+0.0000001));
k2=(((p2(2,1)-(sir/2))+0.0000001)/((p2(1,1)-duz)+0.0000001));
k3=(((p1(2,1)-naj_pojed(2,1))+0.0000001)/((p1(1,1)-naj_pojed(1,1))+0.0000001));
k4=(((naj_pojed(2,1)-p2(2,1))+0.0000001)/((naj_pojed(1,1)-p2(1,1))+0.0000001));


%-------------------------------------------
%b - odsječci na y osi
b1=p1(2,1)-k1*p1(1,1);
b2=p2(2,1)-k2*p2(1,1);
b3=naj_pojed(2,1)-k3*naj_pojed(1,1);
b4=naj_pojed(2,1)-k4*naj_pojed(1,1);
```

```
%-----------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:1:r

   y_prep_rec_p1(i)=k1*nova_mat_prepreke(1,i)+b1;
   y_prep_rec_p2(i)=k2*nova_mat_prepreke(1,i)+b2;
   y_prep_rec_p3(i)=k3*nova_mat_prepreke(1,i)+b3;
   y_prep_rec_p4(i)=k4*nova_mat_prepreke(1,i)+b4;
end

%-----------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:1:r

    if (nova_mat_prepreke(2,i)<=y_prep_rec_p4(1,i)) &
(nova_mat_prepreke(2,i)>y_prep_rec_p2(1,i))

       disp('Prepreka je u donjem području.');
       prep_dole(1,s)=nova_mat_prepreke(1,i);
       prep_dole(2,s)=nova_mat_prepreke(2,i);
       s=s+1;

   else
       disp('Prepreka se NE nalazi unutar prostora!');
   end
end



    if prep_dole>0

        nova_prep_dole(1:2,1)=naj_pojed;
        nova_prep_dole(:,2:s)=prep_dole;
        nova_prep_dole(1:2,(s+1))=p2;
        nova_prep_dole;
    end
end


%ispitivanje da li se prepreka nalazi u gornjem području
%ROBOT1

function [nova_prep_gore,novo_najbolji_pojedinac] =
f(nova_mat_prep,najbolji_pojedinac,duz,sir)
%-----------------------------------------------
%zadavanje po?etnih uvjeta
e=0; f=0;,m=0;n=0;p=1; s=1;
prep_gore=0;

nova_prep_gore=0;
g=1;h=1;

[a,b] = size(nova_mat_prep);
```

```
p1=nova_mat_prep(1:a,1);
matrica_prep=nova_mat_prep(:,2:(b-1));
p2=nova_mat_prep(1:a,b);

%-----------------------------------------------
%pronaći najbojeg pojedinca koji odgovara sljedećoj točci u koju ide robot

[e,f]=size(matrica_prep);

naj_pojed=[matrica_prep(1,1);matrica_prep(2,1)];

for i=2:1:f

      if (matrica_prep(1,i)<naj_pojed(1,1)) | (matrica_prep(1,i)==naj_pojed(1,1) &
matrica_prep(2,i)<naj_pojed(2,1))
        g=i;
          naj_pojed(1,1)=matrica_prep(1,i);
        naj_pojed(2,1)=matrica_prep(2,i);
      end
end
naj_pojed;


%-----------------------------------------------
%uvrstiti najboljeg pojedinca u gen najboljih pojedinaca

[c,d]=size(najbolji_pojedinac);
a1=0;
for i=1:1:d

    if (najbolji_pojedinac(1,i)==p1(1,1))& (najbolji_pojedinac(2,i)==p1(2,1))
        a1=i;
    end

end

novo_najbolji_pojedinac(1:2,(1:a1))=najbolji_pojedinac(1:2,(1:a1));
novo_najbolji_pojedinac(1:2,(a1+1))=naj_pojed;
novo_najbolji_pojedinac(1:2,(a1+2):(d+1))=najbolji_pojedinac(1:2,((a1+1):d));

novo_najbolji_pojedinac;

if f>1

%-----------------------------------------------
%pronaći točke koje se nalaze u novoj putanji robota

for i=1:1:f

      if i~=g
       nova_mat_prepreke(1,h)=matrica_prep(1,i);
       nova_mat_prepreke(2,h)=matrica_prep(2,i);
       h=h+1;
      end
```

```
end


[j,r]=size(nova_mat_prepreke);

%-----------------------------------------------
%k - koeficijenti smijera -> k=tg?
k1=((p1(2,1)-(sir/2))+0.0000001)/((p1(1,1)-0)+0.0000001);
k2=((p2(2,1)-(sir/2))+0.0000001)/((p2(1,1)-0)+0.0000001);
k3=((p1(2,1)-naj_pojed(2,1))+0.0000001)/((p1(1,1)-naj_pojed(1,1))+0.0000001);
k4=((naj_pojed(2,1)-p2(2,1))+0.0000001)/((naj_pojed(1,1)-p2(1,1))+0.0000001);



%-----------------------------------------------
%b - odsječci na y osi
b1=p1(2,1)-k1*p1(1,1);
b2=p2(2,1)-k2*p2(1,1);
b3=naj_pojed(2,1)-k3*naj_pojed(1,1);
b4=naj_pojed(2,1)-k4*naj_pojed(1,1);

%-----------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:1:r

   y_prep_rec_p1(i)=k1*nova_mat_prepreke(1,i)+b1;
   y_prep_rec_p2(i)=k2*nova_mat_prepreke(1,i)+b2;
   y_prep_rec_p3(i)=k3*nova_mat_prepreke(1,i)+b3;
   y_prep_rec_p4(i)=k4*nova_mat_prepreke(1,i)+b4;
end

%-----------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:1:r

    if k3>0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)>=y_prep_rec_p3(1,i)) &
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;

    elseif k3<0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)<=y_prep_rec_p3(1,i)) &
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;
```

```
    else
       disp('Prepreka se NE nalazi unutar prostora!');
   end
end


    if prep_gore>0

        nova_prep_gore(1:2,1)=p1;
        nova_prep_gore(:,2:p)=prep_gore;
        nova_prep_gore(1:2,(p+1))=naj_pojed;
        nova_prep_gore;
    end

end
```

```
%ispitivanje da li se prepreka nalazi u gornjem području
%ROBOT2

function [nova_prep_gore,novo_najbolji_pojedinac] =
f(nova_mat_prep,najbolji_pojedinac,duz,sir)
%-------------------------------------------
%zadavanje po?etnih uvjeta
e=0; f=0;,m=0;n=0;p=1; s=1;
prep_gore=0;

nova_prep_gore=0;
g=1;h=1;

[a,b] = size(nova_mat_prep);

p1=nova_mat_prep(1:a,1);
matrica_prep=nova_mat_prep(:,2:(b-1));
p2=nova_mat_prep(1:a,b);

%-------------------------------------------
%pronaći najbojeg pojedinca koji odgovara sljedećoj točci u koju ide robot

[e,f]=size(matrica_prep);

naj_pojed=[matrica_prep(1,1);matrica_prep(2,1)];

for i=2:1:f

      if (matrica_prep(1,i)>naj_pojed(1,1)) | (matrica_prep(1,i)==naj_pojed(1,1) &
matrica_prep(2,i)<naj_pojed(2,1))
        g=i;
          naj_pojed(1,1)=matrica_prep(1,i);
        naj_pojed(2,1)=matrica_prep(2,i);
      end
end
naj_pojed;
```

```
%-------------------------------------------
%uvrstiti najboljeg pojedinca u gen najboljih pojedinaca

[c,d]=size(najbolji_pojedinac);
a1=0;
for i=1:1:d

    if (najbolji_pojedinac(1,i)==p1(1,1))& (najbolji_pojedinac(2,i)==p1(2,1))
        a1=i;
    end

end

novo_najbolji_pojedinac(1:2,(1:a1))=najbolji_pojedinac(1:2,(1:a1));
novo_najbolji_pojedinac(1:2,(a1+1))=naj_pojed;
novo_najbolji_pojedinac(1:2,(a1+2):(d+1))=najbolji_pojedinac(1:2,((a1+1):d));

novo_najbolji_pojedinac;

if f>1

%-------------------------------------------
%pronaći točke koje se nalaze u novoj putanji robota

for i=1:1:f

      if i~=g
       nova_mat_prepreke(1,h)=matrica_prep(1,i);
       nova_mat_prepreke(2,h)=matrica_prep(2,i);
       h=h+1;
      end
end


[j,r]=size(nova_mat_prepreke);

%-------------------------------------------
%k - koeficijenti smijera -> k=tg?
k1=(((p1(2,1)-(sir/2))+0.0000001)/((p1(1,1)-duz)+0.0000001));
k2=(((p2(2,1)-(sir/2))+0.0000001)/((p2(1,1)-duz)+0.0000001));
k3=(((p1(2,1)-naj_pojed(2,1))+0.0000001)/((p1(1,1)-naj_pojed(1,1))+0.0000001));
k4=(((naj_pojed(2,1)-p2(2,1))+0.0000001)/((naj_pojed(1,1)-p2(1,1))+0.0000001));


%-------------------------------------------
%b - odsječci na y osi
b1=p1(2,1)-k1*p1(1,1);
b2=p2(2,1)-k2*p2(1,1);
b3=naj_pojed(2,1)-k3*naj_pojed(1,1);
b4=naj_pojed(2,1)-k4*naj_pojed(1,1);

%-------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
```

```
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:1:r

   y_prep_rec_p1(i)=k1*nova_mat_prepreke(1,i)+b1;
   y_prep_rec_p2(i)=k2*nova_mat_prepreke(1,i)+b2;
   y_prep_rec_p3(i)=k3*nova_mat_prepreke(1,i)+b3;
   y_prep_rec_p4(i)=k4*nova_mat_prepreke(1,i)+b4;
end

%-------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:1:r

    if k3>0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)<=y_prep_rec_p3(1,i)) &
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;

    elseif k3<0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)>=y_prep_rec_p3(1,i)) &
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;

    else
       disp('Prepreka se NE nalazi unutar prostora!');
   end
end



    if prep_gore>0

        nova_prep_gore(1:2,1)=p1;
        nova_prep_gore(:,2:p)=prep_gore;
        nova_prep_gore(1:2,(p+1))=naj_pojed;
        nova_prep_gore;
    end

end
```

```
%ispitivanje gdje se nalaze prepreke, definiranje matrica za gornje i
%donje područje
%ROBOT11

function [nova_prep_gore,nova_prep_dole,novo_najbolji_pojedinac] =
f(nova_mat_prep,najbolji_pojedinac,duz,sir)
%-----------------------------------------------
%zadavanje po?etnih uvjeta
e=0; f=0;,m=0;n=0;p=1; s=1;
prep_gore=0;
prep_dole=0;
nova_prep_dole=0;
nova_prep_gore=0;
g=1;h=1;

[a,b] = size(nova_mat_prep);

p1=nova_mat_prep(1:a,1);
matrica_prep=nova_mat_prep(:,2:(b-1));
p2=nova_mat_prep(1:a,b);

%-----------------------------------------------
%pronaći najbojeg pojedinca koji odgovara sljedećoj točci u koju ide robot

[e,f]=size(matrica_prep);

naj_pojed=[matrica_prep(1,1);matrica_prep(2,1)];

for i=2:1:f

      if (matrica_prep(1,i)<naj_pojed(1,1)) | (matrica_prep(1,i)==naj_pojed(1,1) &
matrica_prep(2,i)<naj_pojed(2,1))
        g=i;
          naj_pojed(1,1)=matrica_prep(1,i);
        naj_pojed(2,1)=matrica_prep(2,i);
      end
end
naj_pojed;


%-----------------------------------------------
%uvrstiti najboljeg pojedinca u gen najboljih pojedinaca

[c,d]=size(najbolji_pojedinac);
a1=0;
for i=1:1:d

    if (najbolji_pojedinac(1,i)==p1(1,1))& (najbolji_pojedinac(2,i)==p1(2,1))
        a1=i;
    end

end
```

```
novo_najbolji_pojedinac(1:2,(1:a1))=najbolji_pojedinac(1:2,(1:a1));
novo_najbolji_pojedinac(1:2,(a1+1))=naj_pojed;
novo_najbolji_pojedinac(1:2,(a1+2):(d+1))=najbolji_pojedinac(1:2,((a1+1):d));

novo_najbolji_pojedinac;

if f>1

%-----------------------------------------------
%pronaći točke koje se nalaze u novoj putanji robota

for i=1:1:f

      if i~=g
       nova_mat_prepreke(1,h)=matrica_prep(1,i);
       nova_mat_prepreke(2,h)=matrica_prep(2,i);
       h=h+1;
      end
end


