

Automatsko punjenje eMiR mobilnog robota

Lukas, Marin

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://um.nsk.hr/um:nbn:hr:235:618137>

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

Download date / Datum preuzimanja: **2025-01-04**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Marin Lukas

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Mladen Crneković, dipl. ing.

Student:

Marin Lukas

Zagreb, 2015.

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

Zahvaljujem se mentoru dr.sc. Mladenu Crnekoviću, red. prof., svojoj obitelji te svim nastavnicima Fakulteta strojarstva i brodogradnje u Zagrebu koji su mi pomogli tijekom studija te time na neki način doprinijeli i ovom radu.

Marin Lukas



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
procesno-energetski, konstrukcijski, brodostrojarški i inženjersko modeliranje i računalne simulacije

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

DIPLOMSKI ZADATAK

Student:

Mat. br.:

Naslov rada na
hrvatskom jeziku:

Naslov rada na
engleskom jeziku:

Opis zadatka:

Zadatak zadan:

Rok predaje rada:

Predviđeni datumi obrane:

Zadatak zadao:

Predsjednik Povjerenstva:

Prof. dr. sc. Zvonimir Guzović

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	IV
POPIS TEHNIČKE DOKUMENTACIJE	V
POPIS OZNAKA	VI
SAŽETAK.....	VII
SUMMARY	VIII
1. UVOD.....	1
2. Mobilni robot eMiR.....	3
2.1. Nadogradnja robota eMiR.....	4
2.2. Računalo Raspberry Pi.....	4
2.3. Sustav komunikacije putem LAN-a.....	5
2.3.1. WLAN komunikacija putem VNC protokola	6
2.3.2. RBF protokol	7
2.3.3. Sigurnosni aspekti VNC protokola	8
2.3.4. VNC protokol na Raspberry Pi računalu	8
3. Konstrukcija sustava za automatsko punjenje robota eMiR.....	10
3.1. Temeljni princip rada sustava za automatsko punjenje	10
3.2. Predajnik signala za navođenje	10
3.2.1. Elektronički sklop predajnika	11
3.2.2. Mehanička konstrukcija predajnika	13
3.2.3. Generiranje signala za navođenje	14
3.3. Prijemnik signala za navođenje.....	15
3.3.1. Detektor infracrvenog signala	15
3.3.2. Elektronički sklop prijemnika signala za navođenje.....	16
3.3.3. Mehanička konstrukcija prijemnika.....	17
3.4. Ispitivanje dometa sustava prijemnik – predajnik.....	18
3.5. Uređaj za spajanje robota na punjač	20
3.6. Montaža sklopa za automatsko punjenje na robota eMiR	22
3.7. Montaža stupa punjača na poligon.....	23
4. Vođenje robota do punjača	24
4.1. Strategija vođenja robota do punjača	24
4.2. Mjerenje kuta punjača u odnosu na robota	25
4.3. Regulacija gibanja robota eMiR	26
4.3.1. Pravila ponašanja	28
5. Programska podrška za vođenje robota prema punjaču	29
5.1. Programski jezik Python	29
5.2. Skripta za upravljanje robotom	29
5.2.1. Funkcija eMiR_citaj()	30
5.2.2. Funkcija eMiR_dock()	30

5.2.3. Funkcije BattStatus() i Chgcon()	32
5.3. Program za ispitivanje rada sustava	33
6. Podešavanje i ispitivanje sustava za automatsko punjenje	35
6.1. Podešavanje konstanti regulatora gibanja	35
6.2. Analiza spajanja robota na stup punjača	38
6.3. Ispitivanje rada sustava za automatsko punjenje	39
7. Zaključak	41
8. Literatura.....	43
9. Prilog – programski kod	45

POPIS SLIKA

Slika 1 Mobilni roboti za kućanstvo na svjetskom tržištu [1].....	1
Slika 2 Mobilni robot eMiR	3
Slika 5 Shematski prikaz WLAN komunikacije	6
Slika 6 Prikaz zaslona Raspberry Pi računala putem VNC protokola	9
Slika 7 Faze automatskog spajanja robota na punjač	10
Slika 8 Karakteristike IR svjetleće diode	11
Slika 9 Elektronička shema predajnika signala za navođenje.....	12
Slika 10 Mehanička konstrukcija predajnika	13
Slika 11 Programski kod i snimka signala za navođenje	15
Slika 12 Shema detektora infracrvenog signala	15
Slika 13 Karakteristike detektora infracrvenog svjetla	16
Slika 14 Elektronička shema prijemnika.....	17
Slika 15 Mehanička konstrukcija prijemnika.....	17
Slika 16 Ovisnost dometa o struje IR diode	18
Slika 17 Dopuštena odstupanja od nominalne frekvencije	19
Slika 18 Uređaj za spajanje robota na punjač	20
Slika 19 Princip djelovanja lisnate opruge [17]	20
Slika 20 Skica ugradnje elektrode na nosač	21
Slika 21 Robot eMiR s montiranim udeđajem za spajanje na punjač.....	22
Slika 22 Dijelovi sustava za automatsko punjenje na robotu eMiR.....	23
Slika 23. Stup punjača na poligonu	23
Slika 24 Robot i punjač u horizontalnoj ravnini	24
Slika 25 Područje mjerenja prijemnika	25
Slika 26 Klasični regulacijski krug	26
Slika 27 Diskontinuirani regulacijski krug.....	27
Slika 28 Dijagram toka funkcije citaj()	30
Slika 29 Dijagram toka emir_dock.....	31
Slika 30 Napon u ovisnosti o stupnju napunjenosti [20].....	32
Slika 31 Dijagram toka programa za ispitivanje	34
Slika 32 Robot na poligonu za vrijeme ispitivanja	35
Slika 33 Ponašanje robota za $w = 5$	36
Slika 35 Ponašanje robota za $w = 10$	37
Slika 34 Ponašanje robota za $w = 15$	37
Slika 36 Ponašanje robota za $w = 20$	38
Slika 37 Robot u trenutku spajanja na punjač	39
Slika 38 Ispitivanje spajanja na punjač uz prepreke	40

