Vođenje mobilnog robota kroz labirint

Svačina, Mario

Master's thesis / Diplomski rad

2015

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:235:442997

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-08-11**

Repository / Repozitorij:

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

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mario Svačina

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

prof. dr. sc. Dubravko Majetić

Student:

Mario Svačina

Zagreb, 2015.

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

Ovom prilikom htio bi se zahvaliti mentoru prof. dr.sc. Dubravku Majetiću na odličnoj suradnji i velikoj pomoći i savjetima pri izradi ovog rada.

Zatim prof. dr. sc. Josipu Kasaću na korisnim savjetima i pruženoj pomoći.

Također bi se htio zahvaliti svojoj djevojci Ani, obitelji i prijateljima što su mi pružali maksimalnu podršku i vjerovali u mene tijekom svih ovih godina.

Mario Svačina

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: Mario Svačina Mat. br.: 0035178321

Naslov rada na hrvatskom jeziku: **Vođenje mobilnog robota kroz labirint**

Naslov rada na engleskom jeziku: **Guidance of mobile robot through the maze**

Opis zadatka:

Jedan od značajnih problema industrijske robotike je nedostatak robotske mobilnosti. Tako stacionarni roboti i manipulatori često puta nemaju dovoljan radni prostor, odnosno isti mora biti uređen prema performansama samog robota. S druge strane, tu su mobilni roboti koji mogu izvršavati različite zadaće i na raznim udaljenim mjestima. Temeljni zadatak na kojem se intenzivno istražuje jest pitanje robotske percepcije prostora i sigurno gibanje robota neuređenim prostorom.

U radu je potrebno načiniti slijedeće:

1. Prihvatiti slike labirinta s digitalne kamere.
2. Obraditi slike umjetnom neuronskom mrežom.
3. Planirati putanju gibanja robota kroz labirint.
4. Razraditi algoritam vođenja robota zadanom putanjom.
5. Testirati izvedene algoritme na realnom mobilnom robotu.
6. Izvesti zaključke rada.

Zadatak zadan: 7. svibnja 2015.

Rok predaje rada: 9. srpnja 2015.

Predviđeni datum obrane: 15., 16. i 17. srpnja 2015.

Zadatak zadao:

Prof. dr. sc. Dubravko Majetić

Predsjednik Povjerenstva:

Prof. dr. sc. Franjo Cajner

Sadržaj

Sadržaj I

Popis slika III

Popis oznaka V

Sažetak VII

Ključne riječi VII

Summary VIII

Key words VIII

1. Uvod 1

2. Cilj i hipoteza rada 3

2.1. Cilj rada 3

2.2. Hipoteza rada 3

3. Traženje putanje mobilnog robota kroz prostor s preprekama 4

3.1. Širenje valova kroz sliku bez prepreka 7

3.2. Matematički izvod podražajnog DRDS-a 9

3.3. Modifikacije postojećeg *podražajnog* DRDS 12

3.4. Testiranja i dobiveni rezultati rada sustava za generiranje putanje kroz labirint 14

3.4.1. Prvi prostor testiranja 16

3.4.2. Drugi prostor testiranja 17

3.4.3. Treći prostor testiranja 18

3.4.4. Četvrti prostor testiranja 19

3.4.5. Peti prostor testiranja 20

4. Generiranje referentne trajektorije 21

5. Slijeđenje generirane referentne trajektorije korištenjem analitičkog neizrazitog regulatora 25

5.1. Neizrazito upravljanje 25

6. Simulacija gibanja robota zadanim prostom 31

6.1. Simulacija gibanja mobilnog robota prvim prostorom 32
6.2. Simulacija gibanja mobilnog robota drugim prostorom 33
6.3. Simulacija gibanja mobilnog robota trećim prostorom 34
6.4. Simulacija gibanja mobilnog robota četvrtim prostorom 35
6.5. Simulacija gibanja mobilnog robota petim prostorom 36
7. Vizijski sustavi 37
7.1. Digitalna slika 37
7.2. Digitalna obrada slike 41
7.2.1. Ispravljanje distorzije na slici 41
7.2.2. Celularne neuronske mreže (Cellural neural network- CNN) 43
7.2.2.1. Integracija diferencijalne jednadžbe CNN 47
7.2.2.2. Korišteni uzorci u ovom radu 48
7.2.2.2.1. Micanje tankih linija sa slike 48
7.2.2.2.2. Micanje malih površina sa slike 48
7.2.2.2.3. Popunjavanje konkavnih površina 49
7.2.2.2.4. Povećanje površina na slici 50
7.2.3. Traženje težišta i orijentacije izoliranog objekta 51
7.3. Cjelokupni algoritam za traženje robota i prepreka 52
8. Izrada rada i rezultati 54
8.1. Izrada rada 55
8.2. Dobiveni rezultati 56
9. Zaključak 57
Literatura 60

Popis slika

Slika 1: Dijagram toka predloženog algoritma za traženje putanje mobilnog robota 7
Slika 2: Širenja dva vala kroz sliku bez prepreka u nekoliko različitih koraka n_0 8
Slika 3: Dijagram reakcije i difuzije DRDS-a s dva morfogena 10
Slika 4: Prikaz širenja vala kroz labirint 15
Slika 5: Prikaz prvog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju 16
Slika 6: Prikaz drugog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju 17
Slika 7: Prikaz trećeg labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju 18
Slika 8: Prikaz četvrtog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju 19
Slika 9: Prikaz petog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju 20
Slika 10: Horizontalno i vertikalno gibanje mobilnog robota 22
Slika 11: Dijagonalno gibanje mobilnog robota bez prepreke 23
Slika 12: Gibanje mobilnog robota oko prepreke 24
Slika 13: Položaj mobilnog robota u odnosu na referentnu trajektoriju 29
Slika 14: Simulacija gibanja mobilnog robota prvim prostorom 32
Slika 15: Simulacija gibanja mobilnog robota drugim prostorom 33
Slika 16: Simulacija gibanja mobilnog robota trećim prostorom 34
Slika 17: Simulacija gibanja mobilnog robota četvrtim prostorom 35
Slika 18: Simulacija gibanja mobilnog robota kroz peti prostor 36
Slika 19: RGB model boja 39
Slika 20: HSV model boja 40
Slika 21: Ispravljanje slike korištenjem naredbe lensdistort: a) neispravljena slika, b) ispravljena slika opcijom „crop“ , c) ispravljena slika opcijom „fit“ 42
Slika 22: Prikaz ćelija CNN-a 44
Slika 23: Prikaz radijusa utjecaja CNN-a 44
Slika 24: Rubne ćelije CNN-a 45
Slika 25: Blokovski prikaz jedne ćelije CNN-a 46
Slika 26: Izlazna funkcija CNN-a 47

Slika 27: Micanje otvorenih kontura na slici 48
Slika 28: Micanje zaostalih malih površina sa slike 49
Slika 29: Popunjavanje konkavnih površina 50
Slika 30: Povećanje površine prepreka 51
Slika 31: Izgled grafičkog sučelja 53
Slika 32: eMIR mobilni roboti 54

Popis oznaka

DRDS- digitalni reakcijsko-difuzijski sustavi

$\mathbf{x_i}$ –koncentracija i-tog morfogena

n_0- broj koraka

n_1 i n_2- prostorne koordinate sustava

R_i- nelinearna kinetička reakcija i-tog morfogena

T_0- vrijeme uzorkovanja

D- matrica koeficijenta difuzije

(l*x)- operator prostorne konvolucije

T_1 i T_2- vremena uzorkovanja na koordinatama prostorne konvolucije

M- broj morfogena

x_p i y_p- početne koordinatne gibanja

x_n i y_n- nove koordinatne gibanja

t- vrijeme

$\left\|\vec{r}\right\|$ - norma, trenutna vrijednost udaljenosti robota od referentne trajektorije

$\vec{r_e} \times \vec{r}$ - vektorski produkt

V_{pn}- normirani vektorski produkt

u_1 i u_2- upravljačke varijable mobilnog robota

v_{max}- maksimalna translacijska brzina mobilnog robota

w_{max}- maksimalna rotacijska brzina mobilnog robota

C(i,j)- ćelija CNN

S_r- područje utjecaja

x_{ij}- trenutno stanje ćelije

y_{kl}- izlaz ćelije

z_{ij}- prag ćelije

A(i,j;k,l)- povratna veza ćelije

B(i,j;k,l)- kontrolni operator

Sažetak

Tema ovog rada je vođenje mobilnog robota kroz labirint. Kako bi se moglo voditi robota kroz labirint prvo je potrebno isplanirati putanju, potom je potrebno napraviti sustav kontrole lokacije robota i prepreka te napraviti sustav kontrole gibanja. Planiranje putanje mobilnog robota se izvodi pomoću podražajnog digitalnog reakcijsko-difuzijskog sustava (DRDS). Modificiranjem postojećeg podražajnog DRDS-a je postignuta stabilnost sustava te mogućnost planiranja putanje i u kompliciranim prostorima. Nakon planiranja putanje pomoću DRDS-a prikazano je kako generirati referentnu trajektoriju analizom gibanja robota planiranom putanjom. Prilikom generiranja trajektorije uzima se način gibanja robota. Potom je napravljena simulacija kretanja mobilnog robota kroz prostor. Kako bi robot što vjernije i kvalitetnije slijedio referentnu trajektoriju u upravljački sustav robota je ugrađen analitički neizraziti regulator. Nakon dokazivanja uspješnosti metode u simulacijama, izrađen je algoritam za lociranje robota i prepreka pomoću kamere smještene iznad labirinta. U algoritmu za lociranje robota i prepreka su korištene celularne neuronske mreže. Potom je napravljen sustav vođenja robota labirintom. Svi algoritmi korišteni u ovom radu su napravljeni u MATLAB programskom paketu.

Ključne riječi

digitalni reakcijsko-difuzijski sustavi, planiranje puta mobilnog robota, analitički neizraziti regulator, celularne neuronske mreže

Summary

The subject of this paper is the guidance of a mobile robot through the maze. The first step of guiding robot through the maze is path planning. The second step is to project the algorithm for controlling location of robot and obstacles. The third step is to make controlling system for guidance of robot. Path planning of mobile robot is performed by excitable digital reaction-diffusion systems (DRDS). Stability of the system and ability of path planning in complicated spaces are achieved by improving existing excitable DRDS. After path planning with DRDS, it was shown how to generate referent trajectory with analyzing the motion of the mobile robot through planed path. While generating trajectory, the way robot moving was considered. Afterwards, the simulation motioning of the mobile robot through the maze was shown. An analytical fuzzy control was built in the control system of the robot in order to make it more consistent with the referent trajectory tracking. When the success of the method was proven through the simulations, an algorithm for locating the robot and the obstacles with camera, placed above the maze, was made. Cellular neural networks were used in the algorithm of locating robot and obstacles. Lastly, a system of guidance of robot through the maze was made. All proposed algorithms where programmed in MATLAB.

Key words

digital reaction-diffusion systems, path planning of mobile robot, analytic fuzzy control, cellular neural networks

1. Uvod

Robotika je, do današnjega dana, postigla ogroman napredak u mnogim granama ljudske djelatnosti. Najveća rasprostranjenost robotike je trenutačno u industriji gdje roboti odrađuju sve veći broj poslova. Jedan od velikih problema industrijske robotike je njen nedostatak mobilnosti [1]. Roboti i manipulatori koji su fiksirani na određenu poziciju nemaju veliki radni prostor. Međutim, mobilni roboti mogu izvršavati različite zadatke i na različitim udaljenim mjestima. Interes za razvijanje i istraživanje mobilnih robota motiviran je velikom potrebnom i željom da roboti obavljaju zadatke sa ljudima i uz ljude u normalnom ljudskom okruženju kao što su: uredi, bolnice, muzeji i galerije, zračne luke, fakulteti i škole i slično [2]. Kako bi se to postiglo, mobilni roboti moraju biti u mogućnosti izvršavati zadatke u svakodnevnoj ljudskoj okolini bez potrebe za intervencijama u toj istoj okolini. Prvi problem s kojim se susrećemo u mobilnoj robotici je samo pokretanje mobilnog robota. Mobilnom robotu je potreban mehanizam koji će mu omogućiti nesmetano kretanje kroz područje djelovanja [2]. Sljedeći problem je pitanje robotske percepcije prostora u kojem se nalazi. Naravno, sve dok robot ne percipira prostor u kojem se nalazi, nije u mogućnosti niti planirati svoje gibanje kroz isti. Postoje razni pristupi istraživanja i percepcije prostora, a jedan od njih je metoda potencijalnih polja. Također, jedan od mogućih načina percipiranja prostora je uzimanje slike sa kamere koja se nalazi iznad prostora u kojem se robot nalazi. Ukoliko se koristi kamera kao sredstvo percepcije prostora, javljaju se problemi obrade slike pri lociranju prepreka i robota.

Međutim, tu se otvara nekoliko novih pitanja. Postavlja se pitanje, što kad je robot istražio prostor i ima informaciju o samom prostoru? Kako planirati njegovo gibanje? Je li potrebno da robot samo stigne do cilja, ili je možda želja da to obavi što kraćim putem, ili u

što kraćem vremenu, ili sa što manjim utroškom energije? Što ako je prostor uvelike kompliciran i postoji opasnost od krivog skretanja?

Ukoliko se svi navedeni problemi riješe te moguće je isplanirati putanju gibanja robota, postavlja se pitanje kako osigurati gibanje robota planiranom putanjom. Koji regulator odabrati? Kolike brzine gibanja robota koristiti?

Ovaj rad će pokušati dati odgovore na ova postavljena pitanja. U drugom poglavlju se postavlja cilj i hipoteza samog rada. U trećem poglavlju se prikazuje matematički model sustava te rezultati postignuti njegovom primjenom. U četvrtom poglavlju je prezentiran način izrade referentne trajektorije. U petom poglavlju je prikazan način izrade regulatora koji omogućava robotu slijeđenje generirane trajektorije. U šestom poglavlju su prikazane simulacije gibanja mobilnog robota željenom putanjom. U sedmom poglavlju je su prikazani korišteni algoritmi za obradu slike te objašnjeno na koji način je sve implementirano za praktičnu primjenu. U osmom poglavlju su prikazani rezultati rada na realnom robotu. U zadnjem poglavlju su izvedeni zaključci rada te predložene promjene u svrhu poboljšanja samog rada.

2. Cilj i hipoteza rada

2.1. Cilj rada

Prethodno postavljena pitanja motiv su izrade ovog rada. Osnovni cilj je odrediti referentnu trajektoriju gibanja mobilnog robota kroz zadani prostor te izvršiti gibanje na realnom robotu. Kako bi cilj bio ostvariv potrebno je uvesti sljedeće pretpostavke:

- početna i krajnja pozicija gibanja robota moraju imati barem jednu mogućnost povezivanja jedne s drugom
- mobilni robot mora fizički moći prolaziti između prepreka koje definiraju prostor.

Također, cilj rada je da određena trajektorija bude što kraća, odnosno da pronađeni put koji mobilni robot treba proći od početne do konačne pozicije bude što je manji mogući.

2.2. Hipoteza rada

Adekvatnim opisom prostora kojim se kreće mobilni robot moguće je isplanirati referentnu trajektoriju od početne do konačne pozicije. Također je među zadanim pozicijama moguće pronaći najkraću moguću putanju gibanja robota. Time se štedi vrijeme i energija za gibanje robota, a ušteda energije je posebno zanimljiva u mobilnoj robotici. Adekvatnom regulacijom moguće je postići gibanje robota isplaniranom putanjom u realnim uvjetima.

3. Traženje putanje mobilnog robota kroz prostor s preprekama

Cilj ovog rada je prezentirati jedan od načina pronalaska putanje gibanja mobilnog robota kroz prostor s preprekama. Predloženo rješenje se temelji na reakcijsko difuzijskom sustavu koje je prvi predložio A. M. Turning 1952. Turning zaključuje da pod određenim uvjetima, kemikalije mogu reagirati i difundirati na način da tvore stabilno stanje heterogenih prostornih uzoraka kemikalija ili koncentracija morfogena [3]. Na temelju izvedenog zaključka predložio je nelinearni mehanizam za opisivanje morfogenoze, odnosno formiranja oblika i uzoraka u prirodi [4]. Prvenstveno su ovi sustavi istraživani od strane biologa koji su se bavili teorijom prirodnog oblikovanja uzoraka i kemičara koji su modelirali dinamičke oscilatorne reakcije. Međutim, ovakav sustav kako ga je predložio Turning, zbog svoje općenite neograničenosti [5] nije bio u potpunosti pogodan za rješavanje problema obrade slike.

Koristeći Belousov-Zhabotinsky-evu kemijsku reakciju [6] implementiranu u reakcijsko-difuzijske sustave, O. Steibock, A. Toth i K. Showalter su 1995. godine predložili sustav za optimalno traženje trajektorije robota. Međutim, upotreba ovih sustava nije zaživjela zbog korištenja pravih kemijskih medija pri korištenju računalnih zadataka i ograničene stabilnosti operacija. Uz to, veličina i kompleksnost prostora koji su se mogli koristiti bili su prilično limitirani [7].

A. S. Sherstinsky i R. W. Picard u svojem radu iz 1996. [5] predložili su „M-lattice“, sustav koji zadržava mehanizam formiranja oblika te je uz to i ograničen. U teorijskoj osnovi ovaj sustav je srodan Hopfield-ovim i celularnim umjetnim neuronskim mrežama, ali uz puno veću fleksibilnost djelovanja parametara. U [5] je pokazan osnovni princip „M-lattice“ sustava te njihova primjena na restauraciji otisaka prstiju. Općenito, većina gore navedenih

sustava su opisani kao vremenski i prostorno kontinuirani reakcijsko-difuzijski sustavi, te stoga nisu direktno upotrebljivi u teoriji digitalnog procesiranja signala [8].

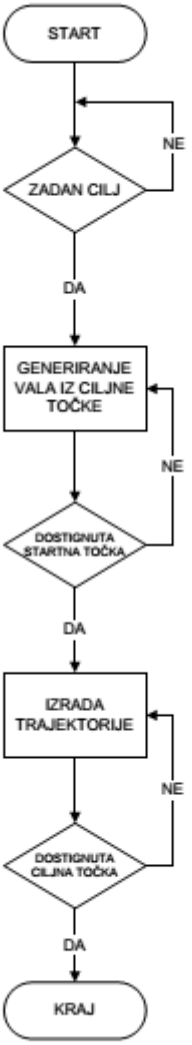
K. Ito, T. Aoki i T. Higuchi su u svojem radu 2001. godine predložili korištenje digitalnih reakcijsko-difuzijskih sustava (DRDS), a koji su prilagođeni digitalnom procesiranju signala [7], [8], [9], [10]. U kasnijim radovima su predložene sustave koristili za generiranje valova kroz prostor u svrhu pronalaska najkraće putanje mobilnog robota iz početne u konačnu poziciju [7], [9]. Oni su predložili *podražajni* (engl. excitable) DRDS baziran na FitzHugh-Nagumo [11] tipu dinamike. Predloženi sustav kreira podražajni val slijedećih karakteristika: val napreduje kroz prostor konstantnom brzinom te se stapa u sudaru s ostalim valovima. Osnovna ideja koju su predstavili K. Ito, T. Aoki i T. Higuchi je generirati val iz ciljne točke mobilnog robota te pustiti val da se giba prostorom. Propagacijom kroz prostor, val istražuje sve moguće smjerove i načine gibanja iz jedne točke u drugu. Gibanje vala kroz prostor završava u trenutku kada val stigne do početne pozicije mobilnog robota. Potom se pristupa evaluaciji točaka kroz koje je val prošao te se na temelju rezultata generira putanja mobilnog robota. Dva su bitna nedostataka pristupa koji su predložili K. Ito, T. Aoki i T. Higuchi. Prvi nedostatak se očituje u potrebi za posebnim mjestom za spremanje točaka kroz koje je val prošao, a drugi je raspršivanje vrijednosti izlaza prema beskonačnosti, što proizlazi iz samog matematičkog modela sustava.

U [12], [13], [14] i [15] također je predložen način izrade valova, ali pomoću celularnih neuronskih mreža, uz razliku što se zapis gibanja vala zapisuje direktno u sliku prostora te se evaluacija udaljenosti očitava direktno sa slike.