[j,r]=size(nova_mat_prepreke);

%-----------------------------------------------
%k - koeficijenti smijera -> k=tg?
k1=((p1(2,1)-(sir/2))+0.0000001)/((p1(1,1)-0)+0.0000001);
k2=((p2(2,1)-(sir/2))+0.0000001)/((p2(1,1)-0)+0.0000001);
k3=((p1(2,1)-naj_pojed(2,1))+0.0000001)/((p1(1,1)-naj_pojed(1,1))+0.0000001);
k4=((naj_pojed(2,1)-p2(2,1))+0.0000001)/((naj_pojed(1,1)-p2(1,1))+0.0000001);


%-----------------------------------------------
%b - odsječci na y osi
b1=p1(2,1)-k1*p1(1,1);
b2=p2(2,1)-k2*p2(1,1);
b3=naj_pojed(2,1)-k3*naj_pojed(1,1);
b4=naj_pojed(2,1)-k4*naj_pojed(1,1);

%-----------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:1:r

   y_prep_rec_p1(i)=k1*nova_mat_prepreke(1,i)+b1;
   y_prep_rec_p2(i)=k2*nova_mat_prepreke(1,i)+b2;
   y_prep_rec_p3(i)=k3*nova_mat_prepreke(1,i)+b3;
   y_prep_rec_p4(i)=k4*nova_mat_prepreke(1,i)+b4;
end

%-----------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:1:r
```

```
    if k3>0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)>=y_prep_rec_p3(1,i)) &
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;

    elseif k3<0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)<=y_prep_rec_p3(1,i))&
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;

   elseif ((k4<0 & nova_mat_prepreke(2,i)<=y_prep_rec_p4(1,i)) |(k4>0 &
nova_mat_prepreke(2,i)>=y_prep_rec_p4(1,i))) &
(nova_mat_prepreke(2,i)>y_prep_rec_p2(1,i))

       disp('Prepreka je u donjem području.');
       prep_dole(1,s)=nova_mat_prepreke(1,i);
       prep_dole(2,s)=nova_mat_prepreke(2,i);
       s=s+1;

   else
       disp('Prepreka se NE nalazi unutar prostora!');
   end
end


    if prep_gore>0

        nova_prep_gore(1:2,1)=p1;
        nova_prep_gore(:,2:p)=prep_gore;
        nova_prep_gore(1:2,(p+1))=naj_pojed;
        nova_prep_gore;
    end


    if prep_dole>0

        nova_prep_dole(1:2,1)=naj_pojed;
        nova_prep_dole(:,2:s)=prep_dole;
        nova_prep_dole(1:2,(s+1))=p2;
        nova_prep_dole;
    end

end
```

```
%ispitivanje gdje se nalaze prepreke, definiranje matrica za gornje i
%donje područje
%ROBOT12

function [nova_prep_gore,nova_prep_dole,novo_najbolji_pojedinac] =
f(nova_mat_prep,najbolji_pojedinac,duz,sir)
%-----------------------------------------------
%zadavanje po?etnih uvjeta
e=0; f=0;,m=0;n=0;p=1; s=1;
prep_gore=0;
prep_dole=0;
nova_prep_dole=0;
nova_prep_gore=0;
g=1;h=1;

[a,b] = size(nova_mat_prep);

p1=nova_mat_prep(1:a,1);
matrica_prep=nova_mat_prep(:,2:(b-1));
p2=nova_mat_prep(1:a,b);

%-----------------------------------------------
%pronaći najbojeg pojedinca koji odgovara sljedećoj točci u koju ide robot

[e,f]=size(matrica_prep);

naj_pojed=[matrica_prep(1,1);matrica_prep(2,1)];

for i=2:1:f

      if (matrica_prep(1,i)>naj_pojed(1,1)) | (matrica_prep(1,i)==naj_pojed(1,1) &
matrica_prep(2,i)>naj_pojed(2,1))
        g=i;
          naj_pojed(1,1)=matrica_prep(1,i);
        naj_pojed(2,1)=matrica_prep(2,i);
      end
end
naj_pojed;


%-----------------------------------------------
%uvrstiti najboljeg pojedinca u gen najboljih pojedinaca

[c,d]=size(najbolji_pojedinac);
a1=0;
for i=1:1:d

    if (najbolji_pojedinac(1,i)==p1(1,1))& (najbolji_pojedinac(2,i)==p1(2,1))
        a1=i;
    end

end
```

```
novo_najbolji_pojedinac(1:2,(1:a1))=najbolji_pojedinac(1:2,(1:a1));
novo_najbolji_pojedinac(1:2,(a1+1))=naj_pojed;
novo_najbolji_pojedinac(1:2,(a1+2):(d+1))=najbolji_pojedinac(1:2,((a1+1):d));

novo_najbolji_pojedinac;

if f>1

%-------------------------------------------
%pronaći točke koje se nalaze u novoj putanji robota

for i=1:1:f

      if i~=g
       nova_mat_prepreke(1,h)=matrica_prep(1,i);
       nova_mat_prepreke(2,h)=matrica_prep(2,i);
       h=h+1;
      end
end


[j,r]=size(nova_mat_prepreke);
nova_mat_prepreke;

%-------------------------------------------
%k - koeficijenti smijera -> k=tg?
k1=(((p1(2,1)-(sir/2))+0.0000001)/((p1(1,1)-duz)+0.0000001));
k2=(((p2(2,1)-(sir/2))+0.0000001)/((p2(1,1)-duz)+0.0000001));
k3=(((p1(2,1)-naj_pojed(2,1))+0.0000001)/((p1(1,1)-naj_pojed(1,1))+0.0000001));
k4=(((naj_pojed(2,1)-p2(2,1))+0.0000001)/((naj_pojed(1,1)-p2(1,1))+0.0000001));


%-------------------------------------------
%b - odsječci na y osi
b1=p1(2,1)-k1*p1(1,1);
b2=p2(2,1)-k2*p2(1,1);
b3=naj_pojed(2,1)-k3*naj_pojed(1,1);
b4=naj_pojed(2,1)-k4*naj_pojed(1,1);

%-------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:1:r

   y_prep_rec_p1(i)=k1*nova_mat_prepreke(1,i)+b1;
   y_prep_rec_p2(i)=k2*nova_mat_prepreke(1,i)+b2;
   y_prep_rec_p3(i)=k3*nova_mat_prepreke(1,i)+b3;
   y_prep_rec_p4(i)=k4*nova_mat_prepreke(1,i)+b4;
end

%-------------------------------------------
%ispitivanje da li se prepreka nalazi u radnom prostoru

for i=1:1:r
```

```
    if k3>0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)<=y_prep_rec_p3(1,i)) &
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;

    elseif k3<0 & (nova_mat_prepreke(2,i)<y_prep_rec_p1(1,i)) &
(nova_mat_prepreke(2,i)>=y_prep_rec_p3(1,i))&
(nova_mat_prepreke(2,i)>naj_pojed(2,1))

        disp('Prepreka je u gornjem području.');
        prep_gore(1,p)=nova_mat_prepreke(1,i);
        prep_gore(2,p)=nova_mat_prepreke(2,i);
        p=p+1;


   elseif ((k4<0 & nova_mat_prepreke(2,i)>=y_prep_rec_p4(1,i)) |(k4>0 &
nova_mat_prepreke(2,i)<=y_prep_rec_p4(1,i))) &
(nova_mat_prepreke(2,i)>y_prep_rec_p2(1,i)) &
(nova_mat_prepreke(1,i)<naj_pojed(1,1))

       disp('Prepreka je u donjem području.');
       prep_dole(1,s)=nova_mat_prepreke(1,i);
       prep_dole(2,s)=nova_mat_prepreke(2,i);
       s=s+1;

   else
       disp('Prepreka se NE nalazi unutar prostora!');

   end
end

     if prep_gore>0

        nova_prep_gore(1:2,1)=p1;
        nova_prep_gore(:,2:p)=prep_gore;
        nova_prep_gore(1:2,(p+1))=naj_pojed;
        nova_prep_gore;
    end


    if prep_dole>0

        nova_prep_dole(1:2,1)=naj_pojed;
        nova_prep_dole(:,2:s)=prep_dole;
        nova_prep_dole(1:2,(s+1))=p2;
        nova_prep_dole;
    end

end
```

```
%Ispitivanje podrucja u kojem se nalaze roboti za početnu konfiguraciju

function
[mat_prepreka_r1,br_prep_unutar_1,mat_prepreka_r2,br_prep_unutar_2,br_prep_sijeku_1
,br_prep_sijeku_2,prep_sijece_dole1,prep_sijece_dole2,prep_sijece_gore1,prep_sijece
_gore2,toc_1_poc_1,toc_1_poc_2,toc_1_kraj_1,toc_1_kraj_2] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,xpr1,ypr1,xpr2,ypr2,duz,sir,r_h
_1,r_h_2,f_h,gibanje_R1,gibanje_R2)

br_prep_r1=0;
br_prep_r2=0;
br_prep_sijeku_1=0;
br_prep_sijeku_2=0;
%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće PRVI robot

px11=[0 toc_1_poc_1(1)]; %x koord početne i krajnje točke pravca p1
py11=[(sir/2) toc_1_poc_1(2)]; %y koord početne i krajnje točke pravca p1

px12=[0 toc_1_kraj_1(1)]; %x koord početne i krajnje točke pravca p2
py12=[(sir/2) toc_1_kraj_1(2)]; %y koord početne i krajnje točke pravca p2

px13=[px11(2) px12(2)];
py13=[py11(2) py12(2)];

%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće DRUGI robot

px21=[duz toc_1_poc_2(1)]; %x koord početne i krajnje točke pravca p1
py21=[(sir/2) toc_1_poc_2(2)]; %y koord početne i krajnje točke pravca p1

px22=[duz toc_1_kraj_2(1)]; %x koord početne i krajnje točke pravca p2
py22=[(sir/2) toc_1_kraj_2(2)]; %y koord početne i krajnje točke pravca p2

px23=[px21(2) px22(2)];
py23=[py21(2) py22(2)];

%-------------------------------------------
%crtanje radnog prosrtora

figure(1)
axis equal
title 'RADNI PROSTOR PRVOG I DRUGOG ROBOTA'
hold on
axis([0 duz 0 sir])

line(px11,py11,'Color','b','LineWidth',1);  %pravac p1 ROBOT 1
line(px12,py12,'Color','b','LineWidth',1);  %pravac p2 ROBOT 1
line(px13,py13,'Color','c','LineWidth',1);  %pravac p3 ROBOT 1
line(px21,py21,'Color','r','LineWidth',1);  %pravac p1 ROBOT 2
line(px22,py22,'Color','r','LineWidth',1);  %pravac p2 ROBOT 2
line(px23,py23,'Color','m','LineWidth',1);  %pravac p3 ROBOT 2
```

```
%ISPITIVANJE DA LI SE DRUGI ROBOT NALAZI U PODRUČJU RADA PRVOG ROBOTA

[prep_sijece_oba1,prep_sijece_gore1,prep_sijece_dole1,br_prep_izvan_1,br_prep_unuta
r_1,mat_prepreka_r1] =
f_isp_prep_rp1_r1(toc_1_poc_1,toc_1_kraj_1,xpr2,ypr2,2,duz,sir);

%-----------------------------------------------
%definiranje prepreka za PRVI robot
    [c,d]=size(mat_prepreka_r1);
%-----------------------------------------------
br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_dole1+prep_sijece_gore1;

if prep_sijece_oba1==0 & prep_sijece_gore1==0 & prep_sijece_dole1==0

    br_prep_sijeku_1=0;

    if br_prep_unutar_1>0

%crtanje prepreka
    for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
    end
    end
end
%-----------------------------------------------
%-----------------------------------------------
if  gibanje_R1(1)==1 & gibanje_R2(2)==1 %2 slučaj
    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_dole1;

if prep_sijece_oba1==0 & prep_sijece_dole1==0 & prep_sijece_gore1>0

    if br_prep_unutar_1>0

%-----------------------------------------------
%crtanje prepreka
 for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_poc_1(2)-(sir/2))/(toc_1_poc_1(1)-0)));
        k2=(atan((toc_1_poc_1(2)-(sir/2))/(toc_1_poc_1(1)-0)));

        toc_1_poc_2(1)= toc_1_poc_1(1)+((cos(k1)*(r_h_2*f_h)));
        toc_1_poc_2(2)= toc_1_poc_1(2)+((sin(k1)*(r_h_2*f_h)));
end
end
%-----------------------------------------------
```