POPIS TABLICA

Tablica 1. Naredbe za upravljanje robotom eMiR **Error! Bookmark not defined.**

POPIS TEHNIČKE DOKUMENTACIJE

BROJ CRTEŽA Naziv iz sastavnice

POPIS OZNAKA

Oznaka	Jedinica	Opis
Oznaka 1	Jedinica	Opis oznake
Oznaka 2	Jedinica	Opis oznake
Oznaka 3	Jedinica	Opis oznake
Oznaka 4	Jedinica	Opis oznake

SAŽETAK

U ovom radu detaljno je opisano rješavanje problema automatskog punjenja mobilnog robota. Kroz rad zložena je metoda vođenja robota prema punjaču korištenjem signala za navođenje, projektirani su i izrađeni prijemnik i predajnik za navođenje te izrađena potrebna programska podrška. Opisani sustav za automatsko punjenje ugrađen je na robota eMiR te je funkcionalnost sustava ispitana na poligonu u laboratoriju za inteligentne proizvodne sustave. Sva programska podrška napisana je korištenjem programskog jezika Python. Dijelovi potrebni za realizaciju projekta izrađeni su aditivnom tehnologijom FDM.

Ključne riječi: Mobilni robot, robotika, automatsko punjenje, vođenje robota, regulacija gibanja.

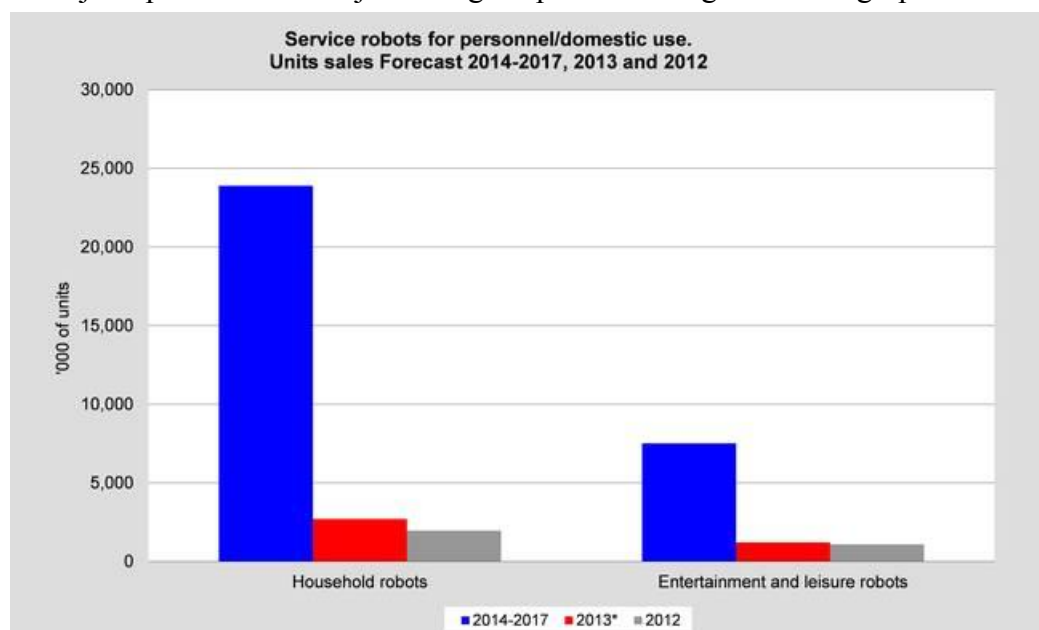
SUMMARY

In this paper system for automatic charging of the mobile robot is described. The described system for automatic charging is installed on the robot eMiR. For robot navigation infrared guidance signal is used. System for automatic charging consists of guidance signal source, signal receiver and docking apparatus. For system testing purposes, robot control software is written. Functionality of the system is tested at laboratory for intelligent manufacturing systems of the Faculty of mechanical engineering and naval architecture. All software is written using the Python programming language. All parts needed for system testing and implementation are constructed using additive technology FDM.