U ovom radu će se koristiti modificirani *podražajni* DRDS na način da će se vrijednosti izlaza sustava ograničiti te mjesta prolaska vala zapisivati direktno na sliku. Međutim, samo ograničavanje vrijednosti izlaza uzrokuje novi problem. On se očituje u povećanom stvaranju

lokalnih minimuma, te samim time onemogućava pronalaženje rješenja za zadane probleme pronalaska puta mobilnog robota kroz zadani prostor. Rješavanjem i ovog problema, sustav će biti spreman za potpunu primjenu u praksi.

Osnovna ideja od koje se polazi pri izradi valova temeljem čijeg gibanja se radi kasnija evaluacija je postavljanje najviše numeričke vrijednosti sustava u točci izvora vala, odnosno ciljnoj točci robota. Pri tome, vrijednosti ostalih točaka kroz koje je val prošao opadaju proporcionalno sa udaljenosti točke od središta vala. Sam proces širenja vala kroz prostor prestaje u onom trenutku kada val dođe do početne pozicije robota. Nakon toga, počinje proces evaluacije okolnih točaka, a kako bi se krenulo u izradu putanje gibanja mobilnog robota. Proces evaluacije se odvija na način da se u susjedstvu piksela, koji predstavlja trenutnu virtualnu poziciju robota, odabere sljedeći piksel koji ima najveću vrijednost. Ovaj postupak se ponavlja sve dok se ne dostigne pozicija s najvećom vrijednosti, odnosno ciljna pozicija zadatka. S ovim zadnjim korakom završava planiranje putanje koju robot treba proći od početne do konačne točke. Prilikom samog procesa evaluacije točaka kroz koje robot treba prolaziti od početne u konačnu poziciju, zapisuju se koordinate tih točaka kako bi se potom moglo pristupiti izradi referentne trajektorije. Sam prostor u kojem se robot nalazi je prezentiran slikom tlocrta prostora robota, a robot je prezentiran veličinom jednog piksela. Na slici 1 je prikazan dijagram toka predloženog načina izrade putanje robota.

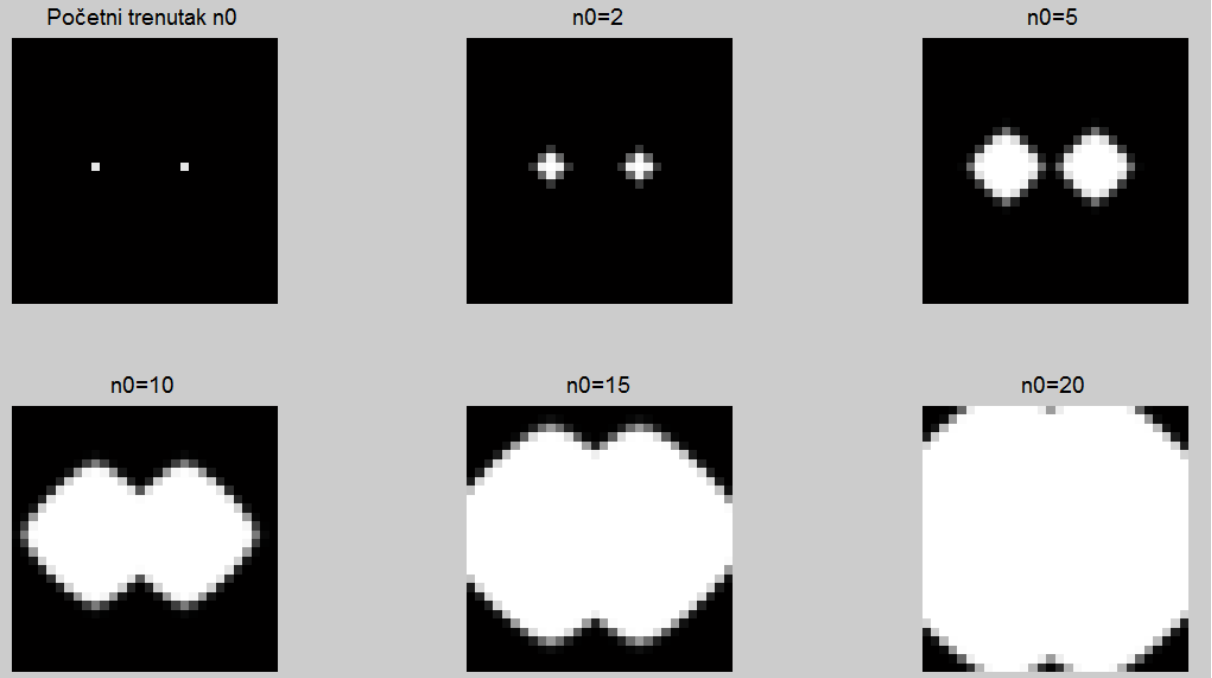


Slika 1: Dijagram toka predloženog algoritma za traženje putanje mobilnog robota

3.1. Širenje valova kroz sliku bez prepreka

Kako bi se pokazala osnovna ideja širenja valova kroz sliku biti će kratko objašnjeno nastajanje i širenje valova. Širenje valova će prvo biti objašnjeno na slici bez prepreka. Na početku je potrebno definirati sliku dimenzija nxm koja ima vrijednosti svih piksela jednaku nuli. Potom na željeno mjesto ili više mjesta, odnosno koordinata piksela postaviti izvor, odnosno izvore valova. Izvor valova je reprezentiran postavljanjem neke vrijednosti koja će poslužiti kao okidač za daljnje širenja, a u ovom radu je ta vrijednost postavljena na 0,9. Nakon što su definirani svi željeni izvori moguće je pokrenuti program širenja valova kroz

sliku. U sljedećem koraku se izračunavaju vrijednosti svakog piksela slike te mu se određenim relacijama dodaju vrijednosti njemu susjednih piksela. Potom ti pikseli poprimaju nove vrijednosti u odnosu na susjedne piksele. Drugim riječima, to bi značilo da su pikseli u susjedstvu izvora poprimili nove vrijednosti različite od nula. Samim time, povećanjem broja koraka sve više se povećava broj piksela vrijednosti većih od nula te se na taj način odvija širenje vala. Također, na taj se način postiže kontinuirano i ravnomjerno kretanje odnosno širenje vala kroz sliku. Sami matematički izrazi iz kojih proizlazi ovaj opis prikazani su u sljedećem potpoglavlju. Prije samog pokazivanja matematičkih izraza na slici 2 je prikazano željeno ponašanje širenja valova kroz sliku.



Slika 2: Širenja dva vala kroz sliku bez prepreka u nekoliko različitih koraka n_0

Kao što se na slikama može vidjeti, širenje valova je jednoliko u svim smjerovima te se valovi međusobno ne poništavaju nego zajednički nastavljaju gibanje prema krajevima slike gdje se val još nije proširio.

3.2. Matematički izvod podražajnog DRDS-a

U ovom potpoglavlju će biti prezentiran način izrade *podražajnog* DRDS-a kako je navedeno u [7], [8], [9], [10]. DRDS se općenito sastoji od niza paralelnih morfogena. Morfogen u biologiji označava tvar koja upravlja razvojem stanica na temelju svoje koncentracije [4], a u primjeru DRDS-a morfogen je svaki paralelni sloj *M* koji predstavlja slike kroz koje se val širi [7]. Pomoću morfogena se kasnije određuje referentna trajektorija.

Općenit izraz za *M*-morfogen reakcijsko-difuzijskog sustava u dvodimenzionalnom koordinatnom sustavu (n_1,n_2) je zapisan kao:

$$\mathbf{x}(n_0, n_1, n_2) = \mathbf{x}(n_0, n_1, n_2) + \mathbf{R}(\mathbf{x}(n_0, n_1, n_2)) + \mathbf{D}(l * \mathbf{x})(n_0, n_1, n_2), \tag{3.1}$$

gdje je n_0 broj koraka, n_1 i n_2 koordinate i-tog piksela, a $\mathbf{x} = [x_1, x_2, ..., x_M]^T$, odnosno $\boldsymbol{x_i}$ je koncentracija i-tog morfogena, $R = T_0 R = [R_1(x), R_2(x), ..., R_M(x)]^T$, gdje je $\boldsymbol{R_i}$ nelinearna kinetička reakcija i-tog morfogena, a T_0 je vrijeme uzorkovanja. **D** je dijagonalna matrica koja sadrži vrijednosti koeficijenta difuzije i-tog morfogena.

Operator $(l * \mathbf{x})(n_0, n_1, n_2)$ označava prostornu konvoluciju definiranu kao:

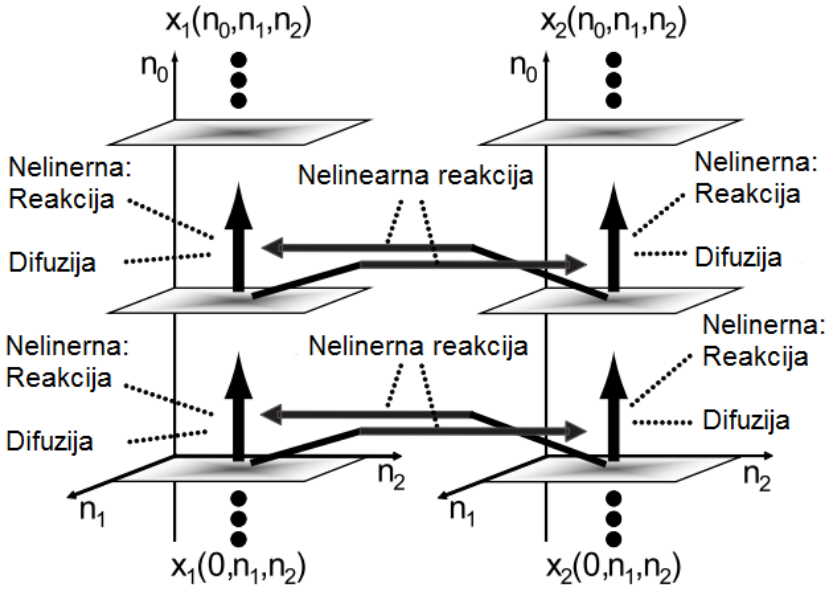
$$(l * \mathbf{x})(n_0, n_1, n_2) = \begin{bmatrix} (l * x_1)(n_0, n_1, n_2) \\ (l * x_2)(n_0, n_1, n_2) \\ \vdots \\ (l * x_M)(n_0, n_1, n_2) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{p_1=-1}^{1} \sum_{p_2=-1}^{1} l(p_1, p_2) x_1(n_0, n_1 - p_1, n_2 - p_2) \\ \sum_{p_1=-1}^{1} \sum_{p_2=-1}^{1} l(p_1, p_2) x_2(n_0, n_1 - p_1, n_2 - p_2) \\ \vdots \\ \sum_{p_1=-1}^{1} \sum_{p_2=-1}^{1} l(p_1, p_2) x_M(n_0, n_1 - p_1, n_2 - p_2) \end{bmatrix}, \tag{3.2}$$

gdje je $l(p_1,p_2)$ definiran kao:

$$l(p_1, p_2) = \begin{cases} \dfrac{1}{T_1^2} & (n_1, n_2) = (-1, 0), (1, 0) \\ \dfrac{1}{T_2^2} & (n_1, n_2) = (0, 1), (0, 1) \\ -2\left(\dfrac{1}{T_1^2} + \dfrac{1}{T_2^2}\right) & (n_1, n_2) = (0, 0) \\ 0 & \textit{ostalo} \end{cases},$$

gdje su T_1 i T_2 vremena uzorkovanja vrijednosti na koordinatama. Slika 3 ilustrira DRDS gdje je *M=2*, odnosno DRDS se sastoji od dva morfogena.



Slika 3: Dijagram reakcije i difuzije DRDS-a s dva morfogena

Pomoću gore navedene definicije DRDS promjenom nelinearne reakcije kinetike ***R*** i njenih parametara moguće je simulirati različite dinamike reakcije i difuzije u sustavu. Kao što je predloženo u [7], [8], [9] i [10] i u ovom radu će se koristiti FitzHugh-Nagumo (FHN) model dinamike sustava. FHN model *podražajnog* DRDS je sastavljen od dva morfogena te je definiran sljedećim izrazom:

$$\begin{bmatrix} x_1(n_0+1,n_1,n_2) \\ x_2(n_0+1,n_1,n_2) \end{bmatrix} = \begin{bmatrix} x_1(n_0,n_1,n_2) \\ x_2(n_0,n_1,n_2) \end{bmatrix} + \begin{bmatrix} R_1(x_1(n_0,n_1,n_2),x_2(n_0,n_1,n_2)) \\ R_2(x_1(n_0,n_1,n_2),x_2(n_0,n_1,n_2)) \end{bmatrix} + \begin{bmatrix} D_1(l*x_1)(n_0,n_1,n_2) \\ D_2(l*x_2)(n_0,n_1,n_2) \end{bmatrix}, \quad (3.3)$$

gdje je $R_1(x_1,x_2) = T_0\left\{\frac{1}{k_1}\left[x_1\left(x_1-k_2\right)\left(1-x_1\right)-x_2\right]\right\}$, $R_2(x_1,x_2) = T_0\left(x_1-k_3x_2\right)$, a vrijednosti parametara redom su: $k_1 = 10^{-3}, k_2 = 10^{-6}, k_3 = 0{,}1, D_1 = 2, D_2 = 0, T_0 = 10^{-3}, T_1 = T_2 = 1.$

Kao što je već navedeno, glavni uvjeti ovakvog sustava za širenje valovu su propagacija valova konstantnom brzinom i stapanje različitih valova u prostoru. Kao što je vidljivo iz jednadžbi sustava oba uvjeta su ostvarena. Međutim, postoji veliki nedostatak predloženog sustava. Problem se vidi iz samog pogleda na jednadžbe sustava, a očituje se neograničenošću vrijednosti izlaza. Drugim riječima, vrijednosti piksela nakon nekoliko koraka teže prema beskonačnosti. Rješenje ovog problema je dano u sljedećem potpoglavlju.

3.3. Modifikacije postojećeg *podražajnog* DRDS

Kao što je već opisano problem *podražajnog* DRDS-a su neograničene vrijednosti izlaza. Problem se očituje iznosima brojeva koji teže u beskonačnost te računalo nije u mogućnosti s takvim iznosima obavljati zadane zadatke. Stoga, potrebno je ograničiti vrijednosti izlaznih varijabli. U ovom radu ograničenje je postignuto korištenjem tangensa hiperbolnog. Time je postignuto ograničenje vrijednosti u svakom koraku te zaustavljen njihov neograničen rast. Primjenom ovih ograničenja jednadžba ovog sustava poprima sljedeći oblik:

$$\begin{bmatrix} x_1(n_0+1,n_1,n_2) \\ x_2(n_0+1,n_1,n_2) \end{bmatrix} = \tanh\left\{ \begin{bmatrix} x_1(n_0,n_1,n_2) \\ x_2(n_0,n_1,n_2) \end{bmatrix} + \begin{bmatrix} R_1(x_1(n_0,n_1,n_2),x_2(n_0,n_1,n_2)) \\ R_2(x_1(n_0,n_1,n_2),x_2(n_0,n_1,n_2)) \end{bmatrix} + \begin{bmatrix} D_1(l*x_1)(n_0,n_1,n_2) \\ D_2(l*x_2)(n_0,n_1,n_2) \end{bmatrix} \right\}, \quad (3.4)$$

gdje je $R_1(x_1,x_2) = T_0\left\{\frac{1}{k_1}\left[x_1\left(x_1-k_2\right)\left(1-x_1\right)-x_2\right]\right\}$, $R_2(x_1,x_2) = T_0\left(x_1-k_3x_2\right)$, a vrijednosti parametara redom su: $k_1 = 10^{-3}, k_2 = 10^{-6}, k_3 = 0{,}1, D_1 = 2, D_2 = 0, T_0 = 10^{-3}, T_1 = T_2 = 2$

Primjenom ovog ograničenja, vrijednosti izlaza sustava su ograničene na vrijednosti između -1 i 1. Dodatno su promijenjeni parametri T_1 i T_2 kako bi se povećao utjecaj srednjeg člana u izrazu za prostornu konvoluciju. S obzirom da x_1 i x_2 označavaju prvi i drugi morfogen sustava, jasno je da njihove matrice reprezentiraju rješenja za dobivanje puta kroz zadani prostor iz početne u konačnu točku. U [7] i [9] je za praćenje vala korišten prvi morfogen, međutim, u ovom radu je odabran drugi morfogen. Razlog korištenja drugog morfogena je to što zbog tangensa hiperbolnog prvi morfogen ulazi nakon dva do tri koraka u zasićenje te njegove vrijednosti teže u jedan. Iz izraza za izlaz drugog morfogena se zaključuje da on ne ulazi u zasićenje zbog ograničenja koje je nametnuto vrijednosti T_0. Međutim, ukoliko bi se koristili jako veliki prostori za izračunavanje postoji mogućnosti da i

drugi morfogen također uđe u zasićenje. Problem se u takvoj situaciji rješava smanjivanjem vrijednosti $k_1,\ k_2,\ k_3,\ D_1,\ D_2,\ T_0,\ T_1\ i\ T_2$.

Međutim, ponuđenim rješenjem ograničavanja izlaznih varijabli, pojavio se novi problem. Problem je nastao u povećanom stvaranju lokalnih minimuma, a to se očitovalo prilikom širenja valova na više od 20-tak koraka. Analizom samog sustava, korijen problema je pronađen u samom izrazu za prostornu konvoluciju sustava. Problematičnim je lociran srednji član koji je u [7], [8], [9] i [10] dan izrazom $-2\left(\frac{1}{T_1^2}+\frac{1}{T_2^2}\right)$, te je iz njega vidljivo da se taj srednji član oduzima od svih susjednih članova. Direktna posljedica toga je stvaranje lokalnih minimuma zbog onemogućavanja rasta numeričke vrijednosti vala na svakom pojedinom pikselu. To je rezultiralo velikim brojem susjednih piksela s identičnim vrijednostima. Problem je riješen postavljanjem pozitivnog predznaka umjesto negativnog. Dodatno je predloženo da na tu vrijednost ne utječu vrijednosti ostalih parametara nego da je ovisna samo o sebi, a u ovom radu je odabrana vrijednost 1. Primjenom ovog rješenja prostorna konvolucija poprima sljedeći oblik:

$$(l*\mathbf{x})(n_0,n_1,n_2)=\begin{bmatrix}(l*x_1)(n_0,n_1,n_2)\\(l*x_2)(n_0,n_1,n_2)\\\vdots\\(l*x_M)(n_0,n_1,n_2)\end{bmatrix}$$

$$=\begin{bmatrix}\sum_{p_1=-1}^{1}\sum_{p_2=-1}^{1}l(p_1,p_2)x_1(n_0,n_1-p_1,n_2-p_2)\\\sum_{p_1=-1}^{1}\sum_{p_2=-1}^{1}l(p_1,p_2)x_2(n_0,n_1-p_1,n_2-p_2)\\\vdots\\\sum_{p_1=-1}^{1}\sum_{p_2=-1}^{1}l(p_1,p_2)x_M(n_0,n_1-p_1,n_2-p_2)\end{bmatrix}, \tag{3.5}$$

gdje je $l(p_1,p_2)$ definiran na način:

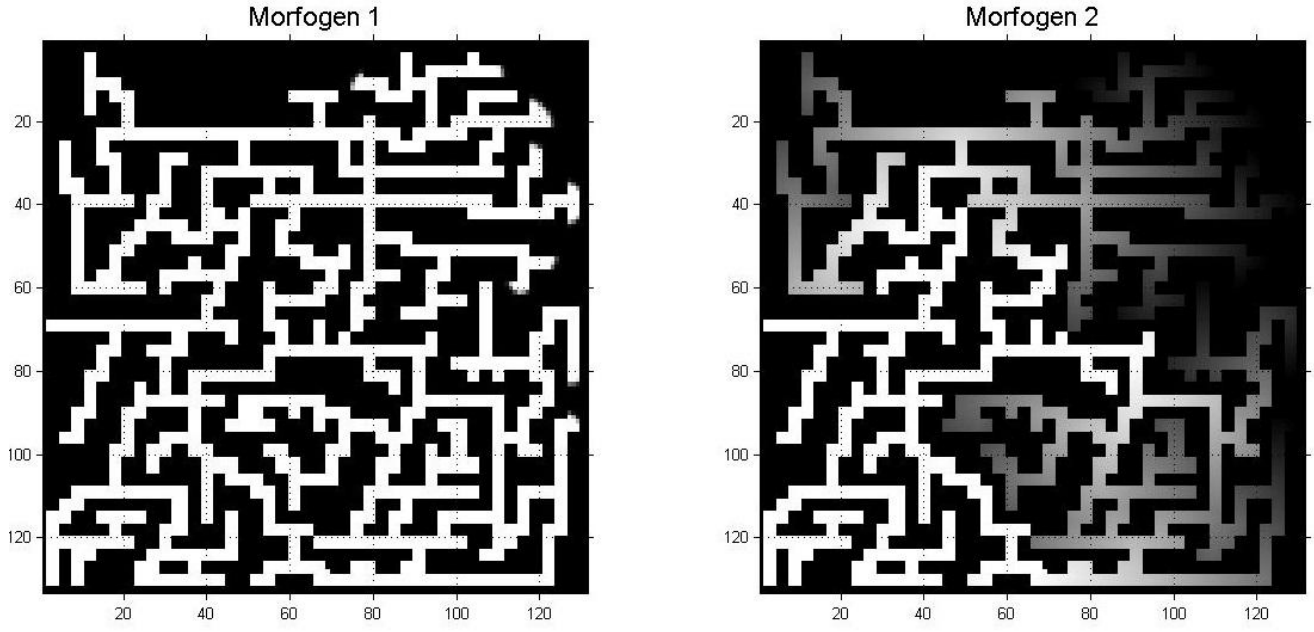
$$l(p_1, p_2) = \begin{cases} \frac{1}{T_1^2} & (n_1, n_2) = (-1, 0), (1, 0) \\ \frac{1}{T_2^2} & (n_1, n_2) = (0, 1), (0, 1) \\ 1 & (n_1, n_2) = (0, 0) \\ 0 & ostalo \end{cases} .$$

Ovom promjenom je postignuto sprječavanje nastajanja lokalnih minimuma koji bi onemogućavali traženje željene referentne trajektorije. Takvom modifikacijom sustava, isti je u potpunosti sposoban riješiti probleme traženja putanje za sve oblike prostora u kojem se robot nalazi.