```
%-----------------------------------------------
if  gibanje_R1(2)==1 & gibanje_R2(2)==1 %1 slučaj
    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_gore1;

if prep_sijece_oba1==0 & prep_sijece_gore1==0 & prep_sijece_dole1>0

    if br_prep_unutar_1>0

%-----------------------------------------------
%crtanje prepreka
 for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_kraj_1(2)-(sir/2))/(toc_1_kraj_1(1)-0)));
        k2=(atan((toc_1_kraj_1(2)-(sir/2))/(toc_1_kraj_1(1)-0)));

        toc_1_poc_2(1)= toc_1_kraj_1(1)+((cos(k1)*(r_h_2*f_h)));
        toc_1_poc_2(2)= toc_1_kraj_1(2)-((sin(k1)*(r_h_2*f_h)));
end
end
%-----------------------------------------------
%-----------------------------------------------
if  gibanje_R1(1)==1 & gibanje_R2(1)==1 %3 slucaj
    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_gore1;

if prep_sijece_oba1==0 & prep_sijece_gore1==0 & prep_sijece_dole1>0

    if br_prep_unutar_1>0

%-----------------------------------------------
%crtanje prepreka
 for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_poc_2(2)-(sir/2))/(duz-toc_1_poc_2(1))));
        k2=(atan((toc_1_poc_2(2)-(sir/2))/(duz-toc_1_poc_2(1))));

        toc_1_kraj_1(1)= toc_1_poc_2(1)-((cos(k1)*(r_h_1*f_h)));
        toc_1_kraj_1(2)= toc_1_poc_2(2)+((sin(k1)*(r_h_1*f_h)));
  end
end

%-----------------------------------------------
%-----------------------------------------------
if  gibanje_R1(2)==1 & gibanje_R2(1)==1 %4 slučaj
    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_dole1;
```

```
if prep_sijece_oba1==0 & prep_sijece_dole1==0 & prep_sijece_gore1>0

    if br_prep_unutar_1>0

%-------------------------------------------
%crtanje prepreka
 for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_poc_2(2)-(sir/2))/(duz-toc_1_poc_2(1))));
        k2=(atan((toc_1_poc_2(2)-(sir/2))/(duz-toc_1_poc_2(1))));

        toc_1_poc_1(1)= toc_1_poc_2(1)-((cos(k1)*(r_h_1*f_h)));
        toc_1_poc_1(2)= toc_1_poc_2(2)-((sin(k1)*(r_h_1*f_h)));
end
end
%-------------------------------------------

%ISPITIVANJE DA LI SE PRVI ROBOT NALAZI U PODRUČJU RADA DRUGOG ROBOTA
%-------------------------------------------
[prep_sijece_oba2,prep_sijece_gore2,prep_sijece_dole2,br_prep_izvan_2,br_prep_unuta
r_2,mat_prepreka_r2] =
f_isp_prep_rp1_r2(toc_1_poc_2,toc_1_kraj_2,xpr1,ypr1,2,duz,sir);
```

```
%Ispitivanje podrucja u kojem se nalaze roboti za krajnju konfiguraciju

function
[mat_prepreka_r1,br_prep_unutar_1,mat_prepreka_r2,br_prep_unutar_2,br_prep_sijeku_1
,br_prep_sijeku_2,prep_sijece_gore1,prep_sijece_gore2,prep_sijece_dole1,prep_sijece
_dole2,toc_1_poc_1,toc_1_poc_2,toc_1_kraj_1,toc_1_kraj_2] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,xpr1,ypr1,xpr2,ypr2,duz,sir,r_h
_1,r_h_2,f_h,gibanje_R1,gibanje_R2)

br_prep_r1=0;
br_prep_r2=0;
br_prep_sijeku_1=0;
br_prep_sijeku_2=0;
%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće PRVI robot

px11=[0 toc_1_poc_1(1)]; %x koord početne i krajnje točke pravca p1
py11=[(sir/2) toc_1_poc_1(2)]; %y koord početne i krajnje točke pravca p1

px12=[0 toc_1_kraj_1(1)]; %x koord početne i krajnje točke pravca p2
py12=[(sir/2) toc_1_kraj_1(2)]; %y koord početne i krajnje točke pravca p2
```

```
px13=[px11(2) px12(2)];
py13=[py11(2) py12(2)];

%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće DRUGI robot

px21=[duz toc_1_poc_2(1)]; %x koord početne i krajnje točke pravca p1
py21=[(sir/2) toc_1_poc_2(2)]; %y koord početne i krajnje točke pravca p1

px22=[duz toc_1_kraj_2(1)]; %x koord početne i krajnje točke pravca p2
py22=[(sir/2) toc_1_kraj_2(2)]; %y koord početne i krajnje točke pravca p2

px23=[px21(2) px22(2)];
py23=[py21(2) py22(2)];

%-----------------------------------------
%crtanje radnog prosrtora

figure(4)
axis equal
title 'RADNI PROSTOR PRVOG I DRUGOG ROBOTA'
hold on
axis([0 duz 0 sir])

line(px11,py11,'Color','b','LineWidth',1);  %pravac p1 ROBOT 1
line(px12,py12,'Color','b','LineWidth',1);  %pravac p2 ROBOT 1
line(px13,py13,'Color','c','LineWidth',1);  %pravac p3 ROBOT 1
line(px21,py21,'Color','r','LineWidth',1);  %pravac p1 ROBOT 2
line(px22,py22,'Color','r','LineWidth',1);  %pravac p2 ROBOT 2
line(px23,py23,'Color','m','LineWidth',1);  %pravac p3 ROBOT 2


%ISPITIVANJE DA LI SE DRUGI ROBOT NALAZI U PODRUČJU RADA PRVOG ROBOTA

[prep_sijece_oba1,prep_sijece_gore1,prep_sijece_dole1,br_prep_izvan_1,br_prep_unuta
r_1,mat_prepreka_r1] =
f_isp_prep_rp2_r1(toc_1_poc_1,toc_1_kraj_1,xpr2,ypr2,2,duz,sir);

%-----------------------------------------
%definiranje prepreka za PRVI robot
 [c,d]=size(mat_prepreka_r1);
%-----------------------------------------
br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_dole1+prep_sijece_gore1;

if prep_sijece_oba1==0 & prep_sijece_dole1==0 & prep_sijece_gore1==0

    br_prep_sijeku_1=0;

    if br_prep_unutar_1>0

%crtanje prepreka
  for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
```

```
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
  end
    end
end
%-------------------------------------------
%-------------------------------------------
if  gibanje_R1(1)==1 & gibanje_R2(1)==1 %5 slučaj

    if prep_sijece_oba1==0 & prep_sijece_gore1>0 & prep_sijece_dole1==0

    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_dole1;

     if br_prep_unutar_1>0

%-------------------------------------------
 %crtanje prepreka
     for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_poc_1(2)-(sir/2))/(toc_1_poc_1(1)-0)));
        k2=(atan((toc_1_poc_1(2)-(sir/2))/(toc_1_poc_1(1)-0)));
        toc_1_kraj_2(1)= toc_1_poc_1(1)+((cos(k1)*(r_h_2*f_h)));
        toc_1_kraj_2(2)= toc_1_poc_1(2)+((sin(k1)*(r_h_2*f_h)));
    end
end
%-------------------------------------------
%-------------------------------------------
if  gibanje_R1(2)==1 & gibanje_R2(2)==1 %6 slučaj

    if prep_sijece_oba1==0 & prep_sijece_gore1>0 & prep_sijece_dole1==0

    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_dole1;

     if br_prep_unutar_1>0

%-------------------------------------------
 %crtanje prepreka
     for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_kraj_2(2)-(sir/2))/(duz-toc_1_kraj_2(1))));
        k2=(atan((toc_1_kraj_2(2)-(sir/2))/(duz-toc_1_kraj_2(1))));
        toc_1_poc_1(1)= toc_1_kraj_2(1)-((cos(k1)*(r_h_1*f_h)));
        toc_1_poc_1(2)= toc_1_kraj_2(2)-((sin(k1)*(r_h_1*f_h)));
    end
end
```

```
%-----------------------------------------------
%-----------------------------------------------
if  gibanje_R1(1)==1 & gibanje_R2(2)==1 %7 slučaj

    if prep_sijece_oba1==0 & prep_sijece_dole1>0 & prep_sijece_gore1==0

    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_gore1;

     if br_prep_unutar_1>0

%-----------------------------------------------
 %crtanje prepreka
     for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_kraj_2(2)-(sir/2))/(duz-toc_1_kraj_2(1))));
        k2=(atan((toc_1_kraj_2(2)-(sir/2))/(duz-toc_1_kraj_2(1))));

        toc_1_kraj_1(1)= toc_1_kraj_2(1)-((cos(k1)*(r_h_1*f_h)));
        toc_1_kraj_1(2)= toc_1_kraj_2(2)+((sin(k1)*(r_h_1*f_h)));
    end
end

%-----------------------------------------------
%-----------------------------------------------
if  gibanje_R1(2)==1 & gibanje_R2(1)==1 %8 slučaj

    if prep_sijece_oba1==0 & prep_sijece_dole1>0 & prep_sijece_gore1==0

    br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_gore1;

     if br_prep_unutar_1>0

%-----------------------------------------------
 %crtanje prepreka
     for i=1:2:d
        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);
        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);
        line(prx,pry,'Color','g','LineWidth',2);
 end
    end
        k1=abs(atan((toc_1_kraj_1(2)-(sir/2))/(toc_1_kraj_1(1)-0)));
        k2=(atan((toc_1_kraj_1(2)-(sir/2))/(toc_1_kraj_1(1)-0)));

        toc_1_kraj_2(1)= toc_1_kraj_1(1)+((cos(k1)*(r_h_2*f_h)));
        toc_1_kraj_2(2)= toc_1_kraj_1(2)+((sin(k1)*(r_h_2*f_h)));
    end
end
%-----------------------------------------------
```

```
%ISPITIVANJE DA LI SE PRVI ROBOT NALAZI U PODRUČJU RADA DRUGOG ROBOTA
%-------------------------------------------
[prep_sijece_oba2,prep_sijece_gore2,prep_sijece_dole2,br_prep_izvan_2,br_prep_unuta
r_2,mat_prepreka_r2] =
f_isp_prep_rp2_r2(toc_1_poc_2,toc_1_kraj_2,xpr1,ypr1,2,duz,sir);
```

```
%kod za zaobilaženje prepreka
%ROBOT 1

function [najbolji_pojedinac,toc_1_poc_1,toc_1_kraj_1] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz_prostora,sir_prostora,mat_p
rep,r_h_1,f_h)


%-------------------------------------------
%zadavanje po?etnih uvjeta
a=0; b=0;a1=0;a11=0;a2=0;a21=0; a22=1; c=0;e=0;
f=0;,m=0;n=0;p=0;b1=0;b11=0;c1=0;c11=0;x=0;
prep_sijece=0;
prep_unutar=0;
prep_vani=0;
%-------------------------------------------
%definiranje prostora u kojem se kreće PRVI robot

px1=[0 toc_1_poc_1(1)];                %x koord početne i krajnje točke pravca p1
py1=[(sir_prostora/2) toc_1_poc_1(2)]; %y koord početne i krajnje točke pravca p1
p1=[px1;py1];
p11=[px1(2);py1(2)];

px2=[0 toc_1_kraj_1(1)];                %x koord početne i krajnje točke pravca p2
py2=[(sir_prostora/2) toc_1_kraj_1(2)]; %y koord početne i krajnje točke pravca p2
p2=[px2;py2];
p22=[px2(2);py2(2)];

%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće DRUGI robot

px21=[duz_prostora toc_1_poc_2(1)];      %x koord početne i krajnje točke pravca p1
py21=[(sir_prostora/2) toc_1_poc_2(2)];  %y koord početne i krajnje točke pravca p1

px22=[duz_prostora toc_1_kraj_2(1)];    %x koord početne i krajnje točke pravca p2
py22=[(sir_prostora/2) toc_1_kraj_2(2)];%y koord početne i krajnje točke pravca p2

px23=[px21(2) px22(2)];  %x koord početne i krajnje točke pravca p3
py23=[py21(2) py22(2)];  %x koord početne i krajnje točke pravca p3

%-------------------------------------------------------------------------
%matrica prepreke
crt_mat_prep=mat_prep;
[a2,a21] = size(mat_prep);
```

```
mat_prep;
   nova_prep=[mat_prep(1,1);mat_prep(2,1)];

for i=2:a21

    for j=1:a22

        if mat_prep(1,i)~=nova_prep(1,j) | mat_prep(2,i)~=nova_prep(2,j)
         x=x+1;

        end
            if x==a22
                nova_prep(1,a22+1)=mat_prep(1,i);
                nova_prep(2,a22+1)=mat_prep(2,i);
                   a22=a22+1;
            end
      end
    x=0;
end

mat_prep=nova_prep;
[a1,a11] = size(mat_prep);
%-----------------------------------------------
%k - koeficijenti smijera -> k=tg?