Key words: Mobile robot, robot guidance, automatic charging, autonomous robot control

1. UVOD

Prema podacima međunarodnog udruženja za robotiku IFR (International Federation of Robotics) [1] potrebe tržišta za mobilnim robotima sve su veće. Posebno veliki rast zabilježen je na području robota za kućanstvo koji se dijele na robote za pranje i čišćenje, robote za pomoć osobama s invaliditetom te robote za zabavu. Statistički podaci istog udruženja govore nam da je u 2013. godini na svjetskom tržištu prodano četiri milijuna robota za kućanstvo, što je oko 20% više nego u 2012. godini. Prema projekcijama do 2017. godine na svjetskom tržištu trebalo bi se prodati oko dvadeset četiri milijuna robota. Usporedbe radi, prodano je 180 000 industrijskih robota na svjetskom tržištu prema podacima IFR-a u 2013. godini iz čega možemo zaključiti da je tržište robota za kućanstvo znatno veće. Roboti u kućanstvu bitno su različiti od industrijskih robota kako po mehaničkoj konstrukciji tako i po načinu vođenja. Roboti u industriji obavljaju strogo propisane zadatke u određenom vremenu na točno određenim mjestima, dok se roboti u kućanstvu nalaze u puno manje uređenoj okolini. Isto tako načini vođenja industrijskih robota nisu primjenjivi u kućanstvima zbog razlike u znanju programiranja te zadacima koje je potrebno obaviti. Da bi se mobilni roboti uspješno integrirali kako u kućanstvo tako i industrijsku okolinu, potreban je čim veći stupanj autonomnosti u njihovom djelovanju. Osim autonomije na području vođenja i prepoznavanja zadataka, jedan od vrlo važnih aspekata je i energetska autonomnost mobilnog robota. Vrijeme rada robota određeno je kapacitetom baterija te snagom potrošača. Ugrađena snaga potrošača uglavnom

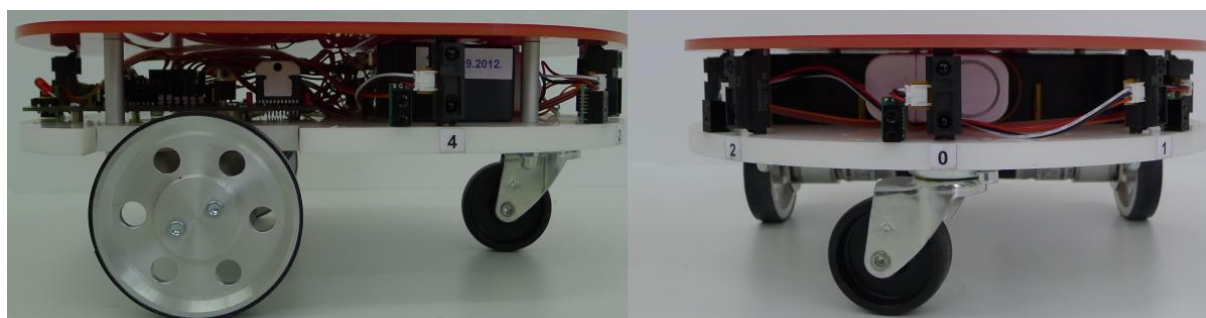


Slika 1 Mobilni roboti za kućanstvo na svjetskom tržištu [1]

proizlazi iz zadataka koje robot treba izvršavati kao što su podizanje tereta, protok i tlak zraka usisavača prašine, dok je maksimalni kapacitet baterije određen mehaničkom konstrukcijom robota. Jedino dostupno rješenje za praktički beskonačnu energetska autonomnost mobilnog robota jest automatsko punjenje baterija. Jedno od najpoznatijih komercijalnih rješenja na tržište je plasirala tvrtka iRobot u svojim modelima robotskih usisavača za kućanstvo Roomba 774, no ono je potpuno prilagođeno proizvodnju tvrtke i u druge se svrhe nemože koristiti.

Postoje razni pristupi rješavanju problema automatskog punjenja baterija koji se uglavnom svode na automatsko vođenje robota do stanice za punjenje (eng. docking station). U radovima [2] Armashinge et al., [3] Jeung Min et al. [4] Cooperstock et al. koriste kamere i računalne vizijske sustave te neuronske mreže, što je moguće samo uz uporabu relativno velikih robotskih platformi opremljenih jakim računalima, dok je implementacija na malim robotima gotovo nemoguća. Također je zanimljiv pristup iz [5] Cassinis et al. inspiriran sustavima sektorskih svjetala za uplovljavanje u luke poznatim iz pomorske tehnike, no iz istih razloga teško je primjenjiv na mobilnim robotima. Druga grupa autora problem vođenja rješava korištenjem infracrvenih signala za navođenje pa tako Kim et al. u radu [6] opisuju sustav za navođenje temeljen na jeftinim infracrvenim prijemnicima uobičajenim u TV uređajima za domaćinstvo. U radu [7] Varon et al. izlažu hibridni pristup u kojem se robota navodi prema stanici za punjenje korištenjem infracrvenog signala za navođenje i ultrazvučnog daljinomjera pomoću kojega robot metodama trijangulacije određuje svoj položaj u odnosu na stanicu za punjenje. U ovom radu opisan je sustav za automatsko punjenje mobilnog robota eMiR detaljno opisanog u radovima [8], [9]. Sustav za automatsko punjenje logični je nastavak nadogradnje na robotsku platformu eMiR na kojoj je izvršena nadogradnja opisana u radu [10] te ispitivanje autonomnog traženja predmeta korištenjem vizijskog sustava [11].