3.4. Testiranja i dobiveni rezultati rada sustava za generiranje putanje kroz labirint

U potpoglavlju 3.1 pokazan je način širenja dva vala kroz sliku bez prepreka. Ovdje su prikazani rezultati sustava kroz pet različitih labirinata, odnosno prostora testiranja. Međutim, prije pokazivanja samih rezultata testiranja potrebno je pobliže objasniti funkcioniranje i definiranje prepreka od kojih se valovi odbijaju te koje robot treba zaobilaziti. S obzirom da je u ovom radu korišten modificirani *podražajni* DRDS sa dva morfogena, potrebno je na početku definirati dimenzije matrica morfogena koje su jednake dimenziji slike labirinta te oni predstavljaju svojevrsnu sliku širenja vala kroz labirint. Robot koji se kreće kroz labirint je predstavljen veličinom jednog piksela. Vrijednosti svih elemenata matrice morfogena je potrebno u početnom trenutku postaviti u nulu. Razlog takvog postavljanja se krije u jednadžbama sustava koje reagiraju i na najmanju moguću promjenu vrijednosti te je pretvaraju u val. Naravno, kako bi širenje vala bilo moguće potrebno je definirati pobudu u točki cilja, a vrijednost pobude kao što je već navedeno je 0,9. Sljedeći problem proizlazi iz

samog uvjeta da su sve ostale vrijednosti nula. Ako su sve vrijednosti nula može se zaključiti da ne postoje granice u sustavu, odnosno da nigdje u matricama morfogena nisu označene prepreke. Problem nastaje u trenutku kada susjedni pikseli koji bi trebali označavat prepreke budu podraženi valom. Iz jednadžbi sustava je vidljivo da će i pikseli koji predstavljaju prepreke biti podraženi od tog vala te poprimiti novu vrijednost veću od nule. Moguća su dva rješenja ovog problema. Prvo rješenje je nakon svakog koraka računanja širenja vala kroz prostor sve promijenjene vrijednosti piksela koji označavaju prepreke resetirati, odnosno ponovno ih postaviti u nulu. A drugo rješenje je računati promjene samo na slobodnim pikselima te na taj način izbjegnuti promijene piksela koji predstavljaju prepreke jer se njihovo stanje neće ni računati, a posljedično tome ni mijenjati. U praksi se drugo rješenje pokazalo kako brže i efikasnije u smislu smanjenja trajanja obrade. Na slici 4 je prikazano širenje vala kroz labirint.



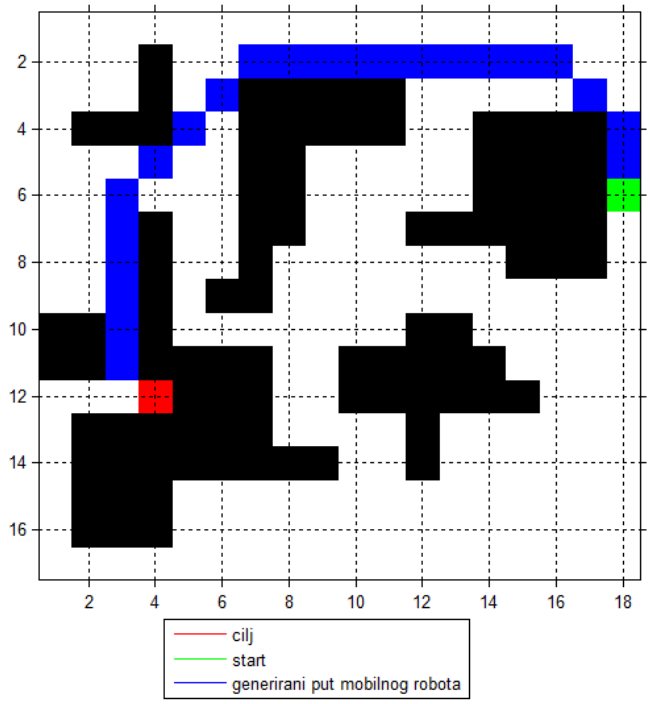
Slika 4: Prikaz širenja vala kroz labirint

Kao što je vidljivo na slici 4, val je proširen po svim prostorima na koje nailazi. Prikaz slike je prikazan na način da bijeli pikseli iznose jedan, a crni nula te sve ostale nijanse sive boje iznose između nula i jedan. Drugi morfogen je prije samog prikaza pomnožen sa deset kako bi

se jače vidjeli svi dijelovi labirinta kroz koje je val prošao. Prvi morfogen je prikazan na način kako mu i vrijednosti izlaza pokazuju, te se može primijetiti kako je većina labirinta otkrivena i vrijednosti su joj postigle jedan. Zbog toga, kao što je već ranije navedeno korišten je drugi morfogen za određivanje puta mobilnog robota.

3.4.1. Prvi prostor testiranja

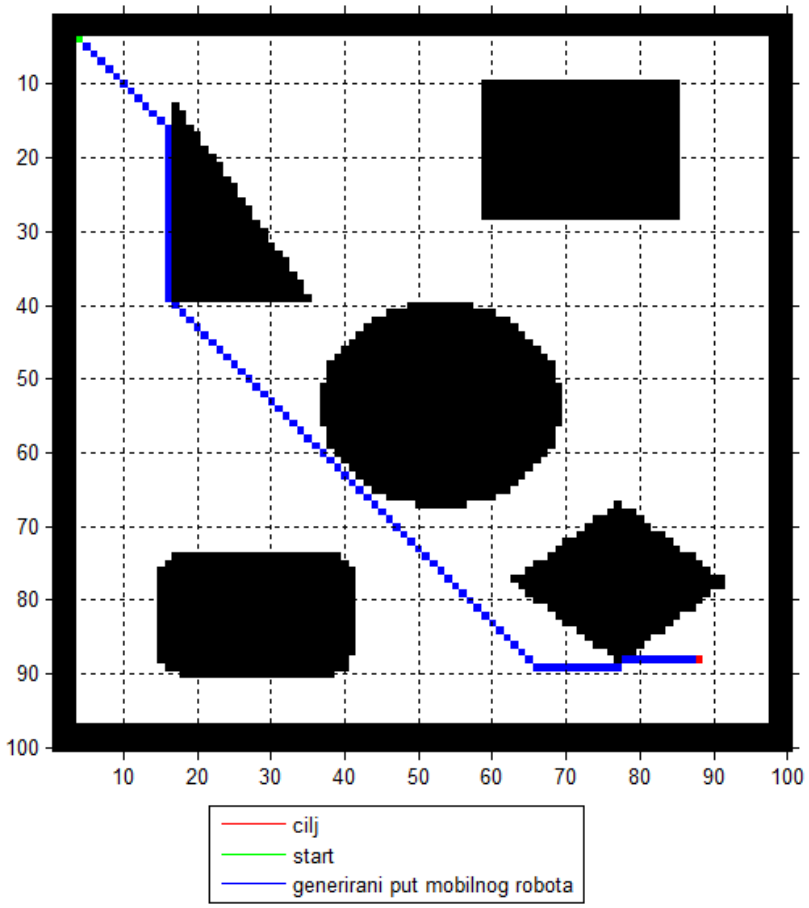
Prvi prostor kojim je testiran predloženi algoritam je dimenzija 18x17 piksela. Prepreke su u svim testiranjima označene crnom bojom, a slobodan prostor bijelom bojom. Na slikama testova zelenom bojom su označena početna pozicija robota, plavom bojom je označena generirana putanja robota, a crvenom bojom je označena ciljna pozicija robota. Traženje putanje mobilnog robota je za početnu poziciju (18,6) i ciljnu poziciju (4,12). Širenje vala kroz prostor se odvijalo kroz 31 korak, a sama željena putanja se proteže kroz 23 točke prostora. Vrijeme trajanja izračuna je iznosi između 0.4 i 0.5 sekundi.



Slika 5: Prikaz prvog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju

3.4.2. Drugi prostor testiranja

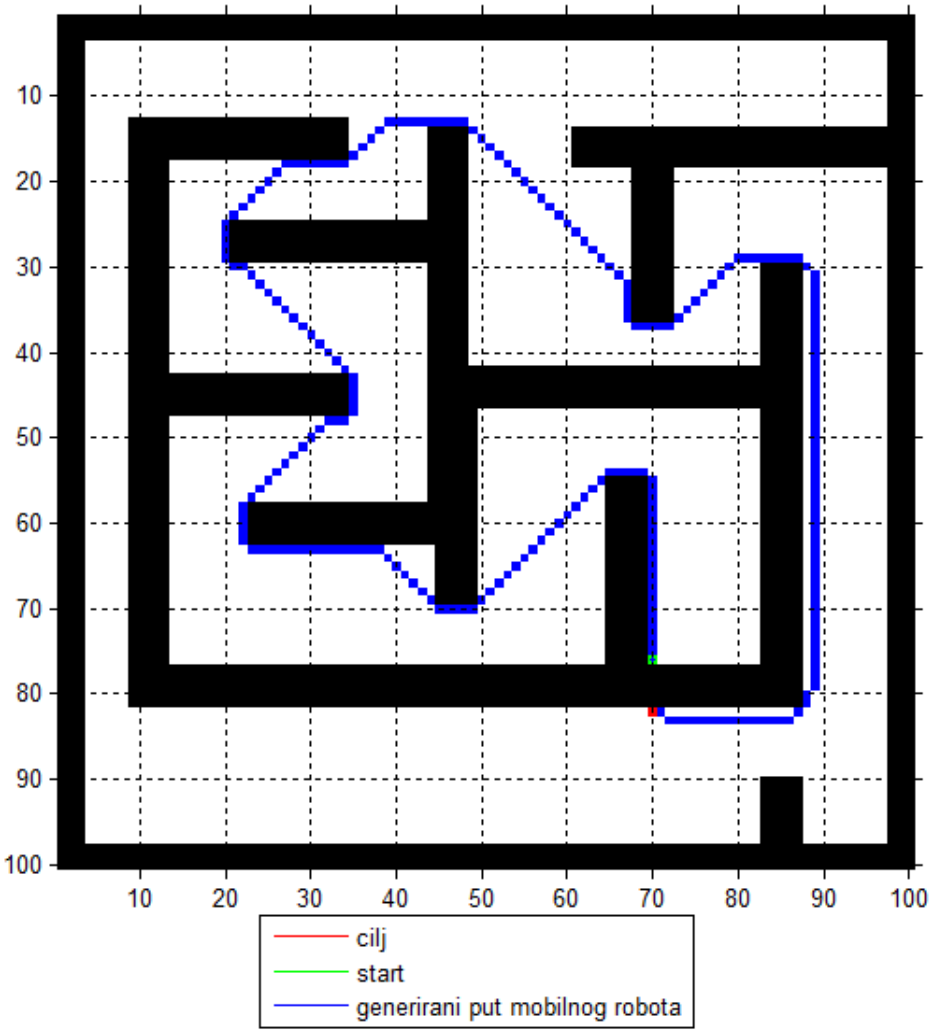
Drugi prostor za testiranje je veličine 100x100 piksela. Ovaj prostor nije klasični labirint kroz koji robot treba pronaći put, ali prikazuje pronalaženje najkraće moguće putanje za definiranu početnu i krajnju poziciju. Kao što je vidljivo iz slike, predložena putanja je u okolici prepreka striktno paralelna s preprekama, a u prostoru gdje nema prepreka se proteže dijagonalno u odnosu na osi koordinatnog sustava. Iz toga se jasno vidi kako je predložena putanja zaista i najkraća. Za početnu poziciju robota je odabrana točka (4,4), a za ciljnu poziciju (88,88). Za zadane početne i konačne pozicije širenje vala se odvijalo u 171 koraku. Dobivena putanja se sastoji od 107 točaka, a vrijeme izračuna je malo više od 9 sekundi.



Slika 6: Prikaz drugog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju

3.4.3. Treći prostor testiranja

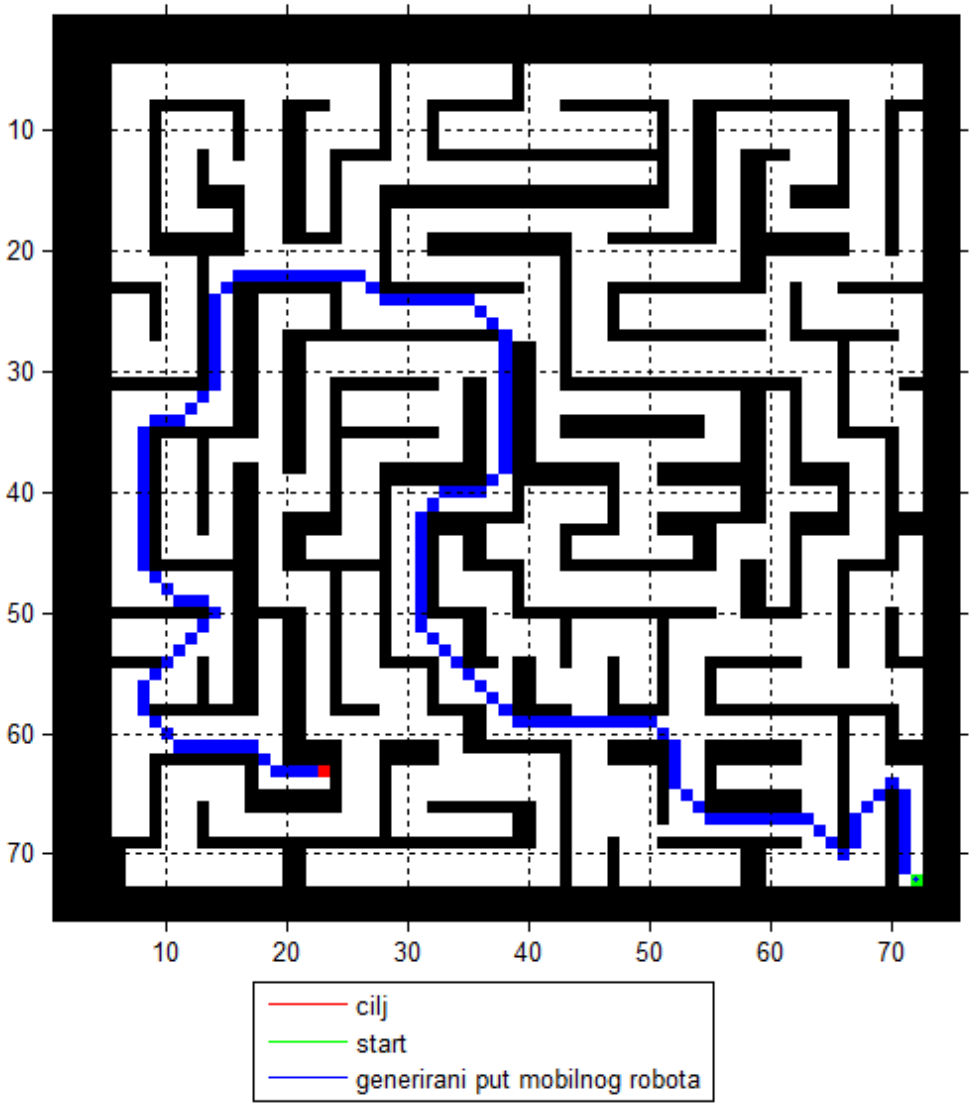
Treći prostor je definiran sa 100x100 piksela. U ovoj situaciji je vidljivo rješenje prolaska kroz labirint za zadanu početnu i krajnju poziciju, ali je također vidljivo i korištenje najkraćih putanja za gibanje kroz sami labirint. Kao početna pozicija je odabrana točka (70,76), a kao ciljna pozicija (70,82). Kako bi val stigao iz ciljne do početne pozicije bilo je potrebno izvršiti 343 koraka, a putanja kojom mobilni robot treba proći proteže se kroz 250 točaka. Vrijeme računanja je oko 17 sekundi.



Slika 7: Prikaz trećeg labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju

3.4.4. Četvrti prostor testiranja

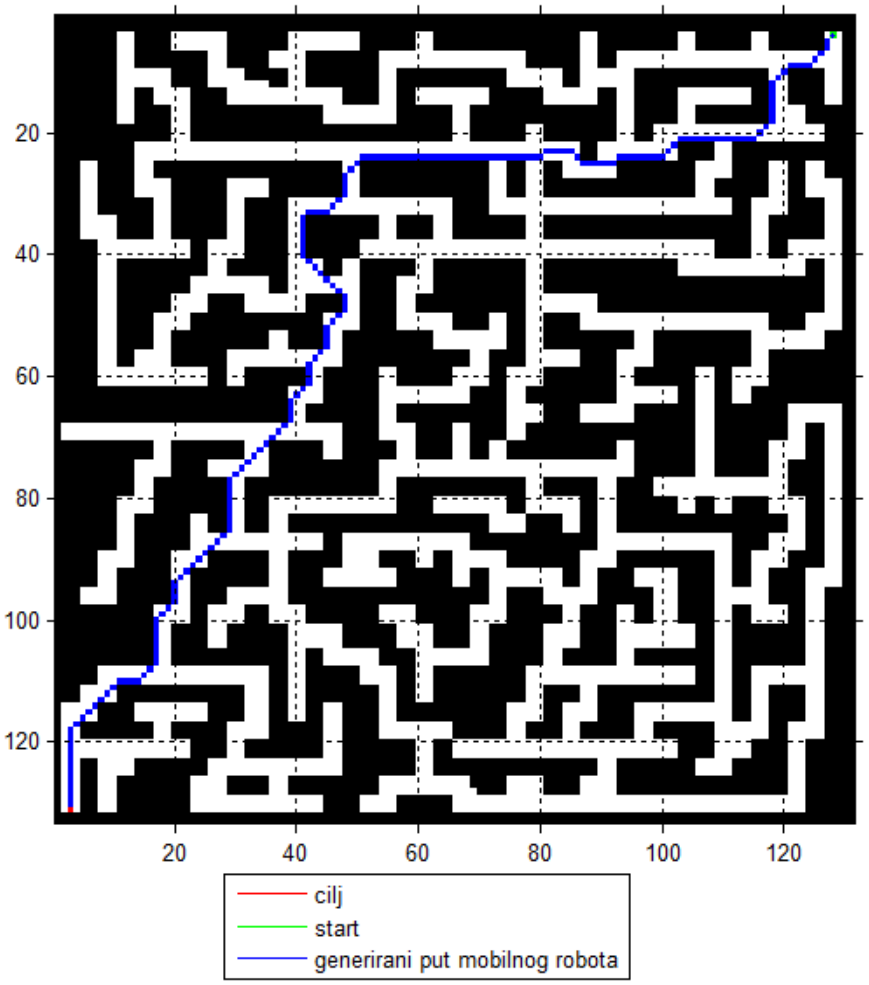
Kao četvrti labirint je korišten prostor veličine 75x75 piksela. Ovim primjerom je prikazano rješavanje problema prolaska robota kroz složenije labirinte. Početna točka robota je (72,72), dok je ciljna točka (23,63). Sama propagacija vala se odvijala u 209 koraka, a sam robot treba proći kroz 156 točaka kako bi došao iz početne u konačnu poziciju. Vrijeme računanja se proteže između 8 i 9 sekundi.