k1=((py1(2)-py1(1))+0.0000001)/((px1(2)-px1(1))+0.0000001);
k2=((py2(2)-py2(1))+0.0000001)/((px2(2)-px2(1))+0.0000001);
k3=((py1(2)-py2(2))+0.0000001)/((px1(2)-px2(2))+0.0000001);

%-----------------------------------------------
%b - odsječci na y osi

b1=py1(2)-k1*px1(2);
b2=py2(2)-k2*px2(2);
b3=py1(2)-k3*px1(2);

%-----------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:a11
    e=e+1;
   y_prep_p1(e)=k1*mat_prep(1,i)+b1;
   y_prep_p2(e)=k2*mat_prep(1,i)+b2;
   y_prep_p3(e)=k3*mat_prep(1,i)+b3;

   if mat_prep(2,i)< y_prep_p1(e) & mat_prep(2,i)> y_prep_p2(e) &
((mat_prep(2,i)>y_prep_p3(e)& px1(2)>px2(2))|(mat_prep(2,i)<y_prep_p3(e)&
px1(2)<px2(2)))
        c=c+1;
        mat_prepreke(1,c)=mat_prep(1,i);
        mat_prepreke(2,c)=mat_prep(2,i);
   end
end
```

```
[a,b] = size(mat_prepreke);

%-------------------------------------------
%definiranje nove matrice prepreka
nova_mat_prep(1:a,1)=p11;
nova_mat_prep(:,2:(b+1))=mat_prepreke;
nova_mat_prep(1:a,(b+2))=p22;
nova_mat_prep;

najbolji_pojedinac=[p11 p22];

[nova_prep_gore,nova_prep_dole,novo_najbolji_pojedinac]=isp_prepreke_r1(nova_mat_pr
ep,najbolji_pojedinac,duz_prostora,sir_prostora);

najbolji_pojedinac=novo_najbolji_pojedinac;


[b1,b11] = size(nova_prep_dole);
[c1,c11] = size(nova_prep_gore);



    if c11~=0
       while c11~=1
        nova_mat_prep=nova_prep_gore;


[nova_prep_gore,novo_najbolji_pojedinac]=isp_prepreke_gore_r1(nova_mat_prep,najbolj
i_pojedinac,duz_prostora,sir_prostora);
        najbolji_pojedinac=novo_najbolji_pojedinac;
        [c1,c11] = size(nova_prep_gore);
       end

    else

        najbolji_pojedinac=novo_najbolji_pojedinac;
  end



    if b11~=0

        while b11~=1

            nova_mat_prep=nova_prep_dole;


[nova_prep_dole,novo_najbolji_pojedinac]=isp_prepreke_dole_r1(nova_mat_prep,najbolj
i_pojedinac,duz_prostora,sir_prostora);
        najbolji_pojedinac=novo_najbolji_pojedinac;
        [b1,b11] = size(nova_prep_dole);
        end
    else
```

```
        najbolji_pojedinac=novo_najbolji_pojedinac;

    end

%skraćenje putanje robota zbog promjera hvataljke

[e,f] = size(najbolji_pojedinac);



    k1=abs(atan((najbolji_pojedinac(2,1)-
(sir_prostora/2))/(najbolji_pojedinac(1,1)-0)));
    k2=(atan((najbolji_pojedinac(2,1)-(sir_prostora/2))/(najbolji_pojedinac(1,1)-
0)));

    naj_pojedinac(1,1)= najbolji_pojedinac(1,1)-(cos(k1)*r_h_1);
    naj_pojedinac(2,1)= najbolji_pojedinac(2,1)-(sin(k2)*r_h_1);

    toc_1_poc_1(1,1)=naj_pojedinac(1,1);
    toc_1_poc_1(1,2)=naj_pojedinac(2,1);
 if f>2

    for i=2:1:(f-1)
        k21=abs(atan((najbolji_pojedinac(2,i)-
(sir_prostora/2))/(najbolji_pojedinac(1,i)-0)));
        k22=(atan((najbolji_pojedinac(2,i)-
(sir_prostora/2))/(najbolji_pojedinac(1,i)-0)));

        k11=abs(atan((najbolji_pojedinac(2,i)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,i))));
        k12=(atan((najbolji_pojedinac(2,i)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,i))));

        Xm=najbolji_pojedinac(1,i)-((cos(k11)*(r_h_1*f_h)));
        Ym=najbolji_pojedinac(2,i)+((sin(k12)*(r_h_1*f_h)));

        naj_pojedinac(1,i)= Xm-((cos(k21)*(r_h_1*f_h)));
        naj_pojedinac(2,i)= Ym-((sin(k22)*(r_h_1*f_h)));

    end
 end

    k1=abs(atan((najbolji_pojedinac(2,f)-
(sir_prostora/2))/(najbolji_pojedinac(1,f)-0)));
    k2=(atan((najbolji_pojedinac(2,f)-(sir_prostora/2))/(najbolji_pojedinac(1,f)-
0)));

    naj_pojedinac(1,f)= najbolji_pojedinac(1,f)-(cos(k1)*r_h_1);
    naj_pojedinac(2,f)= najbolji_pojedinac(2,f)-(sin(k2)*r_h_1);

    toc_1_kraj_1(1,1)=naj_pojedinac(1,f);
    toc_1_kraj_1(1,2)=naj_pojedinac(2,f);

najbolji_pojedinac=naj_pojedinac; %NE BRISATI
```

```
figure(9)
axis equal
title 'PUTANJA PRVOG ROBOTA'
hold on
axis ([0 duz_prostora 0 sir_prostora]);

line(px1,py1,'Color','b','LineWidth',1);    %pravac p1 ROBOT 1
line(px2,py2,'Color','b','LineWidth',1);    %pravac p2 ROBOT 1

%-----------------------------------------------
%definiranje i crtanje prepreka
for i=1:2:a21
    prx(1,1)=crt_mat_prep(1,i);
    pry(1,1)=crt_mat_prep(2,i);

    prx(1,2)=crt_mat_prep(1,i+1);
    pry(1,2)=crt_mat_prep(2,i+1);

line(prx,pry,'Color','k','LineWidth',2);
end

[e,f] = size(najbolji_pojedinac);
%-----------------------------------------------
%crtanje putanje robota
for i=1:1:f-1
    tocx(1,1)= najbolji_pojedinac(1,i);
    tocy(1,1)= najbolji_pojedinac(2,i);

    tocx(1,2)= najbolji_pojedinac(1,i+1);
    tocy(1,2)= najbolji_pojedinac(2,i+1);

    line(tocx,tocy,'Color','m','LineWidth',2);
end
hold off



%-----------------------------------------------
%provlačenje krivulje viseg reda po putanji robota
figure(10)
axis equal
title 'PUTANJA PRVOG ROBOTA'
hold on
axis ([0 duz_prostora 0 sir_prostora]);

line(px1,py1,'Color','b','LineWidth',1);    %pravac p1 ROBOT 1
line(px2,py2,'Color','b','LineWidth',1);    %pravac p2 ROBOT 1

%-----------------------------------------------
%definiranje i crtanje prepreka
for i=1:2:a21
    prx(1,1)=crt_mat_prep(1,i);
    pry(1,1)=crt_mat_prep(2,i);

    prx(1,2)=crt_mat_prep(1,i+1);
    pry(1,2)=crt_mat_prep(2,i+1);
```

```
line(prx,pry,'Color','k','LineWidth',2);
end

%-----------------------------------------
%crtanje najboljeg pojedinca
for i=1:f
    x(i)= najbolji_pojedinac(1,i);
    y(i)=najbolji_pojedinac(2,i);
end
t=1:f;
ts=1:1/100:f;
xs=spline(t,x,ts);
ys=spline(t,y,ts);
hold on
plot(xs,ys,'Color','b','LineWidth',2);
hold off
```

```
%kod za zaobilaženje prepreka
%ROBOT 2

function [najbolji_pojedinac,toc_1_poc_2,toc_1_kraj_2] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz_prostora,sir_prostora,mat_p
rep,r_h_2,f_h)


%-----------------------------------------
%zadavanje po?etnih uvjeta
a=0; b=0;a1=0;a11=0;a22=1;a2=0;a21=0;c=0;e=0;
f=0;,m=0;n=0;p=0;b1=0;b11=0;c1=0;c11=0;x=0;
prep_sijece=0;
prep_unutar=0;
prep_vani=0;
%-----------------------------------------
%definiranje prostora u kojem se kreće PRVI robot

px1=[0 toc_1_poc_1(1)];                %x koord početne i krajnje točke pravca p1
py1=[(sir_prostora/2) toc_1_poc_1(2)]; %y koord početne i krajnje točke pravca p1

px2=[0 toc_1_kraj_1(1)];               %x koord početne i krajnje točke pravca p2
py2=[(sir_prostora/2) toc_1_kraj_1(2)];%y koord početne i krajnje točke pravca p2

px13=[px1(2) px2(2)];  %x koord početne i krajnje točke pravca p3
py13=[py1(2) py2(2)];  %y koord početne i krajnje točke pravca p3


%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće DRUGI robot

px21=[duz_prostora toc_1_poc_2(1)]; %x koord početne i krajnje točke pravca p1
py21=[(sir_prostora/2) toc_1_poc_2(2)]; %y koord početne i krajnje točke pravca p1
```

```
p1=[px21;py21];
p11=[px21(2);py21(2)];

px22=[duz_prostora toc_1_kraj_2(1)];    %x koord početne i krajnje točke pravca p2
py22=[(sir_prostora/2) toc_1_kraj_2(2)];%y koord početne i krajnje točke pravca p2
p2=[px22;py22];
p22=[px22(2);py22(2)];

px23=[px21(2) px22(2)];  %x koord početne i krajnje točke pravca p3
py23=[py21(2) py22(2)];  %y koord početne i krajnje točke pravca p3

%-------------------------------------------------------------------------
%matrica prepreke
mat_prep;
crt_mat_prep=mat_prep;
[a2,a21] = size(mat_prep);

   nova_prep=[mat_prep(1,1);mat_prep(2,1)];

for i=2:a21
    for j=1:a22
        if mat_prep(1,i)~=nova_prep(1,j) | mat_prep(2,i)~=nova_prep(2,j)
         x=x+1;
        end
            if x==a22
                nova_prep(1,a22+1)=mat_prep(1,i);
                nova_prep(2,a22+1)=mat_prep(2,i);
                   a22=a22+1;
            end
      end
    x=0;
end

mat_prep=nova_prep;
[a1,a11] = size(mat_prep);
%-------------------------------------------

%k - koeficijenti smijera -> k=tg?

k1=(((py21(2)-py21(1))+0.0000001)/((px21(2)-px21(1))+0.0000001));
k2=(((py22(2)-py22(1))+0.0000001)/((px22(2)-px22(1))+0.0000001));
k3=(((py21(2)-py22(2))+0.0000001)/((px21(2)-px22(2))+0.0000001));

%-------------------------------------------
%b - odsječci na y osi

b1=py21(2)-k1*px21(2);
b2=py22(2)-k2*px22(2);
b3=py21(2)-k3*px21(2);

%-------------------------------------------
%eksplicitna jednadžba pravca y=kx+b
%računanje y koord za unaprijed zadanu x koord prepreke

for i=1:a11
    e=e+1;
```

```
   y_prep_p1(e)=k1*mat_prep(1,i)+b1;
   y_prep_p2(e)=k2*mat_prep(1,i)+b2;
   y_prep_p3(e)=k3*mat_prep(1,i)+b3;

   if mat_prep(2,i)< y_prep_p1(e) & mat_prep(2,i)> y_prep_p2(e) &
((mat_prep(2,i)>y_prep_p3(e)& px21(2)<px22(2))|(mat_prep(2,i)<y_prep_p3(e)&
px21(2)>px22(2))|(mat_prep(1,i)>px21(2) & px21(2)==px22(2)) )
        c=c+1;
        mat_prepreke(1,c)=mat_prep(1,i);
        mat_prepreke(2,c)=mat_prep(2,i);
   end
end


[a,b] = size(mat_prepreke);

%-----------------------------------------------
%definiranje nove matrice prepreka
nova_mat_prep(1:a,1)=p11;
nova_mat_prep(:,2:(b+1))=mat_prepreke;
nova_mat_prep(1:a,(b+2))=p22;
nova_mat_prep;

najbolji_pojedinac=[p11 p22];