2. Mobilni robot eMiR



Slika 2 Mobilni robot eMiR

Robot eMiR konstruiran je i izrađen na Katedri za strojarSKU automatiku Fakulteta strojarstva i brodogradnje u Zagrebu. Robot je diferencijalne strukture što omogućava okretanje oko vertikalne osi u mjestu te pojednostavnjuje konstrukciju. Dva stražnja kotača pogone se neovisnim istosmjernim motorima s planetarnim reduktorima. Enkoderi su inkrementalnog tipa i smješteni su na vratilo motora. Motori i kotači postavljeni su na ploču od PMMA (poli metil metakrilat) te pričvršćeni vijcima. Kotači robota izrađeni su od aluminijske legure te obloženi elastomernom oblogom kako bi se povećalo trenje između kotača i podloge te time smanjilo proklizavanje. Iznad temeljne ploče nalazi se pokrovna ploča koja pridržava senzore za mjerenje udaljenosti raspoređene na vanjskoj strani robota i štiti upravljačku elektroniku od vanjskih utjecaja. Na pokrovnu ploču moguće je smjestiti dodatnu opremu. Upravljačka jedinica robota eMiR također je konstruirana i izrađena na Katedri za strojarSKU automatiku Fakulteta strojarstva i brodogradnje u Zagrebu. Glavni dio upravljačke jedinice je mikrokontroler iz porodice 8051 [8] u kojem se nalazi program za regulaciju brzina translacije i rotacije. Motorima se upravlja pomoću širinsko impulsne modulacije (PWM), a snaga se motorima dovodi preko integriranih krugova sa ugrađenim H-mostovima. Enkoderi motora spojeni su direktno na mikrokontroler. Komunikacija između glavnog računala i upravljačke jedinice odvija se putem serijske veze RS232. Kako bi se izbjeglo spajanje kablova na upravljačku jedinicu, komunikacija se uspostavlja putem Bluetooth modula instaliranog na pločicu. Za mjerenje udaljenosti između robota i ostalih objekata u okolini koriste se analogni infracrveni daljinomjeri Sharp. Analogni signal pretvara se u digitalni korištenjem analognog digitalnog pretvornika te se nakon toga u digitalnom obliku prosljeđuje na mikrokontroler. S obzirom da se unutar upravljačke jedinice nalazi PI regulator kutne i translacijske brzine, korisniku izvana nije potreban pristup širinsko impulsnoj modulaciji i signalima sa enkodera, već se robotu putem serijske veze šalju reference kutne brzine ω i translacijske brzine v_t . Komunikacija se odvija

brzinom od 57600BPS što je dovoljno s obzirom da je interval prijema podataka 0.1s. Svaka naredba poslana robotu započinje znakom „#“, a završava znakom „/“. Između navedenih znakova nalaze se informacije o modu rada, translacijskoj i kutnoj brzini te kontrolna vrijednost (Checksum). Sve broježane vrijednosti zadaju se u heksadekatskom sustavu. Osim kontrole gibanja robotu je moguće slati naredbe za uključivanje senzora. Kada robot primi naredbu za uključivanje senzora, upravljačka jedinica računala šalje paket podataka u kojem se nalaze vrijednosti daljinomjera, trenutna kutna i translacijska brzina te napon baterije. Sve broježane vrijednosti također su poslone u heksadekatskom brojevnom sustavu. Neke od naredbi za komunikaciju s robotom te pripadajuće radnje prikazane su u Tablici 1.

NN	Command	Description
0	# 00 00 00 00 /	Stop the robot and exit from a current task.
1	# 01 vv rr CS /	Move the robot with translation speed vv and rotation speed rr (-100 to +100%).
17	# 11 PP 00 CS /	Define returned information package PP (0, 1 or 2).
19	# 13 tt 00 CS /	Turn on internal horn for time tt tenths of seconds.
255	# FF 00 00 01 /	Turn off the robot.

Tablica 1 Naredbe za upravljanje robotom eMiR

2.1. Nadogradnja robota eMiR

Kako bi se povećala autonomija robota, jedan od modela nadograđen je računalom Raspberry Pi te je uveden komunikacijski protokol VNC. Pomoću protokola je moguće komunicirati s robotom na veće udaljenosti i moguća je komunikacija između pojedinih robota. U daljnjem tekstu ukratko je opisan sustav nadograđenog robota te način komunikacije putem VNC protokola.