Slika 8: Prikaz četvrtog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju

3.4.5. Peti prostor testiranja

Kao posljednji prikazani labirint je izabran prostor dimenzija 133x131 piksel. Ovaj labirint je još jedan predstavnik složenijeg labirinta, ali uz gotovo dva puta veće dimenzije od prethodno opisanog. U ovom labirintu kao početna točka je (128,4), a ciljna točka je (3,131). Kako bi se val proširio do obje točke bilo je potrebno 269 koraka, a generirana putanja robota prolazi kroz 197 točaka. Vrijeme trajanja je najveće od svih do sad prikazanih labirinata te iznosi nešto više od 23 sekunde.



Slika 9: Prikaz petog labirinta sa generiranom putanjom za zadanu početnu i konačnu poziciju

4. Generiranje referentne trajektorije

U trećem poglavlju je prikazano kako generirati točke kroz koje mobilni robot treba prolaziti kako bi stigao iz početne u konačnu poziciju. Međutim, da bi robot zaista i prošao kroz te točke, iste se koriste za generiranje referentne trajektorije mobilnog robota. Način generiranja referentne trajektorije u ovom radu će se izvršavati analizom putanje dobivene korištenjem DRDS-a. Nakon generiranja trajektorije bit će prikazana simulacija slijeđenja generirane trajektorije mobilnim robotom korištenjem analitičkog neizrazitog regulatora.

Prvotna ideja generiranja referentne trajektorije je bila korištenjem metode potencijalnih polja. Međutim, ta metoda je pokazala manjkavosti koje su se očitavale konstantnim zastajkivanjem robota u simulacijama. Razlog takvog ponašanja ove metode leži u činjenici da je generirana putanja često izrazito komplicirana te nije moguće postaviti atraktivni potencijal u ciljnu točku nego je potrebno prethodno postaviti cilj u niz alternativnih točaka te na posljetku u ciljnu točku. Alternativne točke su dobivene planiranjem putanje korištenjem DRDS-a. Zbog postavljanja atraktivnog potencijala u alternativne točke te približavanjem robota istima posljedično se usporava razmak između trenutne i konačne točke te robot sve sporije i sporije prilazi točci sve dok na posljetku ne stane u njoj, tek potom bi se uzimala nova sljedeća alternativna točka te bi robot nastavio gibanje.

Kako bi se riješili problemi zastajkivanja robota u svakoj alternativnoj točci potrebno je osmisliti način generiranja putanje koji je ovisan o vremenu. Predloženo rješenje se temelji na analizi točaka dobivenih pomoću DRDS-a. S obzirom na poznavanje rasporeda prepreka te poznavanje točaka kroz koje robot treba prolaziti moguće je sa sigurnošću utvrditi probleme koji bi se mogli pojaviti pri gibanju tom putanjom. Kako bi se generirala trajektorija robota analizira se prostorni položaj trenutačne i sljedeće točke robota. Analizom tih točaka gibanje robota se može podijeliti u dvije kategorije: gibanje robota paralelno sa koordinatnim osima te

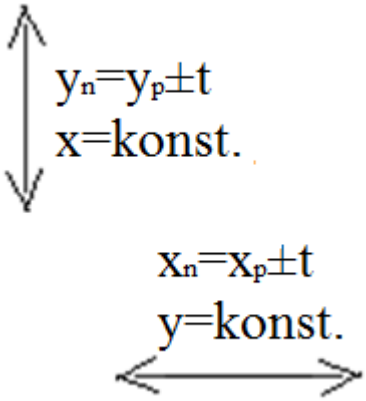
dijagonalno gibanje. Pri paralelnom gibanju s koordinatnim osima robot nije u opasnosti od udaranja u prepreke te nije potrebno koristiti dodatna sigurnosna rješenja. Generiranje referentne trajektorije pri ovakvom gibanju koristi izraz za običnu jednadžbom pravca gdje promjena ovisna o vremenu t. Ukoliko se robot giba paralelno sa osi x, vrijednost po y osi ostaje konstantna, a koordinate po osi x se mijenjaju po izrazu:

$$x_n = x_p \pm t\,, \tag{4.1}$$

gdje je x_n nova koordinata, x_p početna koordinata gibanja, a t je proteklo vrijeme. Predznak ispred promjene vremena ovisi o smjeru gibanja u odnosnu na os x. Analogno izrazu za gibanje paralelno osi x, moguće je zapisati izraz za gibanje paralelno s osi y:

$$y_n = y_p \pm t\,, \tag{4.2}$$

gdje je referentna vrijednost po x osi konstantna.



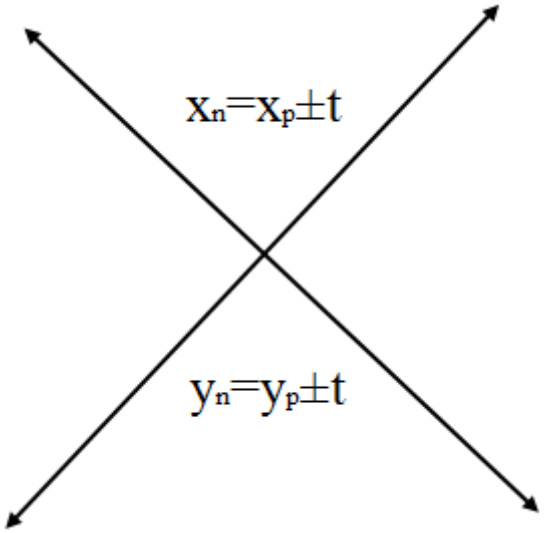
Slika 10: Horizontalno i vertikalno gibanje mobilnog robota

Pri dijagonalnom gibanju razlikujemo dvije vrste gibanja: gibanje oko prepreke i čisto dijagonalno gibanje. Pri čistom dijagonalnom gibanju generiranje referentne trajektorije se generira također pomoću jednadžbe pravca pri čemu se mijenjaju i x i y koordinate, pa izrazi za referentnu trajektoriju glase:

$$x_n = x_p \pm t$$

(4.3)

$$y_n = y_p \pm t$$



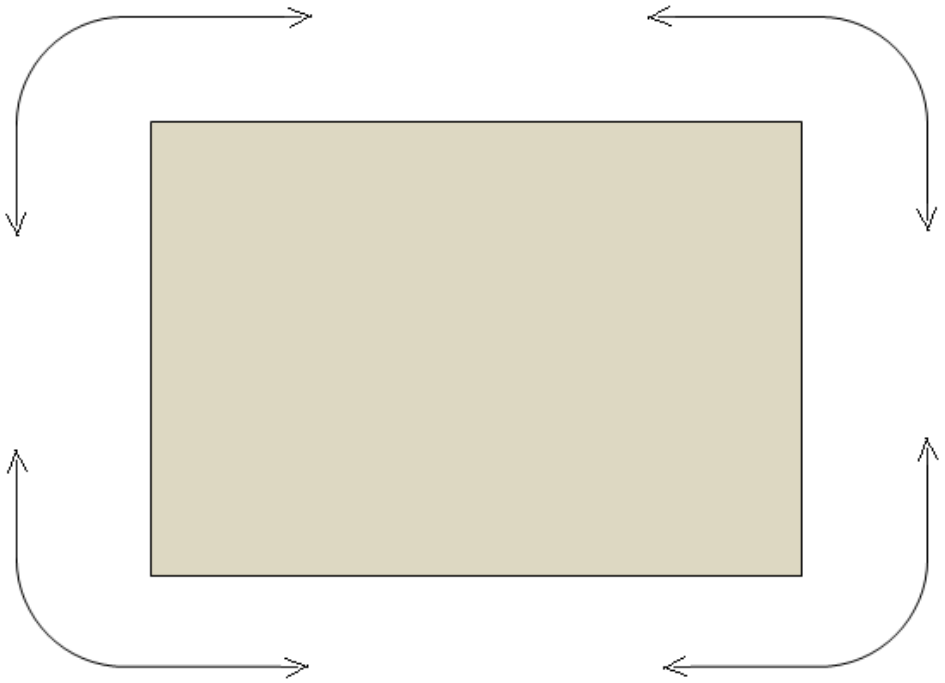
Slika 11: Dijagonalno gibanje mobilnog robota bez prepreke

Međutim, ukoliko se robot giba oko prepreke potrebno je tu prepreko zaobići kako ne bi došlo do kolizije robota i prepreke. Predloženo rješenje se temelji na kombiniranom gibanju po pravcu te na gibanju po kružnici, kako bi se izbjegla potreba za naglim skretanjem robota. Ovo gibanje je moguće podijeliti na tri djela. Prvi dio se odnosi na gibanje po pravcu paralelno sa nekom od osi sustava, drugi dio se odnosi na kružno gibanje kojim se zaobilazi prepreka te se treći dio odnosi na gibanje po pravcu koje je okomito na gibanje u prvom dijelu gibanja. Razlog ovakvog gibanja ležu u činjenici da se točke generirane DRDS-om često uvelike približavaju prepreci te postoji velika opasnost od kolizije sa preprekama pri dijagonalnom gibanju u blizini prepreke. Izrazi za prvi dio gibanja su istovjetni izrazima (4.1) i (4.2) te njihov izbor ovisi tome s kojom osi se robot paralelno giba. Izrazi za drugi dio gibanja su:

$$x_n = x_p \pm r \cdot \cos(2 \cdot \pi \cdot t) \text{ i}$$

$$y_n = y_p \pm r \cdot \sin(2 \cdot \pi \cdot t), \tag{4.4}$$

gdje je r radijus obilaženja prepreke. Drugi dio gibanja završava nakon što robot izvrši gibanje jedne četvrtine kružnice. Potom, robot počinje sa gibanjem koje je okomito na prvi dio gibanja pri čemu se koordinate referentne trajektorije također mijenjaju po izrazima (4.1) odnosno (4.2).



Slika 12: Gibanje mobilnog robota oko prepreke

5. Slijeđenje generirane referentne trajektorije korištenjem analitičkog neizrazitog regulatora

Nakon što je u prethodnom potpoglavlju prikazan način generiranja željene referentne, u ovom potpoglavlju će biti prikazan način izrade simulacijskog modela mobilnog robota upravljanog analitičkim neizrazitim regulatorom. Kako bi bilo moguće napraviti simulaciju kretanja mobilnog robota, potrebno je odrediti kinematički model mobilnog robota. U ovom radu je korišten jednostavan kinematički model mobilnog robota upravljanog sa dvije upravljačke veličine. Prva upravljačka veličina u_1 upravlja iznosom translacijske brzine, dok upravljačka veličina u_2 upravlja iznosom rotacijske brzine mobilnog robota. S obzirom da se u ovom radu razmatra kretanje mobilnog robota u x-y ravnini, kinematički model robota ima oblik:

$$\dot{x} = u_1 \cos(\theta), \tag{5.1}$$

$$\dot{y} = u_1 \sin(\theta), \tag{5.2}$$

$$\dot{\theta} = u_2, \tag{5.3}$$

gdje je θ kut zakreta mobilnog robota u odnosu na os x. Ovakav kinematički model mobilnog robota omogućava mu okretanje u mjestu, te samom tom mogućnošću uvelike olakšava kretanje u relativno uskim prostorima. Generiranom referentnom trajektorijom te korištenjem analitičkog neizrazitog regulatora određuju se upravljačke veličine u_1 i u_2.

5.1. Neizrazito upravljanje

Kako bi bilo moguće upravljati nekim procesom, potrebno je imati znanja o samom tom procesu. Ta znanja koja posjedujemo se mogu podijeliti u dvije kategorije: objektivno znanje, odnosno znanje koje je najčešće prezentirano matematičkim jednadžbama formiranim na temelju osnovnih fizikalnih zakona te subjektivno znanje, prezentirano lingvističkim

informacijama koje je gotovo nemoguće izraziti primjenom tradicionalnog matematičkog formalizma. Primjer objektivnog znanja o procesu su diferencijalne jednadžbe koje opisuju dinamičke procese u mehanici. Primjer subjektivnog znanja je sljedeće pravilo pri vožnji automobila: „Ako je nailazeće vozilo jako daleko i ako vozilo ispred vozi jako sporo onda treba dodati srednje jaki gas i preteći vozilo ispred sebe“. Kao što je vidljivo iz prethodnog primjera za upravljanje složenim tehničkim sustavom nije potrebno poznavanje njegovog matematičkog modela nego je potrebno poznavanje određenog skupa iskustvenih, heurističkih pravila [16]. Također, iz primjera je vidljiv sadržaj koji definiraju pravila, a iskazani su jezičnim formama poput „srednje“, „jaki“ i slično, te ih je nemoguće implementirati klasičnim algoritmima koji operiraju preciznim numeričkim podacima.

Ova problematika je inspirirala Lofti Zadeh 1965. godine da uvede koncept neizrazitih skupova. Zadeh je uočio da unatoč „načelu nezdruživosti složenosti i preciznosti“, ljudi imaju sposobnost analize i razumijevanja nepreciznih koncepata koje suštinski ne razumiju, npr. sposobnost vožnje automobila bez razumijevanja njegovog dinamičkog modela [16]. Problemi nastaju pri transformiranju tih znanja u numeričke vrijednosti. Zedeh je pretpostavio da je sposobnost analize i izvođenja uzročno-posljedičnih veza na temelju nepreciznih koncepata ključno svojstvo ljudskog misaonog procesa. S obzirom na nemogućnost numeričkog izražavanja uveo je teoriju neizrazitih skupova. Teorija neizrazitih skupova je značajan korak prema rješavanju problema ljudskog i računalnog jezika. Glavna značajka teorije neizrazitih skupova je njena mogućnost baratanja neodređenošću i višeznačnošću lingvističkih izraza te mogućnost preslikavanja takvih izraza u neizrazite skupove koji mogu biti interpretirani računalnim jezikom. Neizraziti skupovi predstavljaju nadogradnju klasične teorije skupova. Kod klasične teorije skupova element može ili pripadati ili ne pripadati skupu. U teoriji neizrazitih skupova element može imati određeni stupanj pripadnosti nekom skupu ili više skupova.

Pomoću klasične teorije skupova moguće je ocjenjivati kategoričke tvrdnje, te se taj način ocjenjivanja naziva klasična logika [16]. S druge strane, u neizrazitim skupovima je moguće ocijeniti i tvrdnje koje ne spadaju u kategoričke tvrdnje, te se takav način ocjenjivanja naziva neizrazita logika [16]. Teorija neizrazitih skupova i neizrazita logika čine osnovu neizrazitih sustava. Osnovu svih neizrazitih sustava čine tri postupka: omekšavanje ili fazifikacija (eng. fuzzyfication), inferencija ili proces odlučivanja te izoštravanje ili defazifikacija (eng. defuzzyfication).

Omekšavanje ili fazifikacija je proces transformiranja ulaznih izrazitih vrijednosti u stupnjeve pripadnosti ulaznih neizrazitih skupova [17]. Glavni dio ovog postupka je:

- određivanje ulaznih i izlaznih jezičnih varijabli,
- određivanje područja definicije navedenih varijabli,
- određivanje broja i oblika funkcija pripadnosti koje prekrivaju područje definicije pojedinih jezičnih varijabli.

Inferencija ili proces odlučivanja je proces transformiranja ulaznih neizrazitih skupova u neizrazite izlazne skupove. Sam proces odlučivanja se određuje iz baze pravila ponašanja zasnovane na obliku IF-THEN pravila.

Izoštravanje ili defazifikacija je proces transformiranja neizrazitih skupova u izlazne vrijednosti, kojima se upravlja procesom na način koji je definiran bazom pravila ponašanja.

Zbog postojanja baze pravila ponašanja kod klasičnih neizrazitih regulatora, javlja se problem eksponencijalnog rasta broja pravila ponašanja sa porastom ulazno-izlaznih varijabli stanja. Posljedica toga je da primjena klasičnih neizrazitih regulatora na multivarijabilne sustave poput robota, u uvjetima upravljanja u realnom vremenu, postaje vrlo zahtjevna sa stanovišta računalne kompleksnosti kao i kompleksnosti same sinteze regulatora [17].

Problem eksponencijalnog rasta broja pravila ponašanja s porastom ulazno-izlaznih varijabli sustava može se riješiti primjenom analitičkog neizrazitog regulatora ili neizrazitog regulatora bez pravila ponašanja [18]. Glavna značajka navedenog pristupa je definicija analitičke funkcije za određivanje centara izlaznih neizrazitih skupova, umjesto definicije pravila ponašanja [17]. Analitička funkcija omogućava direktan postupak nelinearnog preslikavanja ulaznih varijabli na centre izlaznih neizrazitih skupova, koji se jednostavno implementira u regulacijskom algoritmu [17]. S obzirom da ne postoji baza pravila ponašanja broj ulaznih i izlaznih varijabli te broj neizrazitih skupova nije ograničen eksponencijalnim rastom broja pravila ponašanja.