[nova_prep_gore,nova_prep_dole,novo_najbolji_pojedinac]=isp_prepreke_r2(nova_mat_pr
ep,najbolji_pojedinac,duz_prostora,sir_prostora);
najbolji_pojedinac=novo_najbolji_pojedinac;

[b1,b11] = size(nova_prep_dole);
[c1,c11] = size(nova_prep_gore);

while nova_prep_gore~= 0

    if nova_prep_gore~=0

        nova_mat_prep=nova_prep_gore;
        najbolji_pojedinac=novo_najbolji_pojedinac;

[nova_prep_gore,novo_najbolji_pojedinac]=isp_prepreke_gore_r2(nova_mat_prep,najbolj
i_pojedinac,duz_prostora,sir_prostora);
        najbolji_pojedinac=novo_najbolji_pojedinac;

    else

        najbolji_pojedinac=novo_najbolji_pojedinac;

    end
end

while nova_prep_dole~=0

    if nova_prep_dole~=0
```

```
        nova_mat_prep=nova_prep_dole;
        najbolji_pojedinac=novo_najbolji_pojedinac;

[nova_prep_dole,novo_najbolji_pojedinac]=isp_prepreke_dole_r2(nova_mat_prep,najbolj
i_pojedinac,duz_prostora,sir_prostora);
        najbolji_pojedinac=novo_najbolji_pojedinac;

    else

        najbolji_pojedinac=novo_najbolji_pojedinac;

    end

end

%skraćenje putanje robota zbog promjera hvataljke

[e,f] = size(najbolji_pojedinac);

    k1=abs(atan((najbolji_pojedinac(2,1)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,1))));
    k2=(atan((najbolji_pojedinac(2,1)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,1))));

    naj_pojedinac(1,1)= najbolji_pojedinac(1,1)+(cos(k1)*r_h_2);
    naj_pojedinac(2,1)= najbolji_pojedinac(2,1)-(sin(k2)*r_h_2);

    toc_1_poc_2(1,1)=naj_pojedinac(1,1);
    toc_1_poc_2(1,2)=naj_pojedinac(2,1);
for i=2:1:(f-1)

        k21=abs(atan((najbolji_pojedinac(2,i)-
(sir_prostora/2))/(najbolji_pojedinac(1,i)-0)));
        k22=(atan((najbolji_pojedinac(2,i)-
(sir_prostora/2))/(najbolji_pojedinac(1,i)-0)));
        k11=abs(atan((najbolji_pojedinac(2,i)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,i))));
        k12=(atan((najbolji_pojedinac(2,i)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,i))));

        Xm=najbolji_pojedinac(1,i)+((cos(k21)*(r_h_2*f_h)));
        Ym=najbolji_pojedinac(2,i)+((sin(k22)*(r_h_2*f_h)));

        naj_pojedinac(1,i)= Xm+((cos(k11)*(r_h_2*f_h)));
        naj_pojedinac(2,i)= Ym-((sin(k12)*(r_h_2*f_h)));

end

    k1=abs(atan((najbolji_pojedinac(2,f)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,f))));
    k2=(atan((najbolji_pojedinac(2,f)-(sir_prostora/2))/(duz_prostora-
najbolji_pojedinac(1,f))));

    naj_pojedinac(1,f)= najbolji_pojedinac(1,f)+(cos(k1)*r_h_2);
    naj_pojedinac(2,f)= najbolji_pojedinac(2,f)-(sin(k2)*r_h_2);
```

```
    toc_1_kraj_2(1,1)=naj_pojedinac(1,f);
    toc_1_kraj_2(1,2)=naj_pojedinac(2,f);


najbolji_pojedinac=naj_pojedinac; %NE BRISATI


figure(13)
axis equal
title 'PUTANJA DRUGOG ROBOTA'
hold on
axis ([0 duz_prostora 0 sir_prostora]);

line(px21,py21,'Color','r','LineWidth',1);  %pravac p1 ROBOT 2
line(px22,py22,'Color','r','LineWidth',1);  %pravac p2 ROBOT 2


%---------------------------------------------
%definiranje i crtanje prepreka
for i=1:2:a21
    prx(1,1)=crt_mat_prep(1,i);
    pry(1,1)=crt_mat_prep(2,i);

    prx(1,2)=crt_mat_prep(1,i+1);
    pry(1,2)=crt_mat_prep(2,i+1);

line(prx,pry,'Color','k','LineWidth',2);
end

%---------------------------------------------
%crtanje putanje robota
for i=1:1:f-1
    tocx(1,1)= najbolji_pojedinac(1,i);
    tocy(1,1)= najbolji_pojedinac(2,i);

    tocx(1,2)= najbolji_pojedinac(1,i+1);
    tocy(1,2)= najbolji_pojedinac(2,i+1);

    line(tocx,tocy,'Color','m','LineWidth',2);
end
hold off



%---------------------------------------------
%provlačenje krivulje viseg reda po putanji robota
figure(14)
axis equal
title 'PUTANJA DRUGOG ROBOTA'
hold on
axis ([0 duz_prostora 0 sir_prostora]);

line(px21,py21,'Color','r','LineWidth',1);  %pravac p1 ROBOT 2
line(px22,py22,'Color','r','LineWidth',1);  %pravac p2 ROBOT 2
```

```
%-------------------------------------------
%definiranje i crtanje prepreka
for i=1:2:a21
    prx(1,1)=crt_mat_prep(1,i);
    pry(1,1)=crt_mat_prep(2,i);

    prx(1,2)=crt_mat_prep(1,i+1);
    pry(1,2)=crt_mat_prep(2,i+1);

line(prx,pry,'Color','k','LineWidth',2);
end

%-------------------------------------------
%crtanje najboljeg pojedinca
for i=1:f
    x(i)= najbolji_pojedinac(1,i);
    y(i)=najbolji_pojedinac(2,i);
end
t=1:f;
ts=1:1/100:f;
xs=spline(t,x,ts);
ys=spline(t,y,ts);
hold on
plot(xs,ys,'Color','b','LineWidth',2);
hold off
```

```
%ROBOT 1
%izračunavanje najkraće putanje robota između dvije točke bez prepreka

function [naj_pojedinac,toc_1_poc_1,toc_1_kraj_1] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz,sir,xpr,ypr,k11,k111,k12,k1
21,r_h_1,r_h_2)

a=0;
b=0;

%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće PRVI robot

px11=[0 (toc_1_poc_1(1)-(cos(k11)*r_h_1))]; %x koord početne i krajnje točke pravca
p1
py11=[(sir/2) (toc_1_poc_1(2)-(sin(k111)*r_h_1))]; %y koord početne i krajnje točke
pravca p1

px12=[0 (toc_1_kraj_1(1)-(cos(k12)*r_h_1))]; %x koord početne i krajnje točke
pravca p2
py12=[(sir/2) (toc_1_kraj_1(2)-(sin(k121)*r_h_1))]; %y koord početne i krajnje
točke pravca p2

px13=[px11(2) px12(2)];
py13=[py11(2) py12(2)];
```

```
toc_1_poc_1=[px11(2) py11(2)];
toc_1_kraj_1=[px12(2) py12(2)];
%-------------------------------------------
%crtanje radnog prosrtora prvog robota

figure(8)
axis equal
title 'PUTANJA PRVOG ROBOTA'
hold on
axis([0 duz 0 sir])

line(px11,py11,'Color','b','LineWidth',1);  %pravac p1 ROBOT 1
line(px12,py12,'Color','b','LineWidth',1);  %pravac p2 ROBOT 1


%-------------------------------------------
%definiranje prepreka u matričnom obliku

 mat_prepreka_r1=[xpr;ypr];


 %-------------------------------------------
 %crtanje prepreka

  [a,b]=size(mat_prepreka_r1);

      for i=1:2:b

        prx(1,1)=mat_prepreka_r1(1,i);
        pry(1,1)=mat_prepreka_r1(2,i);

        prx(1,2)=mat_prepreka_r1(1,i+1);
        pry(1,2)=mat_prepreka_r1(2,i+1);

        line(prx,pry,'Color','k','LineWidth',2);
      end

%-------------------------------------------
%definiranje najboljeg pojedinca

   naj_pojedinac=[px13;py13];


%-------------------------------------------
%crtanje najboljeg pojedinca

      line(px13,py13,'Color','m','LineWidth',2);
  hold off
```

```
%ROBOT 2
%izračunavanje najkraće putanje robota između dvije točke bez prepreka


function [naj_pojedinac,toc_1_poc_2,toc_1_kraj_2] =
f(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz,sir,xpr,ypr,k21,k211,k22,k2
21,r_h_1,r_h_2)

a=0;
b=0;

%-------------------------------------------------------------------------
%definiranje prostora u kojem se kreće DRUGI robot

px21=[duz (toc_1_poc_2(1)+(cos(k21)*r_h_2))]; %x koord početne i krajnje točke
pravca p1
py21=[(sir/2) (toc_1_poc_2(2)-(sin(k211)*r_h_2))];   %y koord početne i krajnje
točke pravca p1

px22=[duz (toc_1_kraj_2(1)+(cos(k22)*r_h_2))]; %x koord početne i krajnje točke
pravca p2
py22=[(sir/2) (toc_1_kraj_2(2)-(sin(k221)*r_h_2))];   %y koord početne i krajnje
točke pravca p2

px23=[px21(2) px22(2)];
py23=[py21(2) py22(2)];

toc_1_poc_2=[px21(2) py21(2)];
toc_1_kraj_2=[(px22(2)) py22(2)];
%-----------------------------------------
%crtanje radnog prosrtora drugog robota

figure(12)
axis equal
title 'PUTANJA DRUGOG ROBOTA'
hold on
axis([0 duz 0 sir])


 line(px21,py21,'Color','r','LineWidth',1);  %pravac p1 ROBOT 2
 line(px22,py22,'Color','r','LineWidth',1);  %pravac p2 ROBOT 2

%-----------------------------------------
%definiranje prepreka u matričnom obliku

 mat_prepreka_r2=[xpr;ypr];


 %-----------------------------------------
 %crtanje prepreka

  [a,b]=size(mat_prepreka_r2);
```

```
      for i=1:2:b

        prx(1,1)=mat_prepreka_r2(1,i);
        pry(1,1)=mat_prepreka_r2(2,i);

        prx(1,2)=mat_prepreka_r2(1,i+1);
        pry(1,2)=mat_prepreka_r2(2,i+1);

        line(prx,pry,'Color','k','LineWidth',2);
      end

%---------------------------------------------
%definiranje najboljeg pojedinca

   naj_pojedinac=[px23;py23];


%---------------------------------------------
%crtanje najboljeg pojedinca

      line(px23,py23,'Color','m','LineWidth',2);
 hold off
```

```
% pozivanje podprograma za pronalaženje najboljeg pojedinca
% za ROBOT 1 i ROBOT2
%-------------------------------------------------------------------------

function
[naj_pojedinac1,naj_pojedinac2,prep_sijeku1,prep_sijeku2,toc_1_kraj_1,toc_1_kraj_2]
=
f(t_1_x_poc_1,t_1_y_poc_1,t_1_x_kraj_1,t_1_y_kraj_1,t_1_x_poc_2,t_1_y_poc_2,t_1_x_k
raj_2,t_1_y_kraj_2,duz_prostora,sir_prostora,r_h_1,r_h_2,r_s_1,r_s_2,f_h)


b=0;c=0;d=0;sjec_1=0;sjec_2=0;sjec=0;
gibanje_R1=[0 0]; gibanje_R2=[0 0];

toc_1cr_poc_1=[t_1_x_poc_1, t_1_y_poc_1];
toc_1cr_kraj_1=[t_1_x_kraj_1, t_1_y_kraj_1];
toc_1cr_poc_2=[t_1_x_poc_2, t_1_y_poc_2];
toc_1cr_kraj_2=[t_1_x_kraj_2, t_1_y_kraj_2];

%-------------------------------------------------------------------------

k11=abs(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));
k111=(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));

k12=abs(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));
k121=(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));
```