2.2. Računalo Raspberry Pi

Raspberry Pi je računalo veličine kreditne kartice, razvijeno od 2006. do 2012. godine u Velikoj Britaniji. Računalo je razvila zaklada Raspberry Pi kao maleno i jeftino računalo namijenjeno učenju programiranja i računalne znanosti. Osnovna komponenta računala je mikrokontroler iz porodice ARM ARM1176JZF-S koji radi na 700MHz. Radna memorija za sada iznosi 512MB. U osnovnoj konfiguraciji, na računalo se spaja memorijska kartica sa inačicom operativnog sustava Lynux prilagođenom za ARM arhitekturu Raspbian Wheezy. Inačica je nastala od

popularne Lynux distribucije Debian za koju postoji relativno velika baza upravljačkih programa i ostale podrške za upravljanje periferijama. Za razliku od uobičajenih PC-a i Netbooka, Raspberry Pi uz standardne USB, LAN, HDMI i 3.5mm AUDIO ulaze/izlaze, posjeduje i GPIO (General Purpose Input/Output) modul kojemu je moguće direktno pristupiti. GPIO omogućuje jednostavno spajanje na upravljačke jedinice koračnih motora, servo motora te ostalih aktuatora i senzora. Preko GPIO modula također je moguće uspostaviti I2C RS232 1-Wire i SPI komunikacije. Pločica se napaja naponom 5V, a kod rada bez puno periferija potrebna je struja od 700mA. Sve navedeno čini Raspberry Pi vrlo pogodnim za ugradnju na edukacijske robote kao što je eMiR. Slika 8 prikazuje računalo Raspberry Pi model B



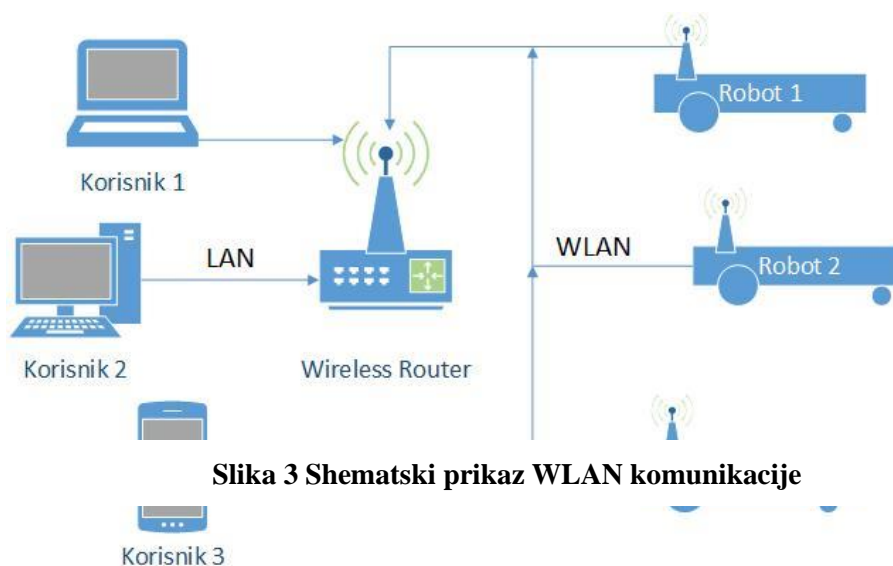
Slika 3 Računalo Raspberry Pi

2.3. Sustav komunikacije putem LAN-a

Komunikacija s robotom odvija se putem bežične lokalne mreže WLAN na frekvenciji 2.4GHz. Komunikacijski protokoli korišteni u LAN (Local Area Network) sustavima znatno su kompliciraniji od komunikacijskih protokola kao što su RS232, I2C, SPI, uobičajenih među kontrolerom i periferijama pa je zbog toga potrebno posvetiti posebnu pažnju ovom dijelu sustava. Da bismo mogli ostvariti komunikaciju putem WLAN-a, osim dva računala koja komuniciraju, potrebno je i treće ugrađeno računalo tzv. „router“ koje raspodjeljuje komunikacijski prostor među sudionicima komunikacije. U ovom obliku komunikacije,

računala se međusobno pozivaju putem IP adresa koje mogu biti statičke i dinamičke. Kod sustava sa statičkim IP adresama korisnici unaprijed zadaju adrese računalima i one ostaju iste za čitavo vrijeme rada, dok kod sustava s dinamičkim IP adresama proces teče nešto drugačije. Dinamičke IP adrese ne određuju korisnici na računalima već „Router“ nakon spajanja računala na mrežu. Dinamički sustav IP adresa izbjegava konflikte koji mogu nastati ukoliko se na mreži pojavi više računala s istom IP adresom zbog toga što adrese ne zadaju korisnici već računalo koje ima uvid u raspoloživ prostor adresa. Za potrebe ovog projekta korišten je sustav statičkih IP adresa te protokol VNC koji je opisan u daljnjem tekstu. Cilj korištenja WLAN-a je korisniku omogućiti lak pristup robotu sa osobnog računala te omogućiti komunikaciju između više robota. Slika 5 shematski prikazuje komunikaciju između više robota i korisnika putem WLAN-a.