U ovom radu će biti korišten analitički neizraziti regulator koji će omogućiti praćenje referentne trajektorije mobilnog robota. Pomoću metode potencijalnih polja, konstantno se generira trajektorija koju neizraziti regulator pokušava slijediti. Kao što je već navedeno, preko upravljačkih varijabli se utječe na kretanje robota. Same upravljačke varijable će biti dobivene temeljem norme i vektorskog produkta robota. Norma je trenutačna vrijednost udaljenosti robota od referentne trajektorije, te se pomoću nje regulira translacijska brzina robota. Ukoliko je udaljenost robota u odnosu na referentnu trajektoriju prevelika, regulator povećava brzinu mobilnog robota kako bi došao na referentnu trajektoriju. Norma se računa preko sljedećeg izraza:

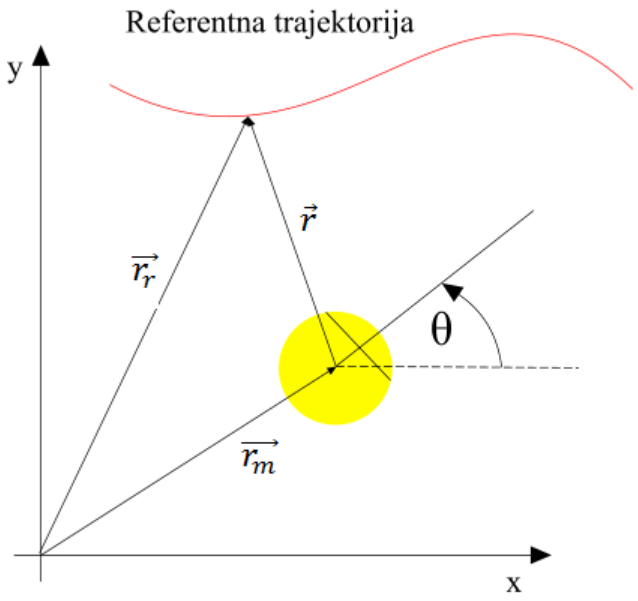
$$\|\vec{r}\| = \sqrt{\left(x_R - x\right)^2 + \left(y_R - y\right)^2} \,. \tag{5.4}$$

Vektorski produkt podrazumijeva vektorski produkt vektora $\vec{r_e}$ i $\vec{r}$ sa slike 13. U dobivenom izrazu predznak vektorskog produkta označava orijentaciju mobilnog robota u odnosu na referentnu trajektoriju. Samim vektorskim produktom utječe se na translacijsku rotacijsku brzinu. Međutim, prije same upotrebe vektorskog produkta za slijeđenje referentne trajektorije potrebno ga je normirati. Sljedeći izraz prikazuje računanje vektorskog produkta:

$$\vec{r}_e \times \vec{r} = (y_R - y)\cos\theta - (x_R - x)\sin\theta . \tag{5.5}$$

Kako bi se izračunao normirani vektorski produkt potrebno je vektorski produkt podijeliti s normom vektora:

$$V_{pn} = \frac{\vec{r}_e \times \vec{r}}{\left\|\vec{r}\right\|} . \tag{5.6}$$



Slika 13: Položaj mobilnog robota u odnosu na referentnu trajektoriju

Izračunavanjem norme i vektorskog produkta moguće je odrediti upravljačke veličine u_1 i u_2:

$$u_1 = v_{\max} \cdot \tanh(k_2 \cdot \left\|\vec{r}\right\|) \cdot e^{(-k_3 \cdot V_{pn})} , \tag{5.7}$$

$$u_2 = w_{\max} \cdot \tanh(k_1 \cdot V_{pn}) . \tag{5.8}$$

Odabirom ovakvih upravljački varijabli je osigurano da se robot pravilno orijentira u odnosu na referentnu trajektoriju, a potom da je i dostigne. Drugim riječima, ukoliko je robot na početku usmjeren suprotno od referentne trajektorije, imat će veliku rotacijsku brzinu i malu translacijsku brzinu kako bi se što prije usmjerio prema referentnoj trajektoriji, ali da se pritom ne udaljuje puno od cilja. Ukoliko je robot udaljen od cilja, a pravilno usmjeren, imat

će veliku translacijsku brzinu te malu rotacijsku brzinu. Ako je robot pravilno usmjeren te je blizu cilju, imati će malu rotacijsku i translacijsku brzinu kako ne bi promašio cilj, umjesto da se u njemu zaustavi.

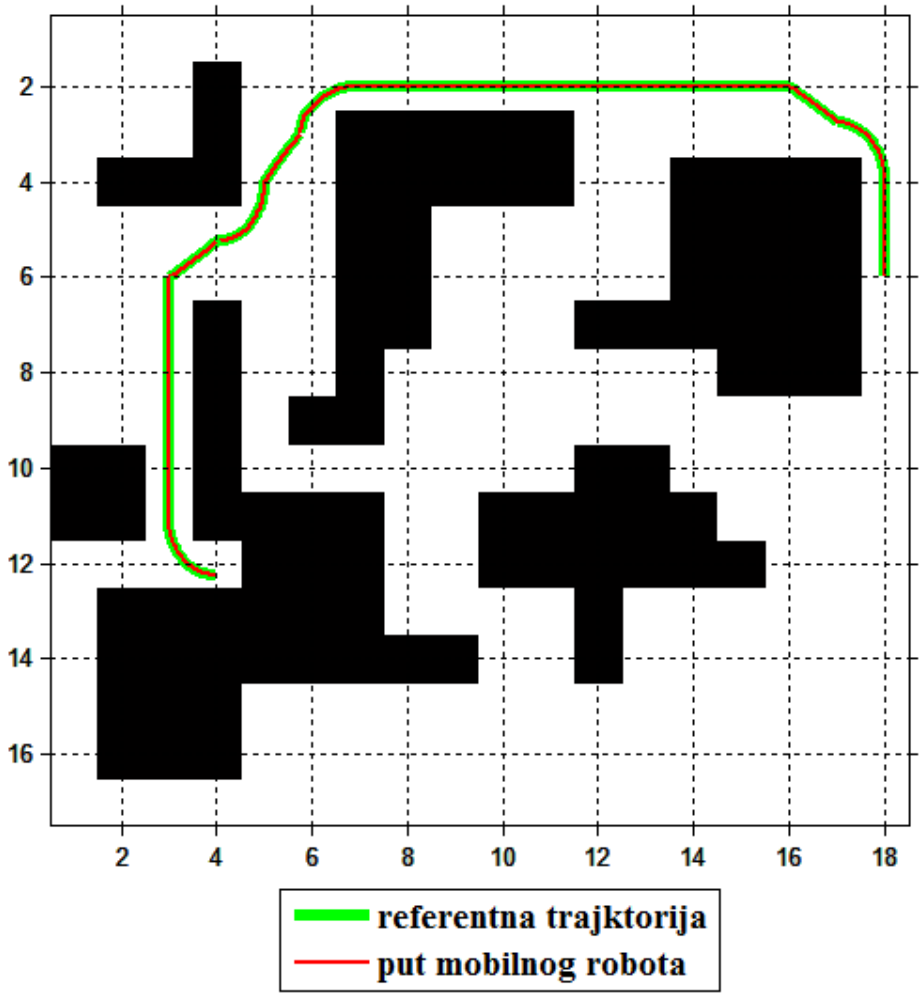
6. Simulacija gibanja robota zadanim prostom

U prethodnim poglavljima je prikaz način pronalaženja puta iz početne u konačnu točku pomoću modificiranog *podražajnog* DRDS-a, a potom je prikazan način generiranja referentne trajektorije u odnosu na dobivene točke kroz koje mobilni robot treba proći. Na kraju je prikazan način izrade analitičkog neizrazitog regulatora koji osigurava praćenje referentne trajektorije. U ovom poglavlju će biti prikazane simulacije gibanja mobilnog robota kroz prostore korištenjem zajedničkog djelovanja svih navedenih principa kako bi se osiguralo gibanje robota zadanim prostorom. Gibanja mobilnog robota koja će biti simulirana su gibanja s istih početnih u iste konačne pozicije kao što je prikazano u poglavlju 4.

Prije samog prikaza simulacije gibanja mobilnog robota kroz labirint, potrebno je objasniti neke detalje na slikama. Zelenom bojom je prikazana dobivena referentna trajektorija mobilnog robota korištenjem metode potencijalnih polja, a ona je rezultat generiranih točaka kroz koje mobilni robot mora prolaziti kako bi došao iz početne u konačnu poziciju. Te točke su generirane korištenjem modificiranog *podražajnog* DRDS-a. Crvenom bojom je prikazana simulacija gibanja mobilnog robota s analitičkim neizrazitim regulatorom pomoću kojeg robot prati željenu trajektoriju. U svim simulacijama, osim u prvoj, prikazane su po dvije slike. Lijeva slika prikazuje cijeli prostor kretanja mobilnog robota, a druga slika prikazuje uvećani prikaz jednog dijela prostora. Razlog ovakvog prikaza je u tome što su linije dodatno podebljane kako bi se što jasnije vidjelo njihovo prostiranje prostorom. Posljedica podebljanja je da na slikama koje prikazuju cijeli prostor izgleda kao da linije dodiruju ili presijecaju prepreke. Iz tog razloga je na desnoj slici prikazan uvećani dio labirinta te se na toj slici vidi da linije ne dodiruju i ne presijecaju prepreke.

6.1. Simulacija gibanja mobilnog robota prvim prostorom

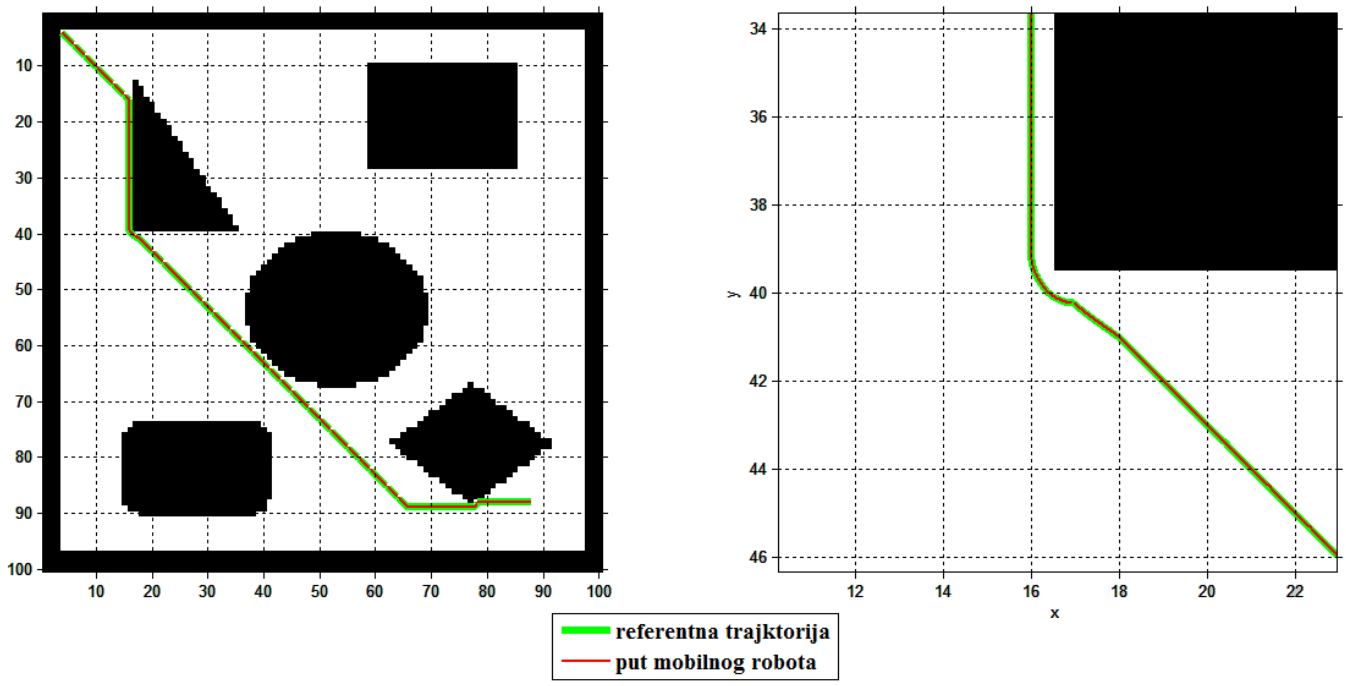
Početna i konačna pozicija mobilnog robota zadana je kao i prilikom generiranja puta pomoću DRDS-a. Početna pozicija je (18,6), a ciljna pozicija je (4,12). Iz same simulacije je vidljivo kako mobilni robot vjerno slijedi referentnu trajektoriju te kako zaobilazi sve prepreke bez opasnosti od kolizije s istima. Samo vrijeme trajanja simulacije koje uključuje generiranje referentne trajektorije i praćenja iste robotom traje između dvije i tri sekunde.



Slika 14: Simulacija gibanja mobilnog robota prvim prostorom

6.2. Simulacija gibanja mobilnog robota drugim prostorom

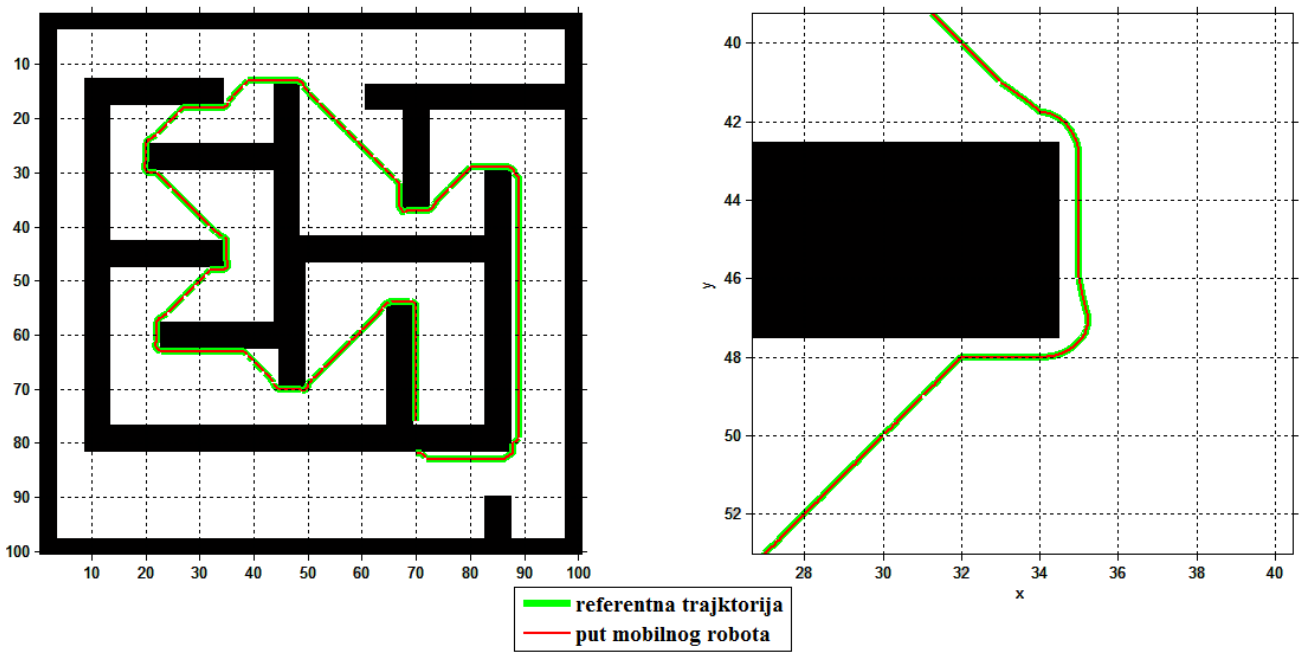
U drugom prostoru za testiranje za početnu točku je odabrana lokacija (4,4) te za ciljnu lokaciju točka (88,88). Vidljivo je da referentna trajektorije ne dolazi u koliziju sa preprekama te da sam mobilni robot ne dolazi u opasnost od udaranja u iste. Također, kao i u prethodnoj simulaciji mobilni robot vjerno slijedi referentnu trajektoriju. Samo vrijeme simulacije, odnosno generiranje trajektorije i kretanje mobilnog robota prostorom, iz početne u konačnu točku je trajalo između osam i devet sekundi.



Slika 15: Simulacija gibanja mobilnog robota drugim prostorom

6.3. Simulacija gibanja mobilnog robota trećim prostorom

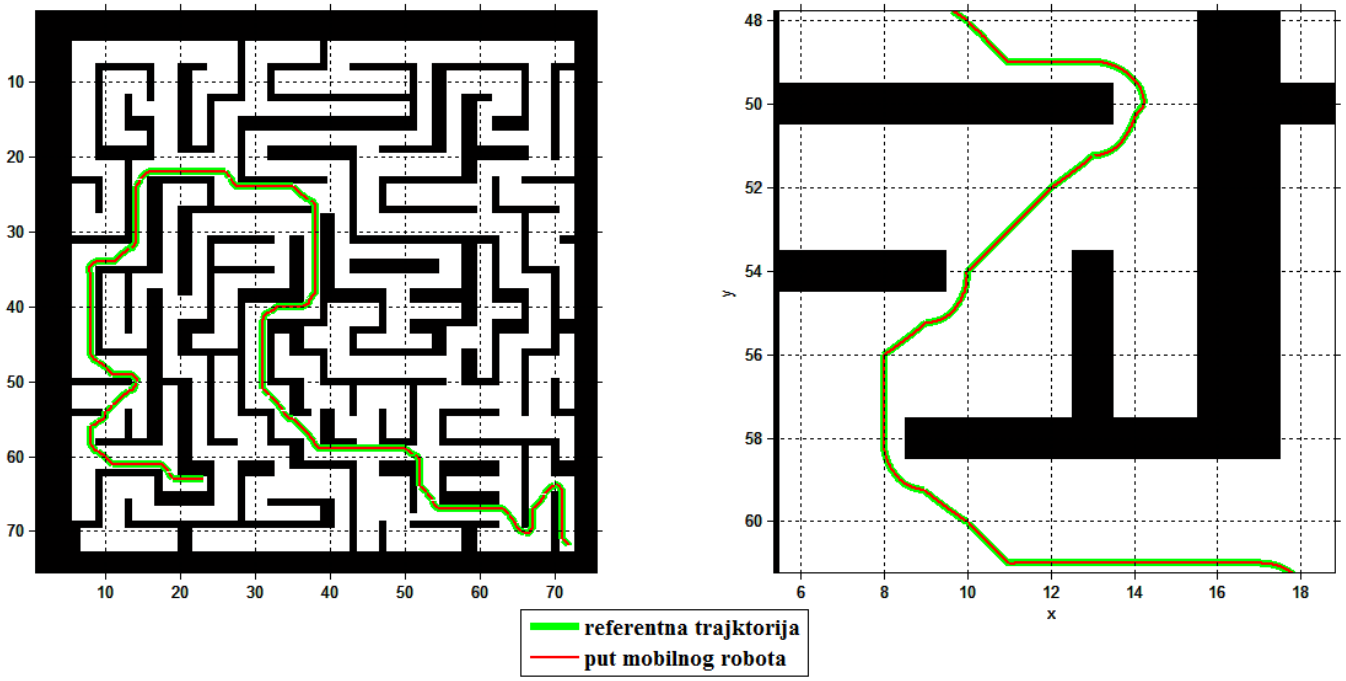
U trećem labirintu za početnu točku je odabrana pozicija (70,76), a za ciljnu točku je odbrana pozicije (70,82). Kao i u prethodnim simulacijama mobilni robot vjerno slijedi referentnu trajektoriju te ni u jednom trenutku ne dolazi u opasnost od sudara s preprekama. Sama simulacija gibanja mobilnog robota je trajala između 18 i 19 sekundi.



Slika 16: Simulacija gibanja mobilnog robota trećim prostorom

6.4. Simulacija gibanja mobilnog robota četvrtim prostorom

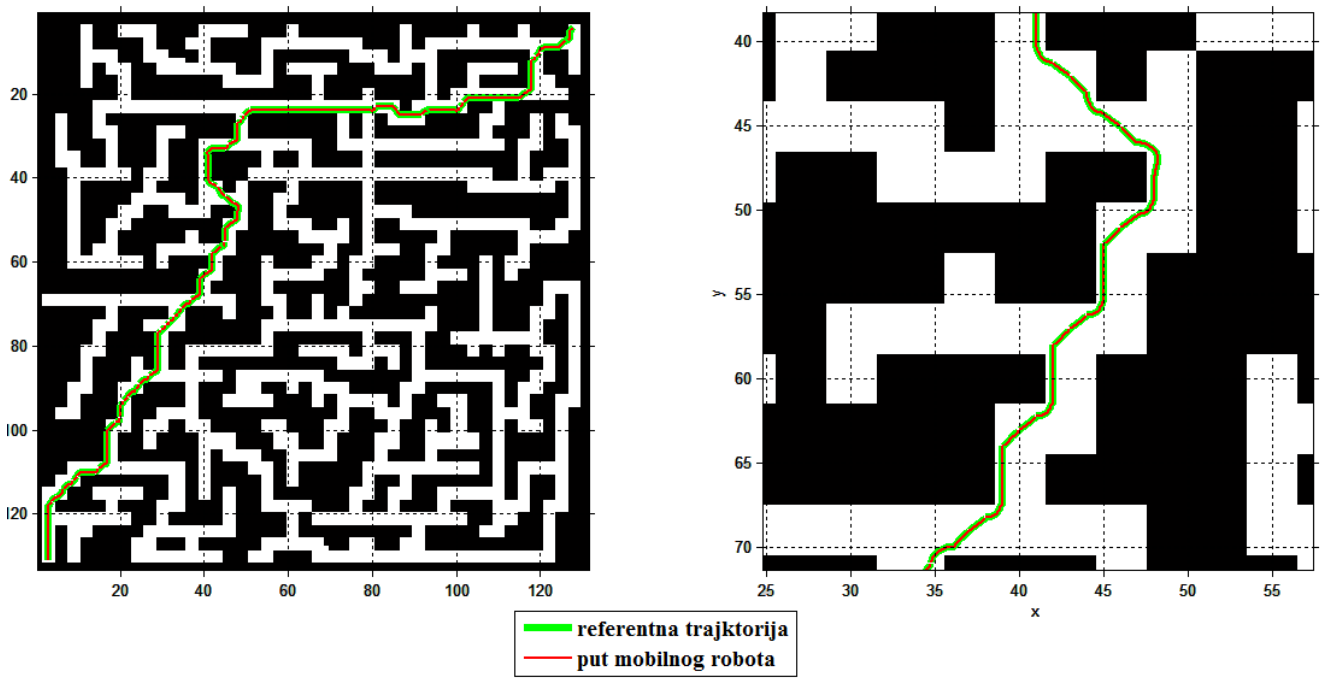
U četvrtom testiranom prostoru za početnu točku je odabrana pozicija (72,72), a ciljna točka je (23,63). Kao i u prethodno testiranim prostorima za simulaciju, generirana trajektorija zaobilazi prepreke na sigurnoj udaljenosti, te nema opasnosti od kolizije robota i prepreka. U ovom prostoru testiranja posebno dolazi do izražaja predloženi algoritam planiranja puta robota, ali i sigurnost mobilnog robota od sudara sa preprekama. Simulacija kretanja mobilnog robota kroz ovaj prostor je trajala između 11 i 12 sekundi.