```
k21=abs(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));
k211=(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));

k22=abs(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));
k221=(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));

if k111>k121
    gibanje_R1=[1 0]; %robot R1 se giba prem dolje
end

if k111<k121
    gibanje_R1=[0 1]; %robot R1 se giba prem gore
    t_1n_x_poc_1=t_1_x_kraj_1;
    t_1n_y_poc_1=t_1_y_kraj_1;
    t_1n_x_kraj_1=t_1_x_poc_1;
    t_1n_y_kraj_1=t_1_y_poc_1;
    t_1_x_poc_1=t_1n_x_poc_1;
    t_1_y_poc_1=t_1n_y_poc_1;
    t_1_x_kraj_1=t_1n_x_kraj_1;
    t_1_y_kraj_1=t_1n_y_kraj_1;
end

if k211>k221
    gibanje_R2=[1 0]; %robot R2 se giba prem dolje
end

if k211<k221
    gibanje_R2=[0 1]; %robot R2 se giba prem gore
    t_1n_x_poc_2=t_1_x_kraj_2;
    t_1n_y_poc_2=t_1_y_kraj_2;
    t_1n_x_kraj_2=t_1_x_poc_2;
    t_1n_y_kraj_2=t_1_y_poc_2;
    t_1_x_poc_2=t_1n_x_poc_2;
    t_1_y_poc_2=t_1n_y_poc_2;
    t_1_x_kraj_2=t_1n_x_kraj_2;
    t_1_y_kraj_2=t_1n_y_kraj_2;
end

k11=abs(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));
k111=(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));

k12=abs(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));
k121=(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));

k21=abs(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));
k211=(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));

k22=abs(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));
k221=(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));

%Koordinata početne i krajnje točke robota s povećanim radijusom
toc_1_poc_1 = [(t_1_x_poc_1+(cos(k11)*r_h_1)), (t_1_y_poc_1+(sin(k111)*r_h_1))]; %
definicija početne tocke prvog robota
toc_1_kraj_1 = [(t_1_x_kraj_1+(cos(k12)*r_h_1)), (t_1_y_kraj_1+(sin(k121)*r_h_1))];
% definicija krajnje tocke prvog robota
```

```
toc_1_poc_2 = [(t_1_x_poc_2-(cos(k21)*r_h_2)), (t_1_y_poc_2+(sin(k211)*r_h_2))]; %
definicija početne tocke drugog robota
toc_1_kraj_2 = [(t_1_x_kraj_2-(cos(k22)*r_h_2)), (t_1_y_kraj_2+(sin(k221)*r_h_2))];
% definicija krajnje tocke grugog robota
%-------------------------------------------------------------------------
toc_1h_poc_1 = [(t_1_x_poc_1-(cos(k11)*r_h_1)), (t_1_y_poc_1-(sin(k111)*r_h_1))]; %
definicija početne tocke prvog robota
toc_1h_kraj_1 = [(t_1_x_kraj_1-(cos(k12)*r_h_1)), (t_1_y_kraj_1-
(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota
toc_1h_poc_2 = [(t_1_x_poc_2+(cos(k21)*r_h_2)), (t_1_y_poc_2-(sin(k211)*r_h_2))]; %
definicija početne tocke drugog robota
toc_1h_kraj_2 = [(t_1_x_kraj_2+(cos(k22)*r_h_2)), (t_1_y_kraj_2-
(sin(k221)*r_h_2))];  % definicija krajnje tocke grugog robota
%-------------------------------------------------------------------------
%definiranje prepreka u kojem se kreće robot s povećanim radijusom

%za prvi robot; KONFIGURACIJA I
xpr11=[       0                              (t_1_x_poc_1-(sin(k111)*r_s_1))
0                 (t_1_x_poc_1+(sin(k111)*r_s_1))  (toc_1h_poc_1(1)-
(sin(k111)*r_h_1)) (toc_1_poc_1(1)-(sin(k111)*r_h_1))
(toc_1h_poc_1(1)+(sin(k111)*r_h_1)) (toc_1_poc_1(1)+(sin(k111)*r_h_1))
(toc_1_poc_1(1)-(sin(k111)*r_h_1)) (toc_1_poc_1(1)+(sin(k111)*r_h_1))];
ypr11=[((sir_prostora/2)+(r_s_1/cos(k11))) (t_1_y_poc_1+(cos(k11)*r_s_1))
((sir_prostora/2)-(r_s_1/cos(k11))) (t_1_y_poc_1-(cos(k11)*r_s_1))
(toc_1h_poc_1(2)+(cos(k11)*r_h_1))  (toc_1_poc_1(2)+(cos(k11)*r_h_1))
(toc_1h_poc_1(2)-(cos(k11)*r_h_1))  (toc_1_poc_1(2)-(cos(k11)*r_h_1))
(toc_1_poc_1(2)+(cos(k11)*r_h_1))  (toc_1_poc_1(2)-(cos(k11)*r_h_1))];

%za prvi robot; KONFIGURACIJA II
xpr12=[        0                              (t_1_x_kraj_1-(sin(k121)*r_s_1))
0                (t_1_x_kraj_1+(sin(k121)*r_s_1))  (toc_1h_kraj_1(1)-
(sin(k121)*r_h_1)) (toc_1_kraj_1(1)-(sin(k121)*r_h_1))
(toc_1h_kraj_1(1)+(sin(k121)*r_h_1)) (toc_1_kraj_1(1)+(sin(k121)*r_h_1))
(toc_1_kraj_1(1)-(sin(k121)*r_h_1)) (toc_1_kraj_1(1)+(sin(k121)*r_h_1))];
ypr12=[((sir_prostora/2)+(r_s_1/cos(k12))) (t_1_y_kraj_1+(cos(k12)*r_s_1))
((sir_prostora/2)-(r_s_1/cos(k12))) (t_1_y_kraj_1-(cos(k12)*r_s_1))
(toc_1h_kraj_1(2)+(cos(k12)*r_h_1))  (toc_1_kraj_1(2)+(cos(k12)*r_h_1))
(toc_1h_kraj_1(2)-(cos(k12)*r_h_1))  (toc_1_kraj_1(2)-(cos(k12)*r_h_1))
(toc_1_kraj_1(2)+(cos(k12)*r_h_1))  (toc_1_kraj_1(2)-(cos(k12)*r_h_1))];
%-------------------------------------------------------------------------

%za drui robot; KONFIGURACIJA I
xpr21=[     duz_prostora                    (t_1_x_poc_2+(sin(k211)*r_s_2))
duz_prostora        (t_1_x_poc_2-(sin(k211)*r_s_2))
(toc_1h_poc_2(1)+(sin(k211)*r_h_2)) (toc_1_poc_2(1)+(sin(k211)*r_h_2))
(toc_1h_poc_2(1)-(sin(k211)*r_h_2)) (toc_1_poc_2(1)-(sin(k211)*r_h_2))
(toc_1_poc_2(1)+(sin(k211)*r_h_2)) (toc_1_poc_2(1)-(sin(k211)*r_h_2))];
ypr21=[((sir_prostora/2)+(r_s_2/cos(k21)))  (t_1_y_poc_2+(cos(k21)*r_s_2))
((sir_prostora/2)-(r_s_2/cos(k21)))  (t_1_y_poc_2-(cos(k21)*r_s_2))
(toc_1h_poc_2(2)+(cos(k21)*r_h_2))  (toc_1_poc_2(2)+(cos(k21)*r_h_2))
(toc_1h_poc_2(2)-(cos(k21)*r_h_2))  (toc_1_poc_2(2)-(cos(k21)*r_h_2))
(toc_1_poc_2(2)+(cos(k21)*r_h_2))  (toc_1_poc_2(2)-(cos(k21)*r_h_2))];

%za drui robot; KONFIGURACIJA II
xpr22=[          duz_prostora              (t_1_x_kraj_2+(sin(k221)*r_s_2))
duz_prostora       (t_1_x_kraj_2-(sin(k221)*r_s_2))
(toc_1h_kraj_2(1)+(sin(k221)*r_h_2)) (toc_1_kraj_2(1)+(sin(k221)*r_h_2))
```

```
(toc_1h_kraj_2(1)-(sin(k221)*r_h_2)) (toc_1_kraj_2(1)-(sin(k221)*r_h_2))
(toc_1_kraj_2(1)+(sin(k221)*r_h_2)) (toc_1_kraj_2(1)-(sin(k221)*r_h_2))];
ypr22=[((sir_prostora/2)+(r_s_2/cos(k22))) (t_1_y_kraj_2+(cos(k22)*r_s_2))
(sir_prostora/2)-(r_s_2/cos(k22))    (t_1_y_kraj_2-(cos(k22)*r_s_2))
(toc_1h_kraj_2(2)+(cos(k22)*r_h_2))   (toc_1_kraj_2(2)+(cos(k22)*r_h_2))
(toc_1h_kraj_2(2)-(cos(k22)*r_h_2))   (toc_1_kraj_2(2)-(cos(k22)*r_h_2))
(toc_1_kraj_2(2)+(cos(k22)*r_h_2))   (toc_1_kraj_2(2)-(cos(k22)*r_h_2))];
%-------------------------------------------------------------------------

%---------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---------------------------------
 xpr=[300 600];  %x koord početne i krajnje točke prepreke
 ypr=[1500 1550]; %y koord početne i krajnje točke prepreke

 [c,d] = size(xpr);
%---------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---------------------------------
x1=0;x2=0;x3=0;x4=0;x5=0;x6=0;x7=0;x8=0;

%Ispitivanje podrucja u kojem se nalaze roboti za KONFIGURACIJU I

[mat_prepreka_r11,br_prep_r11,mat_prepreka_r21,br_prep_r21,sjec_11,sjec_21,prep_sij
ece_dole11,prep_sijece_dole21,prep_sijece_gore11,prep_sijece_gore21,toc_1xy_poc_1,t
oc_1xy_poc_2,toc_1xy_kraj_1,toc_1xy_kraj_2] =
isp_radnog_podrucja_kon_I(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,xpr11,y
pr11,xpr21,ypr21,duz_prostora,sir_prostora,r_h_1,r_h_2,f_h,gibanje_R1,gibanje_R2)

%robot1

if (prep_sijece_dole11>0 & prep_sijece_gore11==0 & gibanje_R1(2)==1 &
gibanje_R2(2)==1)
 x1=1
   t_1_x_poc_2=toc_1xy_poc_2(1);
   t_1_y_poc_2=toc_1xy_poc_2(2);

   k21=abs(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));
   k211=(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));

   toc_1_poc_2 = [(t_1_x_poc_2-(cos(k21)*r_h_2)), (t_1_y_poc_2+(sin(k211)*r_h_2))];
% definicija početne tocke drugog robota
   toc_1h_poc_2 = [(t_1_x_poc_2+(cos(k21)*r_h_2)), (t_1_y_poc_2-
(sin(k211)*r_h_2))]; % definicija početne tocke drugog robota

   xpr21=[     duz_prostora                    (t_1_x_poc_2+(sin(k211)*r_s_2))
duz_prostora         (t_1_x_poc_2-(sin(k211)*r_s_2))
(toc_1h_poc_2(1)+(sin(k211)*r_h_2)) (toc_1_poc_2(1)+(sin(k211)*r_h_2))
(toc_1h_poc_2(1)-(sin(k211)*r_h_2)) (toc_1_poc_2(1)-(sin(k211)*r_h_2))
(toc_1_poc_2(1)+(sin(k211)*r_h_2)) (toc_1_poc_2(1)-(sin(k211)*r_h_2))];
   ypr21=[((sir_prostora/2)+(r_s_2/cos(k21)))  (t_1_y_poc_2+(cos(k21)*r_s_2))
((sir_prostora/2)-(r_s_2/cos(k21)))  (t_1_y_poc_2-(cos(k21)*r_s_2))
(toc_1h_poc_2(2)+(cos(k21)*r_h_2))  (toc_1_poc_2(2)+(cos(k21)*r_h_2))
(toc_1h_poc_2(2)-(cos(k21)*r_h_2))  (toc_1_poc_2(2)-(cos(k21)*r_h_2))
(toc_1_poc_2(2)+(cos(k21)*r_h_2))  (toc_1_poc_2(2)-(cos(k21)*r_h_2))];
```