2.3.1. WLAN komunikacija putem VNC protokola



Slika 3 Shematski prikaz WLAN komunikacije

VNC (eng. Virtual Network Computing) je sustav za udaljeno povezivanje koji koristi RFB (eng. remote framebuffer) protokol za upravljanje udaljenim računalom. On obavlja prijenos događaja s jednog računala (npr. ulaza s tipkovnice ili miša) na drugo omogućavajući prikaz promjena na zaslonu. Neovisan je o platformi, što znači da se VNC preglednik na jednom operacijskom sustavu može povezati s VNC poslužiteljem na istom ili bilo kojem drugom operacijskom sustavu. Također, više korisnika može se povezati s istim VNC poslužiteljem istovremeno. Popularna uporaba ove tehnologije uključuje udaljenu tehničku podršku i pristup datotekama na radnom računalu s kućnog računala

Jedan VNC sustav sastoji se od tri komponente :

1. **VNC poslužitelj** – program na računalu koji služi za pružanje kontrole klijentu.
2. **VNC klijent** – program koji služi za pregled, kontrolu i interakciju s poslužiteljem.
3. **VNC protokol** – RFB protokol, vrlo jednostavni protokol za prijenos slike ili poruka o događajima između poslužitelja i klijenta.

Komunikacija se odvija tako da se klijent poveže s poslužiteljem preko priključka (obično 5900). Alternativa je povezivanje preglednika s poslužiteljem kada se koristi priključak 5800, ili povezivanje poslužitelja s klijentom u „modu za slušanje“ na priključku 5500. Prednost tog načina rada jest da poslužiteljska strana ne mora imati konfiguriran vatrozid za omogućavanje pristupa na priključku 5900 ili 5800. Poslužitelj šalje male podatkovne blokove (eng. rectangles of the framebuffer) klijentu. U najjednostavnijem obliku, VNC protokol koristi puno resursa pa su razvijene metode kodiranja tj. efikasnijeg prijenosa podataka. VNC protokol omogućuje klijentu i poslužitelju dogovaranje oko metode kodiranja koja će se koristiti. Najjednostavnije kodiranje, koje podržavaju svi klijenti i poslužitelji, jest kodiranje reda (eng. raw encoding). Radi se o metodi kod koje se podaci o pikselu (najmanji grafički element slike) šalju u redosljed od lijeve strane prema desnoj, a nakon slanja cijele slike šalju se samo dijelovi koji se mijenjaju. Ovo kodiranje je korisno i efikasno samo ako se mali dijelovi slike mijenjaju od okvira do okvira. To je slučaj kod pomicanja miša na statičnoj pozadini ili upisa teksta u okvir. Međutim, kod velikih promjena slike, kao kod pregleda video zapisa, ova metoda postaje neuporabljiva. Uporaba VNC-a preko Interneta je moguća ako korisnik ima širokopojasnu vezu na strani klijenta i poslužitelja. Međutim, kako bi se ostvarila veza, ponekad je potrebno konfigurirati NAT (eng. Network Address Translation), vatrozid ili usmjeritelj. Neki korisnici mogu odabrati uporabu VPN (eng. Virtual Private Network) mreža za lakše korištenje VNC-a preko Interneta. Alternativno, VNC veza može se ostvariti kao LAN (eng. Local Area Network) veza.

2.3.2. **RFB protokol**

RFB je jednostavan protokol za udaljeni pristup računalu koji pruža grafičko korisničko sučelje. Budući da je implementiran na tzv. framebuffer razini (logički sloj za prikaz grafičkih promjena na uređajima), dostupan je za sve sustave (X11 za Linux, Windows i Macintosh inačice) te aplikacije. Proširen je s dodatnim značajkama (poput funkcije prijenosa podataka) i puno sofisticiranijim postupcima za kompresiju te sigurnosnim tehnikama. Kako bi se omogućila kompatibilnost više različitih inačica klijenata i poslužitelja, uveden je postupak dogovaranja parametara veze prije povezivanja. Pri tome klijent i poslužitelj dogovaraju inačicu RFB kodiranja, najprikladniju kompresiju i najbolje sigurnosne opcije koje podržavaju obje strane. Protokol je prvotno razvijen kao tehnologija za udaljene zaslone koju bi koristio klijent s ATM (eng. Asynchronous Transfer Mode) vezom. Sljedeću i najznačajniju ulogu, protokol RFB pronašao je u VNC sustavu kada je objavljen kao slobodan i besplatan za uporabu.