Slika 17: Simulacija gibanja mobilnog robota četvrtim prostorom

6.5. Simulacija gibanja mobilnog robota petim prostorom

Posljednji prostor u kojem je izvršena simulacija je i najveći prostor. Kao početna točka u simulaciji je odabrana pozicija (128,4), a ciljna točka je (3,131). Kao i u prethodnoj simulaciji, iz slika je vidljiva složenost prostora u kojem je izvršena simulacija. Dobiveni rezultati simulacije govore u prilog predloženoj metodi. Iz slike 16 je vidljivo kako mobilni robot i u ovom slučaju zaobilazi sve prepreke na sigurnoj udaljenosti te ne dolazi u opasnost od udaranja u iste. Također je vidljivo da, iako je moguće na više načina doći iz početne u konačnu poziciju, korištenjem predloženog algoritma osiguran je pronalazak najkraće putanje kroz prostor. Simulacija u ovom prostoru je trajala između 16 i 17 sekundi.



Slika 18: Simulacija gibanja mobilnog robota kroz peti prostor

7. Vizijski sustavi

Vizijski sustavi su sustavi koji omogućavaju korištenje kamere u svrhu optičkog, beskontaktnog i automatskog primanja informacija o prizoru te njegove daljnje obrade. Vizijski sustavi se sastoje od kamere, uređaja za komunikaciju, računala, programa za obradu digitalne slike i drugo. Digitalna obrada slike predstavlja skup metoda za obradu slike pomoću računala. Slike koje se obrađuju mogu imati različito porijeklo. Osim sa kamere, slike mogu potjecati i sa satelita i drugih letjelica, uređaja za medicinsku dijagnostiku, memorijskih medija za pohranu, radara te mnogih drugih uređaja.

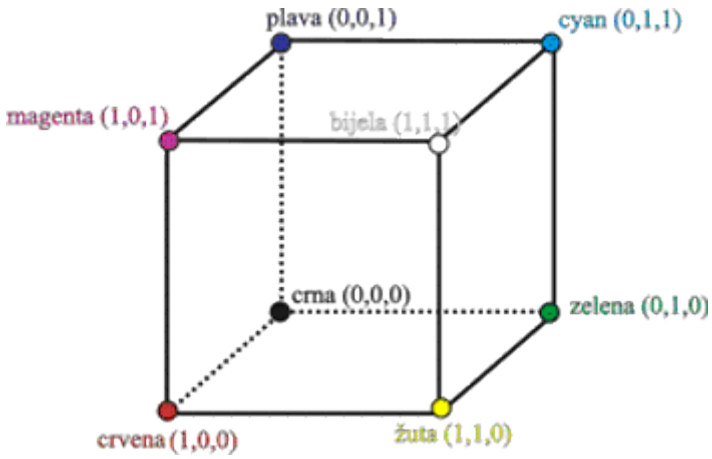
7.1. Digitalna slika

Ranih 1920-tih godina počela prva primjena digitalnih slika u novinskoj industriji, gdje je prva slika poslana prekomorskim kablom između Londona i New Yorka [19]. Ovim postupkom se skratilo vrijeme putovanja slike sa više od tjedan dana na manje od tri sata. Prvi sustavi digitalne fotografije su bili sposobni koristiti pet razina sive boje, do 1929. godine taj broj se povećao na 15 razina [21]. Iako su se ove slike smatrale digitalnim one nisu to bile u današnjem smislu digitalne slike jer u njihovom stvaranju nije sudjelovalo računalo [19]. Iako je prvo digitaliziranje slika nastalo prije samog računala sama digitalna obrada slike se počela razvijati tek njihovom pojavom. Tek 1960-tih godina su se pojavila prva računala koja su bila sposobna izvršavati osnovne zadatke digitalne obrade slike [19]. Razlog toga je što su digitalne slike zahtijevale mnogo prostora za pohranu te velike brzine procesora. Od 1960-tih sve do danas razvoj digitalne obrade slike nezaustavljivo raste. Digitalnu obradu slike tako danas susrećemo u mnogim područjima ljudske djelatnosti kao što su medicina, industrija, promet i drugo.

Od samih početaka razvoja digitalnih slika želja je analizirati slike te pomoću analize saznati informacije o samoj slici. Lako može doći do zablude i mišljenja kako je to ustvari

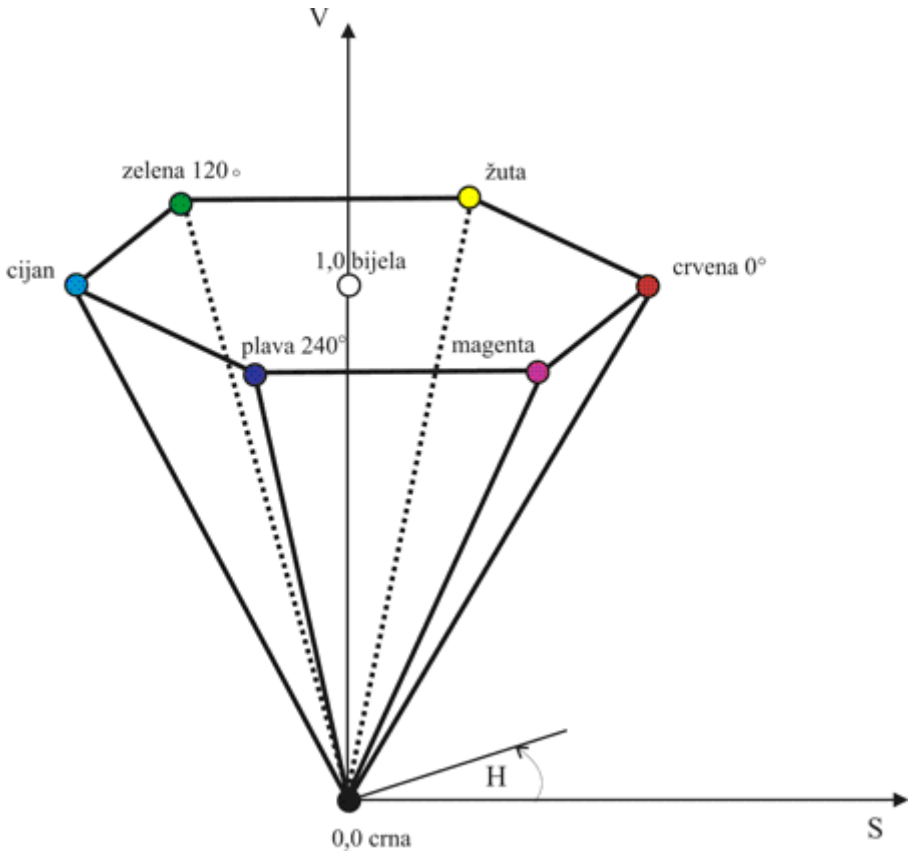
jednostavno, jer smo mi ljudi vizualna bića te jednostavno možemo analizirati sliku koju vidimo vlastitim očima. Razlog jednostavne i brze analize slike leži u činjenici da su se naši organi za vid, oči, i procesiranje slike, mozak, evolucijski razvijali milijunima godina te su na taj način dovedeni gotovo do savršenstva. Ljudski mozak dijeli vidni signal na mnogo kanala koji prenose različite informacije u naš mozak. Mozak ima sustav koji utvrđuje bitne dijelove na slici dok ostale dijelove potiskuje. Nažalost, nije moguće takav kompliciran sustav prenijeti na računalo te pomoću njega obavljati takve operacije.

Kako bi računalu bilo moguće raspoznavati što se nalazi na slici potrebno je prvo znati kako računalo vidi tu sliku. Slika je na računalu definirana kao dvodimenzionalna funkcija f(x,y), gdje su x i y prostorne koordinate, a amplituda te funkcije je razina jakosti u toj točci. Digitalna slika je sastavljena od konačnog broja x i y koordinata te od konačne razine amplitude. Element određenog prostornog položaja i određene amplitude se naziva piksel, te je on osnovni dio svake slike. Kod prikaza crno-bijelih slika svaki piksel sadrži samo jednu komponentu čija je vrijednost proporcionalna osvjetljenju. Takav prikaz slike se još naziva i gray-scale. Ukoliko se radi o slici u boji svakom pikselu se pridružuje više komponenti, a najčešće tri. Neki od više komponentnih zapisa slike su RGB, HSV, HSL, CMYK. RGB model boja je najčešće korišteni zapis. Kada se govori o bojama na zaslonu monitora, ustvari se misli na RGB zapis. RGB znači R-crveno (eng. Red), G-zeleno (eng. Green) i B-plavo (eng. Blue). Drugim riječima, svaka boja na zaslonu je prikazana nijansom crvene, zelene i plave boje. Svaka od tri boje može varirati od 0-100%. Ukoliko su sve tri boje na 0% na zaslonu je crna boja, a ukoliko su na 100% na zaslonu je bijela boja.



Slika 19: RGB model boja

RBG model boje je jednostavan za računalo, ali nije prikladan za čovjeka. Crvena, zelena i plava komponenta međusobno su korelirane tako da je čovjeku vrlo teško izborom tih komponenata definirati željenu boju u RGB modelu boja [20]. Stoga se najčešće u obradi slika koriste drugi modeli boja kao što su HSV ili HSL. U ovom radu će biti korišten HSV model. Ovaj model se definira u cilindričnom koordinatnom sustavu. Boje su prikazane u podskupu prostora omeđenom šesterostranom piramidom [20]. Vrh piramide je u ishodištu i odgovara crnoj boji. Vertikalna os V određuje sjajnost boje (eng. value). Baza piramide je na razini V=1 što odgovara skupu sjajnih boja. Nijansa (eng. hue) određena je kutom zakreta H oko vertikalne osi V. Kutovi zakreta komplementarnih boja razlikuju se za 180°. Radijalna udaljenost od osi V određuje razinu zasićenosti boje (eng. saturation). Korištenjem HSV modela boja moguće puno lakše i jednostavnije traženje pojedinih boja na slici.



Slika 20: HSV model boja

Najčešće se digitalne slike dobivaju sa digitalnih fotoaparata ili kamera. Tako dobivene slike mogu sadržavati razne distorzije. Distorzije nastaju kao posljedica različitog povećanja predmeta u različitim točkama na slici, pri čemu je bitno naglasiti da se distorzijom ne gube nikakvi podaci sa slike. Postoje dvije vrste distorzija, a to su bačvasta i jastučasta. Idealna bi situacija bila kad bi zaslon propuštao cijeli snop svjetlosnih zraka na isti način kao i centralnu zraku. Centralna je zraka ona koja prolazi kroz središte leće i ne otklanja se prilikom prolaza. No u stvarnosti nije tako, rubni se snopovi koji prolaze blizu ruba zaslona otklanjaju. Ukoliko se zaslon nalazi ispred leće, zrake skreću prema centru. Rubovi slike, a pogotovo kutovi, jer je kod njih put zrake najduži, ugibaju se prema unutra pa stranice snimljenog objekta djeluju kao da su izbočene, te se ta vrsta distorzije naziva bačvastom. Ova vrata distorzije se najčešće susreće kod korištenja širokokutnih objektiva kojima se želi postići visoki kut gledanja sa male udaljenosti od kamere.

7.2. Digitalna obrada slike

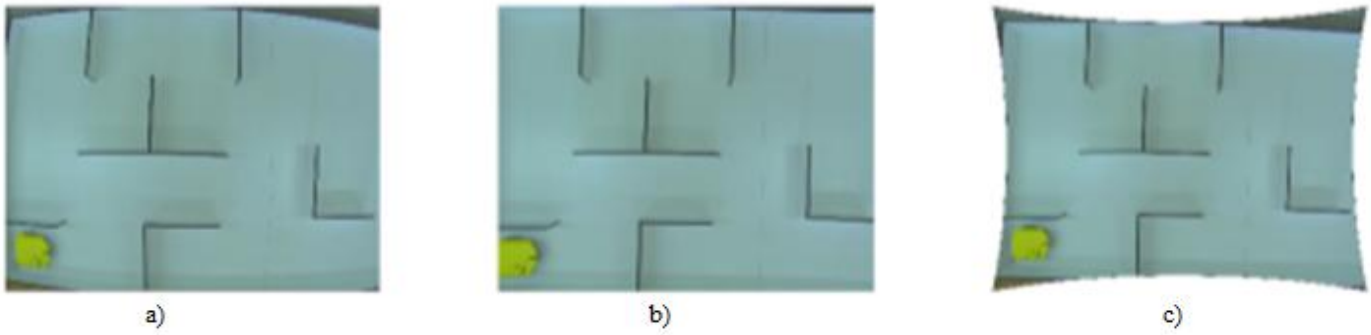
Digitalna obrada slike ili vizija predstavlja skup metoda za obradu slike pomoću računala. U viziji se potrebno služiti raznim trikovima (pretpostavkama, heuristikom i specifičnim znanjima za analizirani prizor) da bismo iz prizora izdvojili željene karakteristike. Kako bi se postiglo izdvajanje željenih karakteristika potrebno se služiti različitim postupcima kako što su tražnje rubova objekata na slici, segmentacija, micanje određenih boja i drugo.

U ovom radu vizija se koristi u planiranju gibanja mobilnog robota kroz labirint te kontrole željenog gibanja robota. Kako bi bilo moguće koristiti sliku dobivenu sa kamere koja se nalazi iznad labirinta potrebno je sliku digitalno obraditi. Digitalna obrada slike se izvršava u nekoliko koraka. U prvom koraku je potrebno sliku preuzeti sa kamere na računalo. U drugom koraku je potrebno ispraviti nastale distorzije zbog nedovoljne udaljenosti kamere od labirinta. Potom je potrebno locirati robota te na slici izolirati prepreke s kojima bi robot mogao doći u koliziju pri gibanju. Zatim je potrebno prilagoditi sliku labirinta kako bi se bila upotrebljiva za korištenje DRDS metode planiranja putanje. Na kraju je potrebno pratiti gibanja robota kroz labirint kako bi se moglo upravljati translacijskim i rotacijskim brzinama robota korištenjem analitičkog neizrazitog regulatora. Sve izvršene operacije u ovom radu su rađene u programskom paketu MATLAB.

7.2.1. Ispravljanje distorzije na slici

Kako bi obrada slike bila moguća sa zadovoljavajućom preciznošću potrebno je dobivenu sliku sa kamere ispraviti zbog bačvaste distorzije slike. Razlog distorzije slike leži u činjenici da se kamera nalazi preblizu labirintu te se koriste široko kutne leće kako bi se snimio što veći prostor. Kao što je već navedeno distorzijom slike se ne gube informacije o prizoru, samo se pojavljuju različita povećanja, odnosno smanjenja objekata na slici u odnosu na centar slike. Distorzijske pogreške je moguće ukloniti programski. Kako bi se slika

ispravila prvo je potrebno postaviti središte koordinatnog sustava slike u centar slike. Potom je potrebno kartezijski koordinatni sustav transformirati u polarni koordinatni sustav. Nakon toga je potrebno primijeniti određene jednadžbe kako bi se dobilo željeno ispravljanje slike. Te je na kraju potrebno vratiti sliku u kartezijski koordinatni sustav. Tako ispravljena slika se može koristiti za daljnju obradu slike. U ovom radu je korišten preuzeti MATLAB kod koji učitava sliku te se odabire koeficijent za koji želimo ispravljati sliku [21]. Naredba koja se koristi je lensdistort. Korištenjem ove naredbe moguće je odabrati dvije opcije ispravljanja. Prva i zadana opcije je „crop“. Korištenjem ove opcije slika se ispravlja te se dodatno pikseli proširuju u ovisnosti o udaljenosti od centra slike, čime se postiže da isti objekti imaju istu površinu na krajevima slike kakvu bi imali da se nalaze na sredini slike. Širenje se odvija po istom koeficijentu kao i ispravljanje slike. Širenjem piksela ispravljena slika bi imala više piksela od početne pa se slika još dodatno smanjuje na početnu veličinu. Time nestaju rubovi slike, ali je slika puno realnija u odnosu na stvarnost. Drugi način ispravljanja slike je „fit“. Na ovaj način se slika ispravlja po zadanom koeficijentu, slika ostaje iste veličine kako i neispravljena slika, ali bez povećanja piksela na rubovima slike.



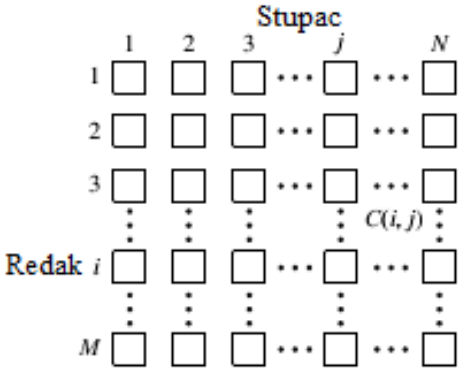
Slika 21: Ispravljanje slike korištenjem naredbe lensdistort: a) neispravljena slika, b) ispravljena slika opcijom „crop“ , c) ispravljena slika opcijom „fit“

U ovom radu je korištena opcija „crop“ za ispravljanje slika te koeficijent ispravljanja -0,18. Ova naredba u kodu izgleda: lensdistort(I, -0,18), gdje je I slika koju želimo ispraviti. Ispravljanjem slike moguća je daljnja obrada slike u svrhu lociranja mobilnog robota.

Obzirom na da dobivena slika veličine 1234 x 1624 piksela slika se smanjuje na 0.15 % prvotne veličine. Razlog velikog smanjenja je duljina trajanja lociranja robota. Nakon smanjenja slike potrebno je povećati kontrast slike. Povećanje kontrasta se izvrši na način da se slika prebaci u HSV model boja te se matrice S i V MATLAB-ovom naredbom imadjust prilagode te se potom slika vrati u RGB zapis. Ovime je poboljšan kontrast slike te je moguća daljnja analiza slike. Nakon ispravljanja slike i povećanja kontrasta pristupa se segmentaciji slike te lociranju robota i prepreka na slici. Za segmentaciju slike te prilagodbu iste za korištenje DRDS metode traženja putanje u ovom radu će biti korištene celularne neuronske mreže.

7.2.2. Celularne neuronske mreže (Cellural neural network- CNN)

Celularne neuronske mreže (CNN) prvi su predstavili Leon O. Chua i Lin Yang 1988 u Berkeley-u. Osnovna građevna jedinica CNN je ćelija. U početku su te ćelije bile analognog tipa te su se sastojale od kondenzatora, nelinearnog naponski kontroliranog izvora struje i nekoliko linearno ovisnih otpora [22]. Daljnjim razvijanjem CNN je ušao u digitalnu upotrebu te je danas moguće njihovu dinamiku programirati i na računalima. U ovom radu je programirana digitalna dinamika CNN. CNN su zamišljene za masovno paralelno izvođenje operacija te obrađivanja podataka u realnom vremenu. Nakon prvog predstavljanja sve do današnjeg dana na temu CNN-a je objavljeno preko 1000 radova te je CNN korišten u mnogim područjima [23]. CNN se najčešće koriste za obradu slike, odnosno u segmentaciji slike. Uobičajena arhitektura CNN sadrži M x N pravokutnu matricu *u*, sa standardnim kartezijskim koordinatama. Sama matrica ovisi o veličini slike koja se obrađuje, odnosno o broju piksela slike, svaki piksel postaje svojevrsnom ćelijom koja se označava s *C(i,j).* gdje su *i* i *j* koordinate retka odnosno stupaca matrice [23]. Na slici 22 prikazan je izgled ćelija CNN-a.