```
   br_prep_r11=0;
   br_prep_r21=0;
end

if (prep_sijece_dole11>0 & prep_sijece_gore11==0 & gibanje_R1(1)==1 &
gibanje_R2(1)==1)
 x3=1

    t_1_x_kraj_1=toc_1xy_kraj_1(1);
    t_1_y_kraj_1=toc_1xy_kraj_1(2);

    k12=abs(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));
    k121=(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));

    toc_1_kraj_1 = [(t_1_x_kraj_1+(cos(k12)*r_h_1)),
(t_1_y_kraj_1+(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota
    toc_1h_kraj_1 = [(t_1_x_kraj_1-(cos(k12)*r_h_1)), (t_1_y_kraj_1-
(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota

    toc_1_kraj_1 = [(t_1_x_kraj_1+(cos(k12)*r_h_1)),
(t_1_y_kraj_1+(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota
    toc_1h_kraj_1 = [(t_1_x_kraj_1-(cos(k12)*r_h_1)), (t_1_y_kraj_1-
(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota
    xpr12=[        0                              (t_1_x_kraj_1-(sin(k121)*r_s_1))
0                (t_1_x_kraj_1+(sin(k121)*r_s_1))  (toc_1h_kraj_1(1)-
(sin(k121)*r_h_1)) (toc_1_kraj_1(1)-(sin(k121)*r_h_1))
(toc_1h_kraj_1(1)+(sin(k121)*r_h_1)) (toc_1_kraj_1(1)+(sin(k121)*r_h_1))
(toc_1_kraj_1(1)-(sin(k121)*r_h_1)) (toc_1_kraj_1(1)+(sin(k121)*r_h_1))];
    ypr12=[((sir_prostora/2)+(r_s_1/cos(k12))) (t_1_y_kraj_1+(cos(k12)*r_s_1))
((sir_prostora/2)-(r_s_1/cos(k12))) (t_1_y_kraj_1-(cos(k12)*r_s_1))
(toc_1h_kraj_1(2)+(cos(k12)*r_h_1))  (toc_1_kraj_1(2)+(cos(k12)*r_h_1))
(toc_1h_kraj_1(2)-(cos(k12)*r_h_1))  (toc_1_kraj_1(2)-(cos(k12)*r_h_1))
(toc_1_kraj_1(2)+(cos(k12)*r_h_1))  (toc_1_kraj_1(2)-(cos(k12)*r_h_1))];

   br_prep_r11=0;
   br_prep_r21=0;
end

if (prep_sijece_dole11==0 & prep_sijece_gore11>0 & gibanje_R1(1)==1 &
gibanje_R2(2)==1)
 x2=1
   t_1_x_poc_2=toc_1xy_poc_2(1);
   t_1_y_poc_2=toc_1xy_poc_2(2);

   k21=abs(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));
   k211=(atan((t_1_y_poc_2-(sir_prostora/2))/(duz_prostora-t_1_x_poc_2)));

   toc_1_poc_2 = [(t_1_x_poc_2-(cos(k21)*r_h_2)), (t_1_y_poc_2+(sin(k211)*r_h_2))];
% definicija početne tocke drugog robota
   toc_1h_poc_2 = [(t_1_x_poc_2+(cos(k21)*r_h_2)), (t_1_y_poc_2-
(sin(k211)*r_h_2))]; % definicija početne tocke drugog robota

   xpr21=[     duz_prostora                    (t_1_x_poc_2+(sin(k211)*r_s_2))
duz_prostora         (t_1_x_poc_2-(sin(k211)*r_s_2))
(toc_1h_poc_2(1)+(sin(k211)*r_h_2)) (toc_1_poc_2(1)+(sin(k211)*r_h_2))
```

```
(toc_1h_poc_2(1)-(sin(k211)*r_h_2)) (toc_1_poc_2(1)-(sin(k211)*r_h_2))
(toc_1_poc_2(1)+(sin(k211)*r_h_2)) (toc_1_poc_2(1)-(sin(k211)*r_h_2))];
   ypr21=[((sir_prostora/2)+(r_s_2/cos(k21)))  (t_1_y_poc_2+(cos(k21)*r_s_2))
((sir_prostora/2)-(r_s_2/cos(k21)))  (t_1_y_poc_2-(cos(k21)*r_s_2))
(toc_1h_poc_2(2)+(cos(k21)*r_h_2))  (toc_1_poc_2(2)+(cos(k21)*r_h_2))
(toc_1h_poc_2(2)-(cos(k21)*r_h_2))  (toc_1_poc_2(2)-(cos(k21)*r_h_2))
(toc_1_poc_2(2)+(cos(k21)*r_h_2))  (toc_1_poc_2(2)-(cos(k21)*r_h_2))];

   br_prep_r11=0;
   br_prep_r21=0;
end

if (prep_sijece_dole11==0 & prep_sijece_gore11>0 & gibanje_R1(2)==1 &
gibanje_R2(1)==1)
 x4=1
   t_1_x_poc_1=toc_1xy_poc_1(1);
   t_1_y_poc_1=toc_1xy_poc_1(2);

   k11=abs(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));
   k111=(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));

   toc_1_poc_1 = [(t_1_x_poc_1+(cos(k11)*r_h_1)), (t_1_y_poc_1+(sin(k111)*r_h_1))];
% definicija početne tocke prvog robota
   toc_1h_poc_1 = [(t_1_x_poc_1-(cos(k11)*r_h_1)), (t_1_y_poc_1-
(sin(k111)*r_h_1))]; % definicija početne tocke prvog robota

   xpr11=[       0                              (t_1_x_poc_1-(sin(k111)*r_s_1))
0                 (t_1_x_poc_1+(sin(k111)*r_s_1))  (toc_1h_poc_1(1)-
(sin(k111)*r_h_1)) (toc_1_poc_1(1)-(sin(k111)*r_h_1))
(toc_1h_poc_1(1)+(sin(k111)*r_h_1)) (toc_1_poc_1(1)+(sin(k111)*r_h_1))
(toc_1_poc_1(1)-(sin(k111)*r_h_1)) (toc_1_poc_1(1)+(sin(k111)*r_h_1))];
   ypr11=[((sir_prostora/2)+(r_s_1/cos(k11))) (t_1_y_poc_1+(cos(k11)*r_s_1))
((sir_prostora/2)-(r_s_1/cos(k11))) (t_1_y_poc_1-(cos(k11)*r_s_1))
(toc_1h_poc_1(2)+(cos(k11)*r_h_1))  (toc_1_poc_1(2)+(cos(k11)*r_h_1))
(toc_1h_poc_1(2)-(cos(k11)*r_h_1))  (toc_1_poc_1(2)-(cos(k11)*r_h_1))
(toc_1_poc_1(2)+(cos(k11)*r_h_1))  (toc_1_poc_1(2)-(cos(k11)*r_h_1))];

   br_prep_r11=0;
   br_prep_r21=0;
end

%---------------------------------
%Ispitivanje podrucja u kojem se nalaze roboti za KONFIGURACIJU II

[mat_prepreka_r12,br_prep_r12,mat_prepreka_r22,br_prep_r22,sjec_12,sjec_22,prep_sij
ece_gore12,prep_sijece_gore22,prep_sijece_dole12,prep_sijece_dole22,toc_1xy_poc_1,t
oc_1xy_poc_2,toc_1xy_kraj_1,toc_1xy_kraj_2] =
isp_radnog_podrucja_kon_II(toc_1_poc_1,toc_1_kraj_1,toc_1_poc_2,toc_1_kraj_2,xpr12,
ypr12,xpr22,ypr22,duz_prostora,sir_prostora,r_h_1,r_h_2,f_h,gibanje_R1,gibanje_R2)
%robot1

if (prep_sijece_dole12==0 & prep_sijece_gore12>0 & gibanje_R1(1)==1 &
gibanje_R2(1)==1)
x5=1
   t_1_x_kraj_2=toc_1xy_kraj_2(1);
   t_1_y_kraj_2=toc_1xy_kraj_2(2);
```

```
   k22=abs(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));
   k221=(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));

   toc_1_kraj_2 = [(t_1_x_kraj_2-(cos(k22)*r_h_2)),
(t_1_y_kraj_2+(sin(k221)*r_h_2))];  % definicija krajnje tocke grugog robota
   toc_1h_kraj_2 = [(t_1_x_kraj_2+(cos(k22)*r_h_2)), (t_1_y_kraj_2-
(sin(k221)*r_h_2))];  % definicija krajnje tocke grugog robota

   xpr22=[          duz_prostora              (t_1_x_kraj_2+(sin(k221)*r_s_2))
duz_prostora       (t_1_x_kraj_2-(sin(k221)*r_s_2))
(toc_1h_kraj_2(1)+(sin(k221)*r_h_2)) (toc_1_kraj_2(1)+(sin(k221)*r_h_2))
(toc_1h_kraj_2(1)-(sin(k221)*r_h_2)) (toc_1_kraj_2(1)-(sin(k221)*r_h_2))
(toc_1_kraj_2(1)+(sin(k221)*r_h_2)) (toc_1_kraj_2(1)-(sin(k221)*r_h_2))];
   ypr22=[((sir_prostora/2)+(r_s_2/cos(k22))) (t_1_y_kraj_2+(cos(k22)*r_s_2))
(sir_prostora/2)-(r_s_2/cos(k22))   (t_1_y_kraj_2-(cos(k22)*r_s_2))
(toc_1h_kraj_2(2)+(cos(k22)*r_h_2))  (toc_1_kraj_2(2)+(cos(k22)*r_h_2))
(toc_1h_kraj_2(2)-(cos(k22)*r_h_2))  (toc_1_kraj_2(2)-(cos(k22)*r_h_2))
(toc_1_kraj_2(2)+(cos(k22)*r_h_2))  (toc_1_kraj_2(2)-(cos(k22)*r_h_2))];

   br_prep_r12=0;
   br_prep_r22=0;
end

if (prep_sijece_dole12==0 & prep_sijece_gore12>0 & gibanje_R1(2)==1 &
gibanje_R2(2)==1)
x6=1
   t_1_x_poc_1=toc_1xy_poc_1(1);
   t_1_y_poc_1=toc_1xy_poc_1(2);

   k11=abs(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));
   k111=(atan((t_1_y_poc_1-(sir_prostora/2))/(t_1_x_poc_1-0)));

   toc_1_poc_1 = [(t_1_x_poc_1+(cos(k11)*r_h_1)), (t_1_y_poc_1+(sin(k111)*r_h_1))];
% definicija početne tocke prvog robota
   toc_1h_poc_1 = [(t_1_x_poc_1-(cos(k11)*r_h_1)), (t_1_y_poc_1-
(sin(k111)*r_h_1))]; % definicija početne tocke prvog robota

   xpr11=[       0                           (t_1_x_poc_1-(sin(k111)*r_s_1))
0                 (t_1_x_poc_1+(sin(k111)*r_s_1))  (toc_1h_poc_1(1)-
(sin(k111)*r_h_1)) (toc_1_poc_1(1)-(sin(k111)*r_h_1))
(toc_1h_poc_1(1)+(sin(k111)*r_h_1)) (toc_1_poc_1(1)+(sin(k111)*r_h_1))
(toc_1_poc_1(1)-(sin(k111)*r_h_1)) (toc_1_poc_1(1)+(sin(k111)*r_h_1))];
   ypr11=[((sir_prostora/2)+(r_s_1/cos(k11))) (t_1_y_poc_1+(cos(k11)*r_s_1))
((sir_prostora/2)-(r_s_1/cos(k11))) (t_1_y_poc_1-(cos(k11)*r_s_1))
(toc_1h_poc_1(2)+(cos(k11)*r_h_1))  (toc_1_poc_1(2)+(cos(k11)*r_h_1))
(toc_1h_poc_1(2)-(cos(k11)*r_h_1))  (toc_1_poc_1(2)-(cos(k11)*r_h_1))
(toc_1_poc_1(2)+(cos(k11)*r_h_1))  (toc_1_poc_1(2)-(cos(k11)*r_h_1))];

   br_prep_r12=0;
   br_prep_r22=0;
end

if (prep_sijece_dole12>0 & prep_sijece_gore12==0 & gibanje_R1(1)==1 &
gibanje_R2(2)==1)
x7=1
```