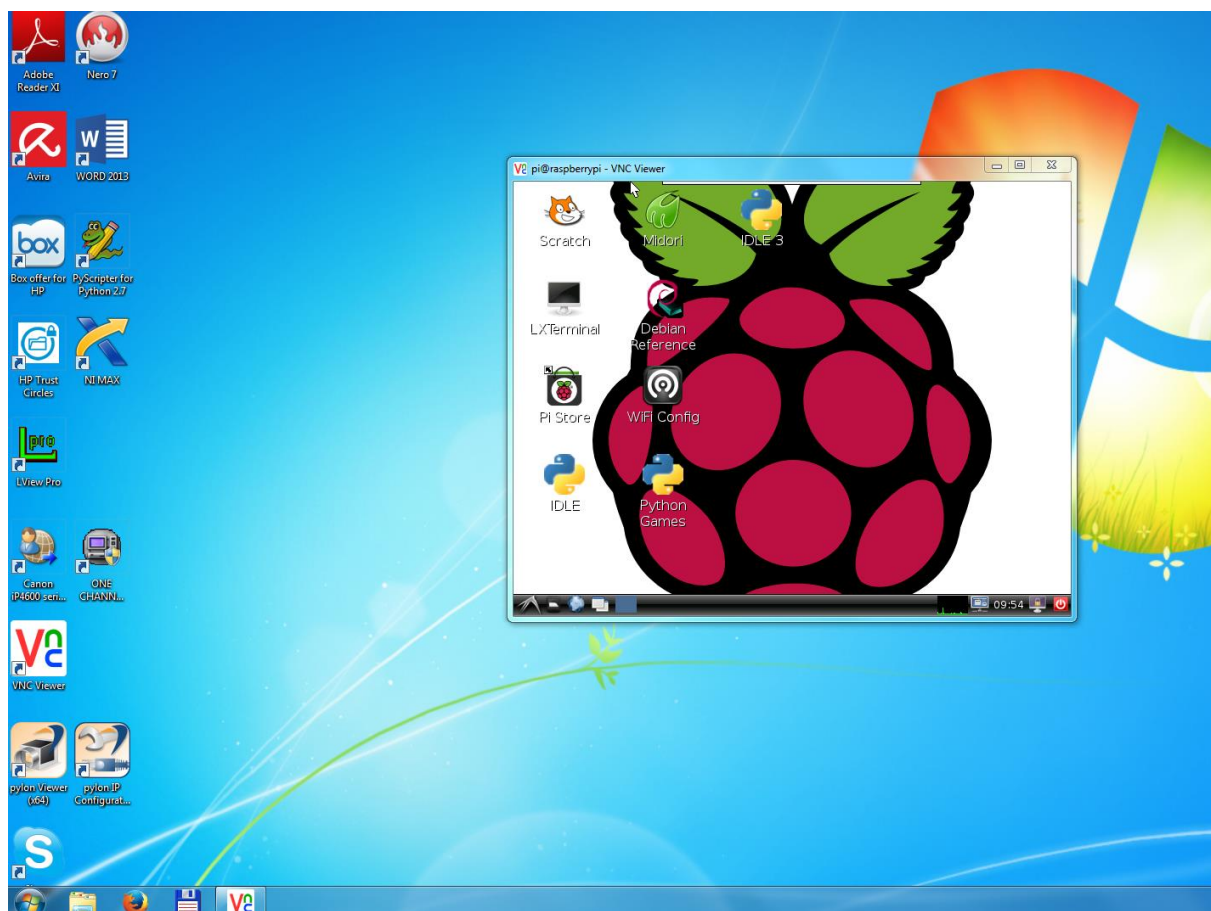
2.3.3. Sigurnosni aspekti VNC protokola

Protokol VNC je po svojoj prirodi nesiguran. Usmjeren je na učinkovit prijenos poruka između korisnika i poslužitelja. Postupci autentifikacije i šifriranje nisu uključeni u izvornu inačicu protokola. Ranije inačice su posebno ranjive na otkrivanje osjetljivih podataka ili njihovo dešifriranje. Kako bi se podigla razina sigurnosti moguće je uspostaviti komunikaciju preko protokola SSH (eng. SecureShell)

2.3.4. VNC protokol na Raspberry Pi računalu

Opisani komunikacijski protokol VNC moguće je jednostavno implementirati na računalu Raspberry Pi. Za potrebe komunikacije na Raspberry Pi računalu instalira se programski paket Tight VNC server. Paket je moguće instalirati korištenjem konzolnih Linux komandi a postupak je opisan u radu [10]. Nakon uspješnog instaliranja paketa na Raspberry Pi, potrebno je instalirati program RealVNC na svako računalo s kojega se želi upravljati robotom. Prilikom pokretanja RealVNC klijenta na korisničkom računalu, potrebno je upisati IP adresu na kojoj se nalazi Raspberry Pi. Nakon uspješnog spajanja, na korisničkom računalu je moguće vidjeti i

koristiti zaslon RaspberryPi računala. Slika 6 prikazuje izgled zaslona korisničkog računala prilikom programiranja na RaspberryPi računalu putem VNC protokola.