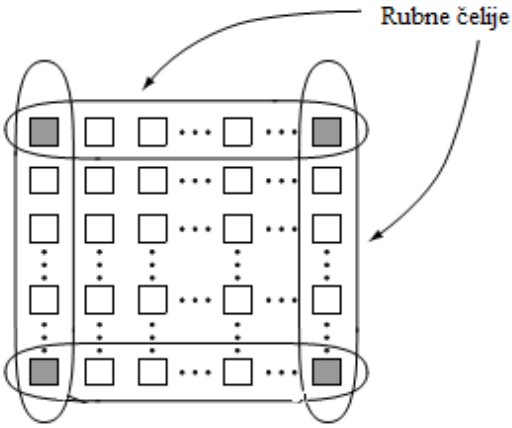


Slika 22: Prikaz ćelija CNN-a

Potrebno je naglasiti kako M i N ne trebaju biti jednakih dimenzija . Prvi korak koji je potrebno napraviti je prilagođavanje slike. Sliku zapisanu u RGB formatu je potrebno transformirati u crno-bijelu sliku. Potom je potrebno izvršiti normiranje slike na način da bijeli pikseli slike poprime vrijednost -1, a crni pikseli vrijednost 1, ostali sivi dijelovi slike poprimaju vrijednosti između -1 i 1 u ovisnosti o intenzitetu bijele, odnosno crne boje. Jedna od razlika ovih neuronskih mreža u odnosnu na klasične neuronske mreže je što svaka ćelija komunicira samo sa svojim susjednim ćelijama, druga razlika je to što celularne neuronske mreže nemaju proces učenja. Način komunikacije samo sa susjednim ćelijama se još naziva područje utjecaja ćelije te se označava sa S_r. Samo područje utjecaja je definirano kao radijus utjecaja *r* koji označava koliko susjednih ćelija utječe na ponašanje pojedine ćelije [22]. Najčešće korišteni radijus utjecaja je *r=1* odnosno područje utjecaja $S_r(3x3)$, također moguće je koristi i ostala područja utjecaja kao što je $S_r(5x5)$ pri čemu je radijus utjecaja jednak 2.

Slika 23: Prikaz radijusa utjecaja CNN-a

Također potrebno je naglasiti da postoje dvije vrste ćelija. Prva vrsta su takozvane regularne ćelije, odnosno ćelije koje imaju sve susjedne ćelije, a druga vrsta su rubne ćelije. Kao što im smo ime govori te ćelije se nalaze na samim rubovima te kao takve nemaju sve susjede. Do problema sa rubnim ćelija dolazi zbog samog područja utjecaja ćelije. Stoga je najprije potrebno matricu *u* okružiti sa virtualnim ćelijama čija je početna vrijednost jednaka 0. Vrijednosti virtualnih ćelija se nikada ne računaju nego one samo služe kao neutralni susjedi rubnim ćelijama.

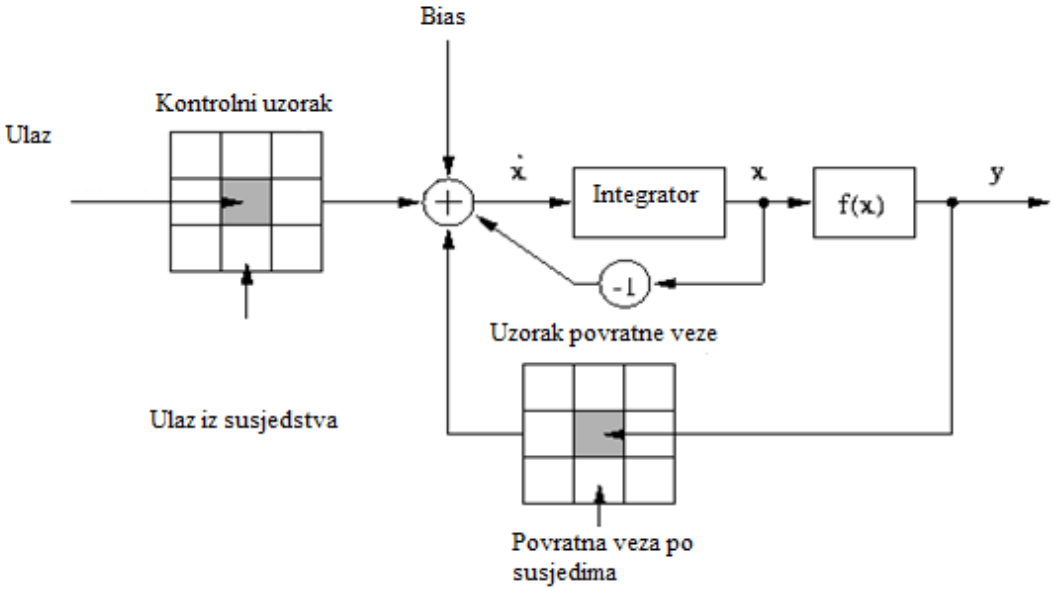


Slika 24: Rubne ćelije CNN-a

Stanje svake ćelije je definirano matematički diferencijalnom jednadžbom [13]:

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l)\cdot y_{k,l} + \sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l)\cdot u_{k,l} + z_{ij} \ , \quad (7.1)$$

gdje su x_{ij}, y_{kl}, z_{ij}, **A**(i,j;k,l) i **B**(i,j;k,l) redom definirani kao trenutno stanje, izlaz ćelije, prag ćelije, povratna veza ćelije i kontrolni operator. Blokovski prikaz stanje ćelije je prikazan na slici 25:



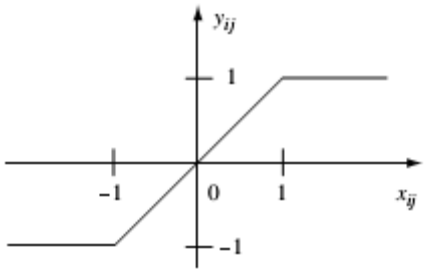
Slika 25: Blokovski prikaz jedne ćelije CNN-a

Matrice **A**, **B** te vrijednost praga *z* mogu varirati i o vremenu i o samim koordinatama ćelija, međutim najčešće su konstantni parametri . Odabirom vrijednosti parametara matrica **A** i **B** i praga *z* određuje se ponašanje same CNN. Neki od primjera odabira tih vrijednosti se mogu odabrati iz predloška za odabir vrijednosti [24], ali i vrijednosti se mogu odabrati i proizvoljno. Ovisno o odabranim vrijednostima matrica **A** i **B** i praga *z* može se postići pronalazak rubova, povećavanje debljine linija, uklanjanje malih površina i drugo.

Izlazna jednadžba ćelije je definirana jednadžbom:

$$y_{ij} = f(x_{ij}) = \frac{1}{2} \cdot \left| x_{ij} + 1 \right| - \frac{1}{2} \cdot \left| x_{ij} - 1 \right|, \tag{7.2}$$

gdje je x_{ij} trenutna vrijednost ćelije.



Slika 26: Izlazna funkcija CNN-a

7.2.2.1. Integracija diferencijalne jednadžbe CNN

Kako bi se moglo izračunati stanje svake ćelije u koraku t+1 diferencijalnu jednadžbu je potrebno integrirati te zapisati u obliku:

$$x(t+\Delta t) = x(t) + \int_{t_0}^{t} f(x(\tau))d\tau \,. \tag{7.3}$$

Rješenje integrala na desnoj strani gornje jednadžbe je moguće numeričkim metodama kao što je Euler-ova integracije, Runge-Kutta i dr. Najjednostavnija i za procesor vremenski najbrža je Euler-ova integracije, međutim Euler-ova integracija unosi i najveću pogrešku računanja [23]. U ovom radu je odabrano korištenje Euler-ove integracije zbog što kraćeg korištenja procesorskog vremena. Euler-ovom integracijom sljedeća jednadžba poprima oblik:

$$x(t+\Delta t) = x_{ij} + \Delta t \cdot (-x_{ij} + \sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l)\cdot y_{k,l} + \sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l)\cdot u_{k,l} + z_{ij}) \,. \tag{7.4}$$

Sređivanjem gornje izlaza dobiva se konačna jednadžba stanja ćelije:

$$x(t+\Delta t) = (1-\Delta t)\cdot x_{ij} + \Delta t \cdot \sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l)\cdot y_{k,l} + \Delta t \cdot \sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l)\cdot u_{k,l} + \Delta t \cdot z_{ij} \,. \tag{7.5}$$

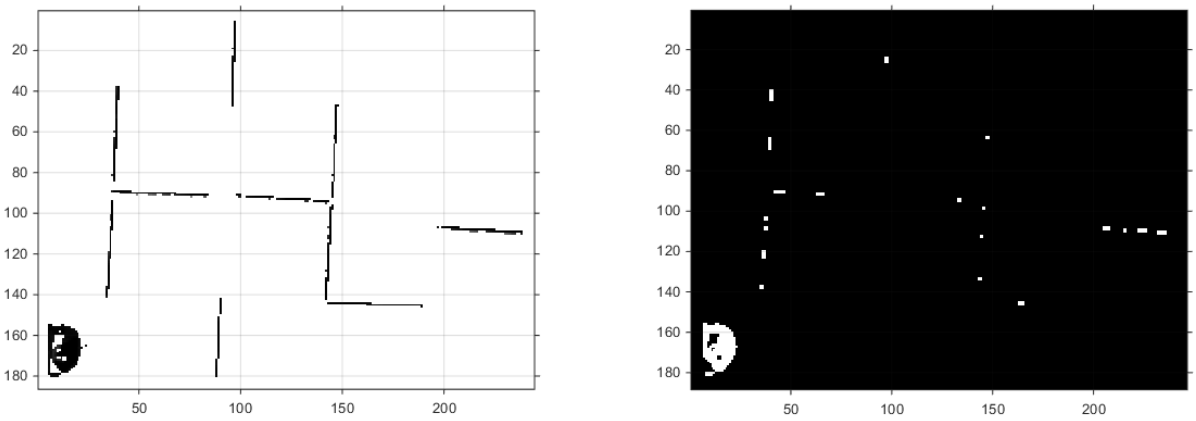
7.2.2.2. Korišteni uzorci u ovom radu

U ovom radu je korišteno nekoliko uzoraka ponašanja CNN. Neki koraci su korišteni jedni iz drugih, dok su neki korišteni odvojeno. Predloženi uzorci su preuzeti iz knjige sa uzorcima sa sveučilišta u Budimpešti te ih se može pronaći u literaturi [24].

7.2.2.2.1. Micanje tankih linija sa slike

Korištenjem ovog uzorka matrice ***A*** i ***B*** te prag *z* se podešavaju na način da se sve linije veličine jednog piksela uklanjaju sa slike. Korištenje ovog uzorka omogućuje micanje većine piksela koji predstavljaju prepreke koje se nalaze u zadanom prostoru. Matrica **A** ima oblik:

$$\mathbf{A} = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 8 & 2 \\ 2 & 2 & 2 \end{bmatrix},$$ matrica **B** je nul matrica dimenzija 3x3, a prag *z* je jednak -2.

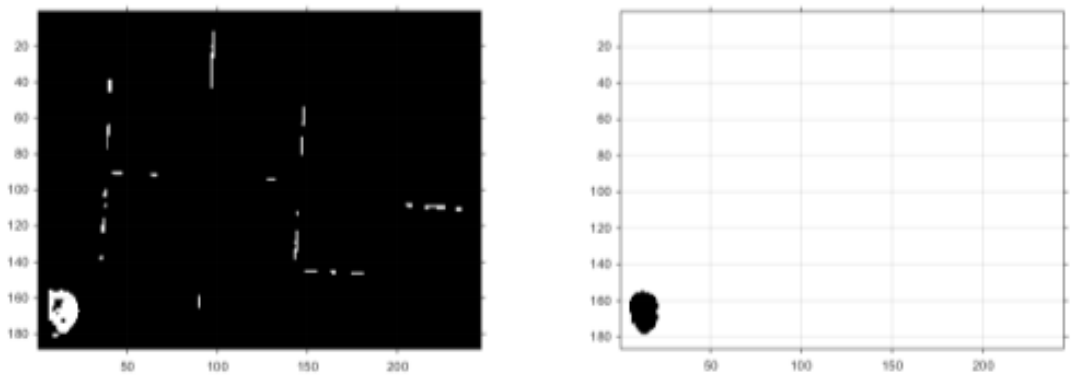


Slika 27: Micanje otvorenih kontura na slici

7.2.2.2.2. Micanje malih površina sa slike

Korištenjem ovog uzorka marice ***A*** i ***B*** te prag *z* se podešavaju na način da se uklanjaju svi sitni objekti koji bi se mogli pojaviti zbog smetnji koje nastaju smanjivanjem slike.

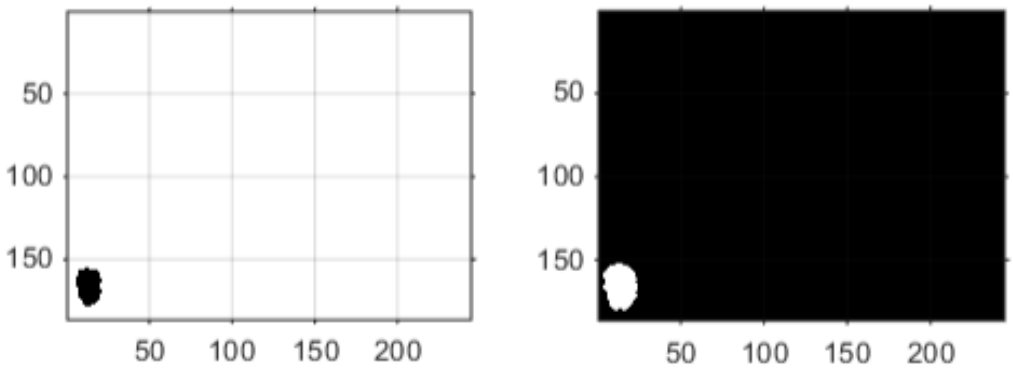
Matrica **A** ima oblik: $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, matrica **B** ima oblik: $\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, a prag *z* je jednak 3,25.



Slika 28: Micanje zaostalih malih površina sa slike

7.2.2.2.3. Popunjavanje konkavnih površina

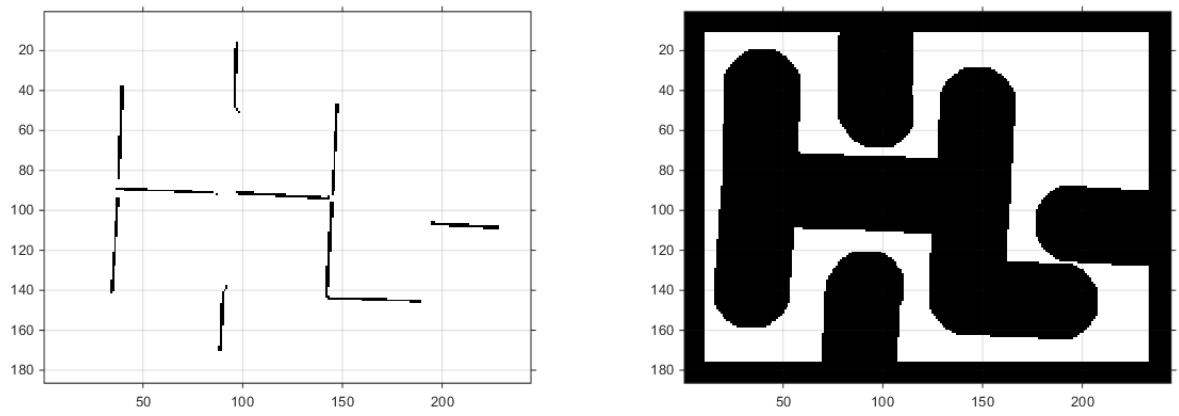
Zbog mogućih pogrešaka u lociranju oblika robota uključen je ovaj predložak. On popunjava sve konkavne površine na slici. Ovime se postiže ispravljanje površinskih nesavršenosti te se osigurava nesmetano uklanjanje robota sa slike nakon što je robot lociran. Uklanjanje robota se izvršava kako bi na slici ostale samo prepreke te kako bi se mogla planirati putanja mobilnog robota početne u konačnu točku. Matrica **A** u ovom predlošku ima oblik: $\mathbf{A} = \begin{bmatrix} 0,5 & 0,5 & 0,5 \\ 0,5 & 3 & 0,5 \\ 0,5 & 0,5 & 0,5 \end{bmatrix}$, matrica **B** je nul matrica dimenzija 3x3, a prag *z* je jednak nula.



Slika 29: Popunjavanje konkavnih površina

7.2.2.2.4. Povećanje površina na slici

Zadnji korišteni uzorak CNN je povećanje površina na slici. Ovaj uzorak se koristi kako bi se slika labirinta prilagodila korištenju DRDS metode planiranja putanje. Kako je pokazano u trećem poglavlju u DRDS metodi robot je predstavljen veličinom jednog piksela. Problem koji se javlja u ovom radu je to što je robot većih dimenzija te ga nije moguće predstaviti samo jednim pikselom. Kako bi bilo moguće izvršavanje DRDS ideja ovog rada je virtualno smanjiti robot te povećati prepreke. Robot bi se smanjilo na način da se za njegovu lokaciju u DRDS metodi odabere centar mase oblika robota. Ukoliko bi prepreke ostale jednakih dimenzija kao u stvarnosti dolazilo bi do kolizije između robota i prepreka zbog toga što DRDS metoda osigurava traženje najkraćeg puta, a najkraći put je u većini slučajeva određen gibanjem uz prepreke. Na taj način bi centar mase robota dolazio izuzetno blizu prepreka te bi kolizija bila neizbježna. Iz tog razloga je potrebno virtualno povećati prepreke. Korištenjem ovog predloška sve površine na slici povećavaju svoju površinu za jedan piksel u svim smjerovima. Ovaj postupak se ponavlja ukupno 18 puta kako robot sigurno zaobilazio sve prepreke. Matrica **A** u ovom predlošku ima oblik: $\mathbf{A} = \begin{bmatrix} 0,5 & 0,5 & 0,5 \\ 0,5 & 0.5 & 0,5 \\ 0,5 & 0,5 & 0,5 \end{bmatrix}$, matrica **B** je nul matrica dimenzija 3x3, a prag *z* je jednak nula.



Slika 30: Povećanje površine prepreka

7.2.3. Traženje težišta i orijentacije izoliranog objekta

Nakon što je robot izoliran na umanjenoj slici potrebno je pronaći njegovu točnu lokaciju. Za lokaciju mobilnog robota se uzima centar mase robota. Centar mase na pronalazi korištenjem MATLAB-ove naredbe regionprops, sa opcijom 'centroid'. Ova naredba u kodu izgleda: regionprops(y, 'centroid'), gdje je y slika na kojoj se nalazi izolirani robot. Izlaz ove naredbe daje x i y lokaciju centra mase na slici. Nakon što je izoliran robot na umanjenoj slici, radi se ispravak lokacije zbog smanjene točnosti na umanjenoj slici. Ispravljanje lokacije se izvršava da se sa početne slike izolira mali dio slike gdje je lociran robot, zatim se taj dio slike ispravi od distorzija te se na kraju utvrdi točna lokacija robota. Na identičan način se kontrolira promjena lokacije robota pri gibanju kroz labirint. Pri gibanju kroz labirint dodatno je potrebno poznavati orijentaciju robota zbog regulatora pomoću kojeg se zadaju rotacijska i translacijska brzina. Orijentacija robota se također izvod sa naredbom regionprops, ali se koristi opcija 'Orientatiton'. Tako naredba za dobivanje orijentacije ima oblik: regionprops(y, 'Orientatiton'). Izlaz ove naredba je orijentacija iskazana u stupnjevima. Ova naredba određuje orijentaciju odnosnom kraćih i duljih stranica mobilnog robota. To je u nekim situacijama predstavljalo problem u ovom radu, pa je zbog toga na robota dodana jedna značajka, u vidu komadića trake na robota, pomoću kojeg se ispravljala orijentacija.