```
   t_1_x_kraj_1=toc_1xy_kraj_1(1);
   t_1_y_kraj_1=toc_1xy_kraj_1(2);

   k12=abs(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));
   k121=(atan((t_1_y_kraj_1-(sir_prostora/2))/(t_1_x_kraj_1-0)));

   toc_1_kraj_1 = [(t_1_x_kraj_1+(cos(k12)*r_h_1)),
(t_1_y_kraj_1+(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota
   toc_1h_kraj_1 = [(t_1_x_kraj_1-(cos(k12)*r_h_1)), (t_1_y_kraj_1-
(sin(k121)*r_h_1))]; % definicija krajnje tocke prvog robota

   xpr12=[        0                          (t_1_x_kraj_1-(sin(k121)*r_s_1))
0               (t_1_x_kraj_1+(sin(k121)*r_s_1))  (toc_1h_kraj_1(1)-
(sin(k121)*r_h_1)) (toc_1_kraj_1(1)-(sin(k121)*r_h_1))
(toc_1h_kraj_1(1)+(sin(k121)*r_h_1)) (toc_1_kraj_1(1)+(sin(k121)*r_h_1))
(toc_1_kraj_1(1)-(sin(k121)*r_h_1)) (toc_1_kraj_1(1)+(sin(k121)*r_h_1))];
   ypr12=[((sir_prostora/2)+(r_s_1/cos(k12))) (t_1_y_kraj_1+(cos(k12)*r_s_1))
((sir_prostora/2)-(r_s_1/cos(k12))) (t_1_y_kraj_1-(cos(k12)*r_s_1))
(toc_1h_kraj_1(2)+(cos(k12)*r_h_1))  (toc_1_kraj_1(2)+(cos(k12)*r_h_1))
(toc_1h_kraj_1(2)-(cos(k12)*r_h_1))  (toc_1_kraj_1(2)-(cos(k12)*r_h_1))
(toc_1_kraj_1(2)+(cos(k12)*r_h_1))  (toc_1_kraj_1(2)-(cos(k12)*r_h_1))];

   br_prep_r12=0;
   br_prep_r22=0;
end

if (prep_sijece_dole12>0 & prep_sijece_gore12==0 & gibanje_R1(2)==1 &
gibanje_R2(1)==1)
x8=1
   t_1_x_kraj_2=toc_1xy_kraj_2(1);
   t_1_y_kraj_2=toc_1xy_kraj_2(2);

   k22=abs(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));
   k221=(atan((t_1_y_kraj_2-(sir_prostora/2))/(duz_prostora-t_1_x_kraj_2)));

   toc_1_kraj_2 = [(t_1_x_kraj_2-(cos(k22)*r_h_2)),
(t_1_y_kraj_2+(sin(k221)*r_h_2))];  % definicija krajnje tocke grugog robota
   toc_1h_kraj_2 = [(t_1_x_kraj_2+(cos(k22)*r_h_2)), (t_1_y_kraj_2-
(sin(k221)*r_h_2))];  % definicija krajnje tocke grugog robota

   xpr22=[          duz_prostora               (t_1_x_kraj_2+(sin(k221)*r_s_2))
duz_prostora       (t_1_x_kraj_2-(sin(k221)*r_s_2))
(toc_1h_kraj_2(1)+(sin(k221)*r_h_2)) (toc_1_kraj_2(1)+(sin(k221)*r_h_2))
(toc_1h_kraj_2(1)-(sin(k221)*r_h_2)) (toc_1_kraj_2(1)-(sin(k221)*r_h_2))
(toc_1_kraj_2(1)+(sin(k221)*r_h_2)) (toc_1_kraj_2(1)-(sin(k221)*r_h_2))];
   ypr22=[((sir_prostora/2)+(r_s_2/cos(k22))) (t_1_y_kraj_2+(cos(k22)*r_s_2))
(sir_prostora/2)-(r_s_2/cos(k22))   (t_1_y_kraj_2-(cos(k22)*r_s_2))
(toc_1h_kraj_2(2)+(cos(k22)*r_h_2))  (toc_1_kraj_2(2)+(cos(k22)*r_h_2))
(toc_1h_kraj_2(2)-(cos(k22)*r_h_2))  (toc_1_kraj_2(2)-(cos(k22)*r_h_2))
(toc_1_kraj_2(2)+(cos(k22)*r_h_2))  (toc_1_kraj_2(2)-(cos(k22)*r_h_2))];

   br_prep_r12=0;
   br_prep_r22=0;
end
```

```
% ---------------------------------

sjec=sjec_11+sjec_21+sjec_12+sjec_22

if sjec==0

% ---------------------------------
% ISPITIVANJE PREPREKA U RADNOM PODRU?JU PRVOG ROBOTA

br_prepreka_1=(d/2)+br_prep_r11+br_prep_r12;

if (br_prep_r11+br_prep_r12)==0

[prep_sijece_oba1,prep_sijece_gore1,prep_sijece_dole1,br_prep_izvan_1,br_prep_unuta
r_1,mat_prep_unutar_1] =
f_isp_prep_r1(toc_1_poc_1,toc_1_kraj_1,xpr,ypr,br_prepreka_1,duz_prostora,sir_prost
ora);

else
  xpr1=xpr;
  ypr1=ypr;

    for i=1:br_prep_r11*2
      xpr1(d+i)=mat_prepreka_r11(1,i);
      ypr1(d+i)=mat_prepreka_r11(2,i);
    end

    [c,d] = size(xpr1);

    for i=1:br_prep_r12*2
      xpr1(d+i)=mat_prepreka_r12(1,i);
      ypr1(d+i)=mat_prepreka_r12(2,i);
    end


[prep_sijece_oba1,prep_sijece_gore1,prep_sijece_dole1,br_prep_izvan_1,br_prep_unuta
r_1,mat_prep_unutar_1] =
f_isp_prep_r1(toc_1_poc_1,toc_1_kraj_1,xpr1,ypr1,br_prepreka_1,duz_prostora,sir_pro
stora);
end

br_prep_sijeku_1=prep_sijece_oba1+prep_sijece_gore1+prep_sijece_dole1;

 if br_prep_sijeku_1>0
    disp('Nije moguće izvesti zadatak! Krivo definiran zadatak.');
     br_prep_sijeku_1;
     prep_sijeku1= br_prep_sijeku_1;
     naj_pojedinac1=[0 0 ; 0 0];

 elseif br_prep_unutar_1==0


[naj_pojed_1,toc_1_poc_1,toc_1_kraj_1]=najbolji_pojedinac_r1(toc_1_poc_1,toc_1_kraj
_1,toc_1_poc_2,toc_1_kraj_2,duz_prostora,sir_prostora,xpr,ypr,k11,k111,k12,k121,r_h
_1,r_h_2);
```

```
      naj_pojed_1;
      br_prep_izvan_1;
      naj_pojedinac1=naj_pojed_1;
      prep_sijeku1=0;
      toc_1cr_poc_1=toc_1_poc_1;
      toc_1cr_kraj_1=toc_1_kraj_1;
 else

     disp('Kod za zaobilaženje prepreka!');
     br_prep_unutar_1;
     mat_prep_unutar_1;


[naj_pojed_1,toc_1_poc_1,toc_1_kraj_1]=ispitivanje_prepreka_gen_r1(toc_1_poc_1,toc_
1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz_prostora,sir_prostora,mat_prep_unutar_1,r_h_1
,f_h);
     naj_pojedinac1=naj_pojed_1;
     prep_sijeku1=0;
     toc_1cr_poc_1=toc_1_poc_1;
     toc_1cr_kraj_1=toc_1_kraj_1;
  end

%---------------------------------
%ISPITIVANJE PREPREKA U RADNOM PODRU?JU DRUGOG ROBOTA

br_prepreka_2=(d/2)+br_prep_r21+br_prep_r22;

if (br_prep_r21+br_prep_r22)==0

[prep_sijece_oba2,prep_sijece_gore2,prep_sijece_dole2,br_prep_izvan_2,br_prep_unuta
r_2,mat_prep_unutar_2] =
f_isp_prep_r2(toc_1_poc_2,toc_1_kraj_2,xpr,ypr,br_prepreka_2,duz_prostora,sir_prost
ora);

else
  xpr2=xpr;
  ypr2=ypr;

    for i=1:br_prep_r21*2
      xpr2(d+i)=mat_prepreka_r21(1,i);
      ypr2(d+i)=mat_prepreka_r21(2,i);
    end

    [c,d] = size(xpr2);

    for i=1:br_prep_r22*2
      xpr2(d+i)=mat_prepreka_r22(1,i);
      ypr2(d+i)=mat_prepreka_r22(2,i);
    end


[prep_sijece_oba2,prep_sijece_gore2,prep_sijece_dole2,br_prep_izvan_2,br_prep_unuta
r_2,mat_prep_unutar_2] =
f_isp_prep_r2(toc_1_poc_2,toc_1_kraj_2,xpr2,ypr2,br_prepreka_2,duz_prostora,sir_pro
stora);
end
```

```
 br_prep_sijeku_2=prep_sijece_oba2+prep_sijece_gore2+prep_sijece_dole2;

 if br_prep_sijeku_2>0
    disp('Nije moguće izvesti zadatak! Krivo definiran zadatak.');
    br_prep_sijeku_2;
    prep_sijeku2= br_prep_sijeku_2;
    naj_pojedinac2=[0 0 ; 0 0];

 elseif br_prep_unutar_2==0

[naj_pojed_2,toc_1_poc_2,toc_1_kraj_2]=najbolji_pojedinac_r2(toc_1_poc_1,toc_1_kraj
_1,toc_1_poc_2,toc_1_kraj_2,duz_prostora,sir_prostora,xpr,ypr,k21,k211,k22,k221,r_h
_1,r_h_2);
    naj_pojed_2;
    br_prep_izvan_2;
    naj_pojedinac2=naj_pojed_2;
    prep_sijeku2=0;
    toc_1cr_poc_2=toc_1_poc_2
    toc_1cr_kraj_2=toc_1_kraj_2
 else
    disp('Kod za zaobilaženje prepreka!');
    br_prep_unutar_2;
    mat_prep_unutar_2;

[naj_pojed_2,toc_1_poc_2,toc_1_kraj_2]=ispitivanje_prepreka_gen_r2(toc_1_poc_1,toc_
1_kraj_1,toc_1_poc_2,toc_1_kraj_2,duz_prostora,sir_prostora,mat_prep_unutar_2,r_h_2
,f_h);

    naj_pojedinac2=naj_pojed_2;
    prep_sijeku2=0;
    toc_1cr_poc_2=toc_1_poc_2;
    toc_1cr_kraj_2=toc_1_kraj_2;
 end
%-------------------------------------
%-------------------------------------
if gibanje_R1(2)==1
    t_1n_x_poc_1=t_1_x_kraj_1;
    t_1n_y_poc_1=t_1_y_kraj_1;
    t_1n_x_kraj_1=t_1_x_poc_1;
    t_1n_y_kraj_1=t_1_y_poc_1;
    toc_1_poc_1=[t_1n_x_poc_1,t_1n_y_poc_1];
    toc_1_kraj_1=[t_1n_x_kraj_1,t_1n_y_kraj_1];
    toc_1cr_poc_1=[t_1n_x_poc_1,t_1n_y_poc_1];
    toc_1cr_kraj_1=[t_1n_x_kraj_1,t_1n_y_kraj_1];
    [c,d] = size(naj_pojed_1);
    g=d;
   for i=1:1:(d)

        naj_pojedinac1(1,g)=naj_pojed_1(1,i);
        naj_pojedinac1(2,g)=naj_pojed_1(2,i);
    g=g-1;
   end
   naj_pojed_1=naj_pojedinac1;
end

if gibanje_R2(2)==1
```

```
    t_1n_x_poc_2=toc_1_kraj_2(1);
    t_1n_y_poc_2=toc_1_kraj_2(2);
    t_1n_x_kraj_2=toc_1_poc_2(1);
    t_1n_y_kraj_2=toc_1_poc_2(2);
    toc_1_poc_2=[t_1n_x_poc_2,t_1n_y_poc_2];
    toc_1_kraj_2=[t_1n_x_kraj_2,t_1n_y_kraj_2];
    toc_1cr_poc_2=[t_1n_x_poc_2,t_1n_y_poc_2];
    toc_1cr_kraj_2=[t_1n_x_kraj_2,t_1n_y_kraj_2];
    [c,d] = size(naj_pojed_2);
    g=d;
   for i=1:1:(d)

        naj_pojedinac2(1,g)=naj_pojed_2(1,i);
        naj_pojedinac2(2,g)=naj_pojed_2(2,i);
    g=g-1;
   end
   naj_pojed_2=naj_pojedinac2;
end

%-------------------------------------
%-------------------------------------
 % CRTANJE KRAJNJEG REZULTATA PUTANJA DVAJU ROBOTA
   if (br_prep_sijeku_1==0 & br_prep_sijeku_2==0)

crtaj_konf=crtaj_konfig(toc_1cr_poc_1,toc_1cr_kraj_1,toc_1cr_poc_2,toc_1cr_kraj_2,d
uz_prostora,sir_prostora,xpr,ypr,naj_pojed_1,naj_pojed_2);
   end

else
naj_pojedinac1=[0 0 ; 0 0];
prep_sijeku1=sjec;
naj_pojedinac2=[0 0 ; 0 0];
prep_sijeku2=sjec;

end

if prep_sijeku1>0 | prep_sijeku2>0
    prep_sijeku1=1;
    prep_sijeku2=1;
end

if prep_sijeku1==0 & prep_sijeku2==0 & (x3==1 | x4==1 |x6==1 | x7==1)
    prep_sijeku1=2;
    prep_sijeku2=3;
end

if prep_sijeku1==0 & prep_sijeku2==0 &  (x1==1 | x2==1 |x5==1 |x8==1)
    prep_sijeku1=3;
    prep_sijeku2=2;

end

 toc_1_kraj_2(1)=duz_prostora-toc_1_kraj_2(1);
```