Kako bi bilo moguće koristiti lokaciju i orijentaciju robota potrebno je te veličine prilagoditi analitičkom neizrazitom regulatoru. U MATLAB-u zapis slike je matrični te koordinate x osi rastu prema desno, ali koordinate y osi rastu prema dolje. To je u suprotnosti sa kartezijskim koordinatnim sustavom te je potrebno promijeniti orijentaciju y osi. Dodatno je potrebno napraviti kalibraciju slike, kako bi se znalo koliku duljinu svaki piksel predstavlja u realnom svijetu. Također, potrebno je promijeniti orijentaciju izraženu u stupnjevima u radijane, jer analitički neizraziti regulator zahtjeva orijentaciju u radijanima. Nakon izvršavanja svih prilagodbi moguće je upravljanje mobilnim robotom.

7.3. Cjelokupni algoritam za traženje robota i prepreka

Kako bi algoritam gibanja robota kroz labirint zaista radio potrebno je sve do sad opisane radnje uskladiti te ih pravilno izvršavati. Redoslijed izvođenja je:

I. učitavanje slike sa kamere,

II. ispravljanje učitane slike,

III. smanjivanje slike,

IV. povećanje kontrasta slike,

V. prebacivanje slike u gray-scale,

VI. uklanjanje tankih linja sa slike,

VII. uklanjanje malih površina sa slike,

VIII. ispunjavanje konkavnih površina na slici,

IX. lociranje robota na maloj slici

X. ispravljanje lokacije robota na velikoj slici,

XI. uklanjanje robota sa male slike kako bi ostale samo prepreke,

XII. povećanje površine prepreka,

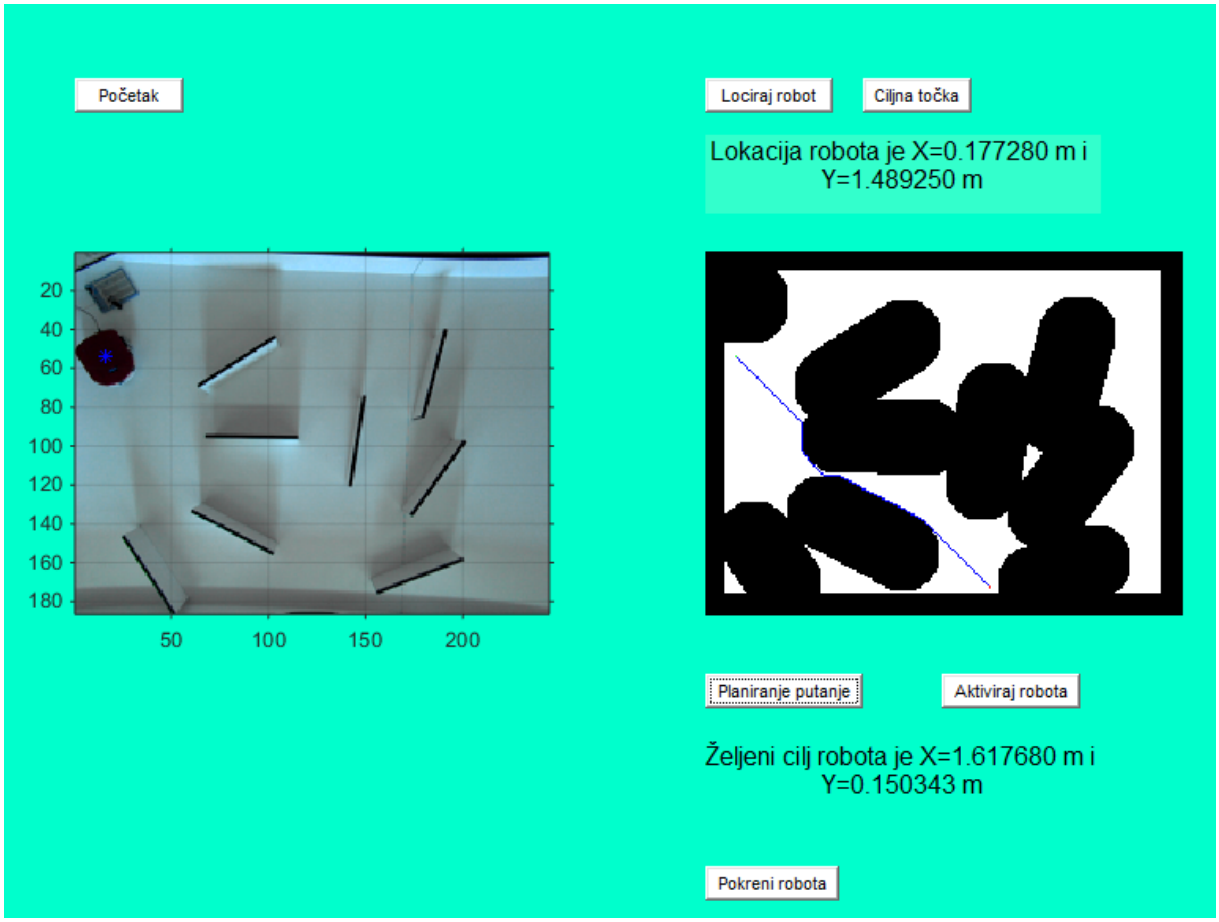
XIII. odabir ciljne točke robota,

XIV. planiranje putanje robota,

XV. uparivanje računala i robota preko bluetooth-a te

XVI. gibanje robota kroz labirint.

Cjelokupni algoritam je integriran u grafičko sučelje izrađeno pomoću MATLAB GUI-a. Sučelje je napravljeno kako bi korisniku osiguralo jednostavno i lako snalaženje bez potrebe za posebnim znanjima o procesima koji se izvršavaju u pozadini.



Slika 31: Izgled grafičkog sučelja

8. Izrada rada i rezultati

Rad je napravljen na Katedri za strojarsku automatiku, korištenjem eMIR mobilnog robota. eMIR mobilni robot su razvijeni na Katedri za strojarsku automatiku za edukaciju studenata. Komunikacija između računala i mobilnog robota je ostvarena preko Bluetooth-a. Upravljanje kretnjama eMIR se izvršava zadavanjem translacijske i rotacijske brzine. Maksimalna translacijska brzina koju eMIR roboti postižu je 0,5 m/s, a maksimalna rotacijska brzina je 240°/s. U ovom radu su korištene brzine u iznosu od 10 % maksimalnih brzina. eMIR robot je veličine 300x250 mm te svojom veličinom zauzima veliki dio prostora u kojem se nalazi, a koji je dimenzija 200x250 cm.

Slika 32: eMIR mobilni roboti

U ovom radu su vrlo bitne i značajke ovih robota, a to su:

- robot će se zaustaviti ukoliko ne primi nikakvu naredbu više od jedne sekunde
- robot će se isključiti ukoliko ne primi naredbu više od 60 sekundi

- robot će se isključiti ukoliko napon baterije padne ispod 10,5 V.

Kao što je već navedeno u ovom radu se gibanje mobilnog robot kontrolira kamerom instaliranom iznad prostora u kojemu se robot nalazi. Kamera proizvod tvrtke Basler te je rezolucije 1234x1624 piksela. Zbog velike količine podataka koje kamera šalje računalo kameru je moguće koristiti jedino sa USB3.0 portovima.

8.1. Izrada rada

Kako bi se napravljeni program izvršavao potrebno je postaviti robota u labirint, postaviti prepreke po labirintu koje definiraju prostor te je potrebno pokrenuti napravljeno grafičko sučelje. Prvi korak nakon otvaranja grafičkog sučelja je pritisak na tipku Početak. Ovom naredbom se uspostavlja komunikacija između kamere i MATLAB-a, te se ujedno uzima trenutačna slika labirinta. Drugi korak je traženje lokacije robota te prilagođavanje labirinta DRDS metodi planiranja putanje. Trajanje izvršavanja drugog koraka je otprilike 7 sekundi. Potrebno je naglasiti kako nakon završetka drugog koraka više nije dozvoljeno izvršavati nikakve izmjene u prostoru u kojem se robot nalazi. Ukoliko se naprave bilo kakve izmjene potrebno je ponoviti drugi korak. Treći korak je odabir željene ciljne točke. Četvrti korak je planiranje putanje DRDS metodom. Trajanje planiranja putanje uvelike ovisi o udaljenosti između početne i ciljne točke. Planiranje putanje se izvršava na slici veličine 15 % početne slike od 1234x1624 piksela. Iako je početna veličina slike umanjena, duljina trajanja planiranja putanje nije idealno kratka. Izvođenjem eksperimenata utvrđeno je da to vrijeme nikada ne prelazi jednu minutu. Peti korak je uspostavljanje komunikacije između robota i računala. Šesti korak ujedno i zadnji je gibanje robota kroz labirint. Iako je u 4 poglavlju predložen način generiranja trajektorije u eksperimentu u laboratoriju za trajektoriju su korištene točke generirane DRDS metodom. Razlog je činjenica da je labirint malih dimenzija u odnosu na sliku kojom se prezentiran te točke dobivene DRDS metodom

dovoljno gusto opisuju željeno gibanje. Drugi razlog je činjenica da je bilo potrebno napraviti programske prilagodbe labirinta kako bi se izbjegle kolizije robota sa preprekama te je na taj način izbjegnuta opasnost kolizije. Nakon završetka gibanja robota program je moguće ponovno pokrenuti sa novom lokacijom te sa mogućim novim rasporedom prepreka.

8.2. Dobiveni rezultati

Glavni problem u ovom radu je predstavljala komunikacija kamera sa MATLAB-om. Vrijeme dobivanja slike na obradu od slanja zahtjeva traje oko 1,2 sekunde, što se očituje konstantnim zastajkivanjem robota pri gibanju, zbog stalne potrebe za kontrolom pozicije i orijentacije robota. Iako traženje nove pozicije i orijentacije na novoj slici traje manje od 0,2 sekundi zbog velikog trajanja dohvata slike nije moguće izbjeći konstantno zastajkivanje robota. Zbog navedenog problema potrebno je ograničiti brzine gibanja robota. Također, zbog značajke da se robot giba jednu sekunda od kad je dobio naredbu bez da se može na njega utjecati zbog kašnjenja slike za obradu, potrebno je uzimati svaku drugu točku generiranu DRDS metodom. Razlog je činjenica da robot može vrlo lako proći slijedeću zadanu točku, pa bi regulator pokušavao vratiti robota nazad do te točke. Time bi se dodatno narušilo gibanje robota. Provedeni su i eksperimenti sa uzimanjem svake treće točke. Međutim, u određenim situacijama robot ne uspijeva stići generiranu trajektoriju te dolazi do kolizija sa preprekama. Kako bi se to izbjeglo preporuka je izvršavanje gibanja sa uzimanjem svake druge točke generirane DRDS metodom.

9. Zaključak

Od samih početaka razvoja mobilnih robota glavni cilj je njihova autonomnost. Kako bi se to osiguralo, potrebno je razviti kvalitetne algoritme vođenja robota. Jedan od najvećih problema je kako robotu osigurati gibanje prostorom, a da pri tom ne ošteti sebe, ali i svoju okolinu, uključujući i ljude koji bi se tamo mogli zateći. Da bi se izbjegle kolizije robota sa svojom okolinom, prvo je potrebno tu okolinu istražiti i na odgovarajući način opisati. U svrhu istraživanja i zapisivanja okoline razvijeni su mnogi algoritmi, primjerice metoda potencijalnih polja. Nakon što je prostor istražen, odnosno percipiran, moguće je planirati kretanje robota u toj percipiranoj okolini. Jedan način planiranja gibanja je dan u ovom radu, a za tu je svrhu primijenjena ovim radom modificirana metoda *podražajnog* DRDS-a. Prilikom predstavljanja rješenja traženja referentne trajektorije u radu je korištena slika tlocrta prostora. Na jednostavan način bi se moglo i rješenje mapiranog prostora pomoću metode potencijalnog polja prikazati slikom tlocrta. U ovom radu je prvo prikazan preuzeti način izrade *podražajnog* DRDS-a te su komentirane prednosti i mane samog sustava. Kasnije je predložen način unaprjeđenja samog sustava te su na kraju prikazani rezultati modificiranog algoritma u gibanju robota kroz više različitih prostora. Dobiveni rezultati su više nego optimistični i u potpunosti dokazuju uspješnost izvedenog algoritma. Ne samo da je pronađen put iz jedne u drugu točku nego je taj put u većini slučajeva najkraći mogući. Razlog što generirana putanja u pojedinim slučajevima nije ujedno najkraća treba tražiti u činjenici da su neki prostori izuzetno kompliciran. U takvim slučajevima, dolazi do konstantnog loma valova te zbog toga ponekad nije moguće dobiti najkraći mogući put. Unatoč tome i u takvim specijalnim slučajevima predložena rješenja zadovoljavaju sve sigurnosne zahtjeve pri kretanju robota željenim prostorom.

U radu je predložena metoda prikazana za nekoliko različitih okolina. Za sve njih metoda je dala izuzetno dobra i kvalitetna rješenja. Stoga ona predstavlja zanimljiv doprinos među ostalim metodama planiranja putanje robota zadanim prostorom.

Jedini uočeni nedostatak predložene metode je ponekad veća duljina trajanja obrade i planiranja putanje. Povećanjem opisanog prostora povećava se i utrošeno procesorsko vrijeme obrade, odnosno traženje putanje mobilnog robota. Kao što je i vidljivo u samim rezultatima rada vrijeme trajanja procesa dostiže i vrijednosti do jedne minute. Naravno, korištenjem moćnijih računala ovo vrijeme se može uvelike smanjiti. Međutim, kvalitetno rješenje ovog problema je korištenje FPGA tehnologije koja pruža mogućnost paralelnog izvršavanja zadane dinamike sustava. Na taj način bi bilo moguće postići pronalaženje referentne trajektorije u realnom vremenu te bi se takvi sustavi mogli koristiti u industrijskoj primjeni gdje se na zahtijevane promjene položaja zahtjeva i što brži odziv.

Nakon što je prikazan način planiranja puta robota kroz željene prostore, prikazan je i način generiranja referentne trajektorije na temelju planiranog puta. Pri samom generiranju trajektorije korištena su znanja o prostoru u kojem se robot nalazi. Generiranom trajektorijom dokazano je da se planirano gibanje robota može zaista odraditi na posve siguran način i za robot i za okolinu. Potom je prikazana simulacija praćenja generirane trajektorije mobilnim robotom upravljanim analitičkim neizrazitim regulatorom. Rezultati simulacija gibanja mobilnog robota su pokazali sve prednosti ove metode jer je gibanje izvršeno na siguran način te je i sama putanja gibanja robota gotovo najkraća moguća.

Nakon dokazivanja uspješnosti predložene metode u simulaciji izrađen je algoritam gibanja mobilnog robota u realnim uvjetima. Eksperiment je proveden na način da se sa slike dobivene kamerom, lociranom iznad labirinta, locira robot i prepreke te se izvršava gibanje korištenjem DRDS metode. Iako su se javili problemi prvenstveno u komunikaciji između

kamere i MATLAB-a, čime je dodatno usporeno gibanje robota, ovim radom je dokazana i praktična primjena predložene metode. Navedeni problemi bi se mogli riješiti promjenom programskog jezika. Trajanje dobivanja slike sa kamere bi se na taj način uvelike smanjilo te bi se postiglo jednoliko i kontinuirano gibanje.

Literatura

[1] R. Siegwart, I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, Massachusetts Institute of Technology, 2004.

[2] U. Nehmzow, Mobile Robotics: A Practical Introduction Second Edition, Springer-Verlag London, 2003.

[3] J. D. Murray, Mathematical Biology II: Spatial Models and Biomedical Applications, Third Edition, Springer, New York, 2003.

[4] A. M. Turing, The Chemical Basis of Morphogenesis, Philosophical Transactions oft he Royal Society of London. Series B, Biological Sciences, Vol. 237, pp. 37-72, Aug. 1952.

[5] A. S. Sherstinsky, R. W. Picard, M-Lattice: From Momhorzenesis to Image Processing, IEEE Transactions on image processing, Vol. 5, No. 7, pp. 1137-1150, July 1996.

[6] O. Steinbock, A.Toth, K. Showalter, Navigating Complex Labyrinths: Optimal Paths from Chemical Waves, Science, Vol. 267, pp. 868-871, Feb. 1995.

[7] K. Ito, M. Hiratsuka, T. Aoki, T. Higuchi, A Shortest Path Search Alorithm Uging an Excitable Digital Reaction-Diffusion, IEICE Trans. Fundamentals, Vol. E89-A, No. 3, pp. 735-743, March 2006.

[8] K. Ito, T. Aoki, T. Higuchi, Digital reaction-diffusion systems and it's application to bio-inspired texture image processing, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer SCIENCES, Vol. E84-A, No. 8, pp. 1909-1918, 2001.

[9] K. Ito, T. Aoki, T. Higuchi, Design of an Excitable Digital Reaction-Diffusion Systems for Shortest Path Search, The 2004 Interantional Technical Conference on

Circuits/Systems, Computers and Communications (ITC-CSCC2004), Japan, pp. 7F3P-1-1-7F3P-1-4, July 2004.

[10] K. Ito, T. Aoki, T. Higuchi, Digital reaction-diffusion systems- A fundation of Bio-inspired texture image processing, IEICE Trans. Fundamentals, Vol. E84-A, No.8, pp. 1909-1918, August 2001.

[11] J. D. Murray, Mathematical Biology: I. An Introduction, Third Edition, Springer, 2007.

[12] A. Gacsadi, T. Maghiar, A CNN path planning for a mobile robot in an environment with obstacles, Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). Proceedings oft he 2002 7th IEEE International Workshop on, pp. 188-194, July 2002.

[13] I. Gavrilut, V. Tiponut, A. Gacsadi, Mobile Robot Navigation based on CNN Images Processing – An Experimental Setup, WSEAS Transactions on Systems, Issue 8, Vol. 8, pp. 947-956, August 2009.

[14] I. Gavrilut, V. Tiponut, A. Gacsadi, Path Planning of Mobile Robots by Using Cellular Neural Networks, 10th International Workshop on Cellular Neural Networks and Their Applications, Istanbul, Turkey, pp. 1-6, August 2006.

[15] I. Gavrilut, A. Gacsadi, C. Grava, V. Tiponut, Vision based algorithm for path planning of a mobile robot by using cellular neural networks, Automation, Quality and Testing, Robotics, 2006 IEEE Interantional Conference on, pp. 306-311, May 2006.

[16] G. Chen, T. T. Pham, Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, CRC Press, 2000.

[17] J. Kasać, Analiza stabilnosti nelinearnih sustava vođenih analitičkim neizrazitim regulatorom, doktorski rad, Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu, 2005.

[18] B. Novakovic, D. Scap, D. Novakovic, An analytic approach to fuzzy robot control synthesis, Engineering Applications of Artificial Intelligence 13, pp. 71–83, 2000.

[19] R. C. Gonzalez, Digital Image Processing, 3 edition, Prentice Hall, August 31, 2007.

[20] R. G. Kuehni, Color Space and Its Divisions: Clolor Order from Antiquity to the Present, Wiley-Interscience, March 31, 2003.

[21] http://www.mathworks.com/matlabcentral/fileexchange/37980-barrel-and-pincushion-lens-distortion-correction/content/lensdistort/lensdistort.m, posječeno dana 26.06.2015.

[22] Leon O. Chua, Lin Yang, Cellular Neural Networks: Theory, IEEE Transactions on circuit and systems, Vol. 35, No 10., 1257–1272, October 1988.

[23] Leaon O. Chua, Tamas Roska, Cellular neural network and visual computing: Foundations and applications, Cambridge University Press, 2002.

[24] Software Library for Cellular Wave Computing Engines, Cellular Sensory and Wave Computing Laboratory of the Computer and Automation Research Inst., Hungarian Academy of Sciences and the Jedlik Laboratories of the Pázmány P. Catholic University, Version 3.1, Budapest, 2010.