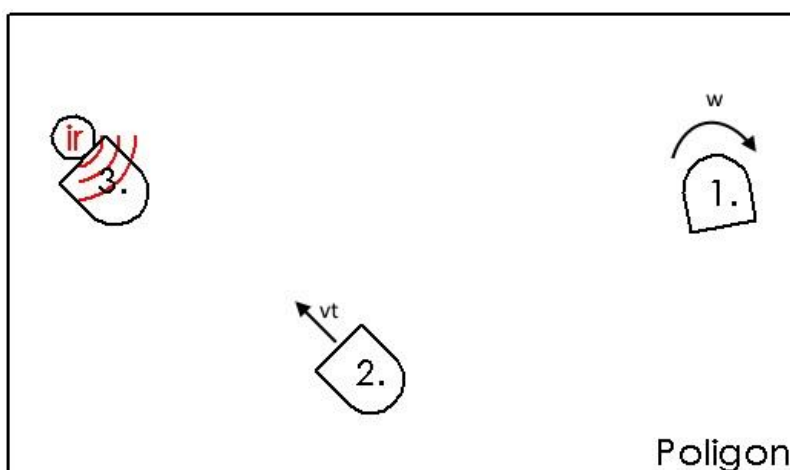


Slika 4 Prikaz zaslona Raspberry Pi računala putem VNC protokola

3. Konstrukcija sustava za automatsko punjenje robota eMiR

3.1. Temeljni princip rada sustava za automatsko punjenje

Kao što je opisano u prvom poglavlju, postoje razni pristupi rješavanju problema automatskog punjenja. U ovom radu odabran je pristup koji koristi infracrveni signal za navođenje te pripadajuće prijemnike na robotu. Prije samog spajanja potrebno je odrediti razinu ispražnjenosti baterije kako bi se automatsko spajanje na punjač počelo izvršavati pravovremeno. Zadatak automatskog spajanja robota na punjač moguće je podijeliti u tri dijela. Prvi dio je traženje punjača, tj. pronalaženje odgovarajućeg smjera kretanja prema stanici za punjenje. U drugom dijelu se robot autonomno kreće prema stanici određenom translacijskom brzinom kako bi ostvario kontakt sa punjačem. U trećoj fazi robot održava kontakt s punjačem sve dok se baterija ne napuni, nakon čega je robot ponovno spreman za daljnji rad. Faze automatskog spajanja na punjač prikazane su na slici 7.



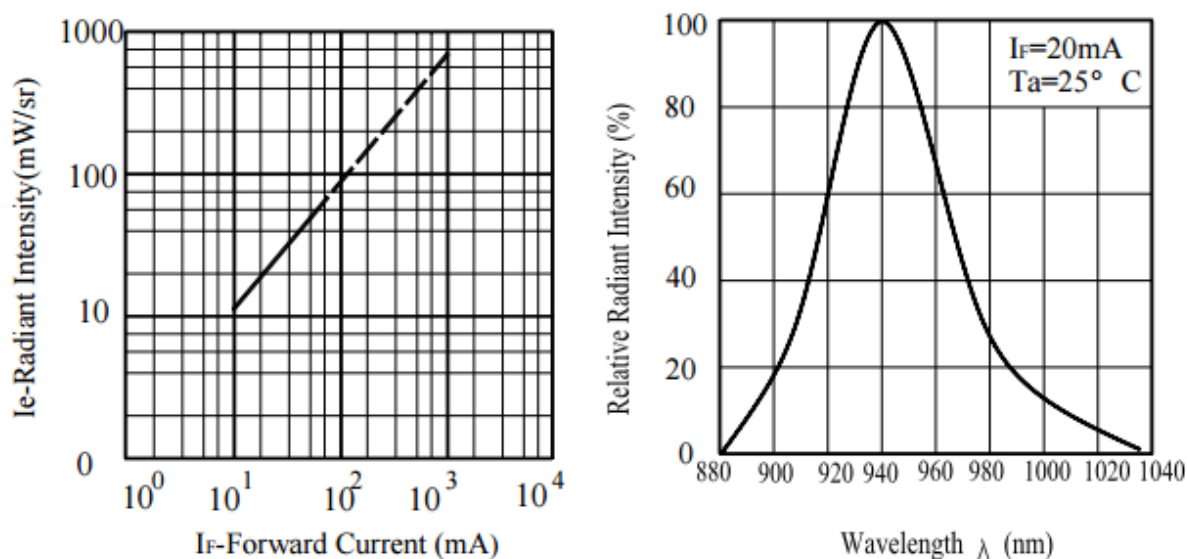
Slika 5 Faze automatskog spajanja robota na punjač

Kako bi se ostvarile opisane automatske radnje potrebno je na robota nadograditi odgovarajući uređaj za spajanje na punjač te senzor za detekciju položaja punjača. Isto tako potrebno je konstruirati i stanicu za punjenje pomoću koje robot može puniti bateriju. U nastavku će biti detaljno opisana konstrukcija svih elemenata.

3.2. Predajnik signala za navođenje

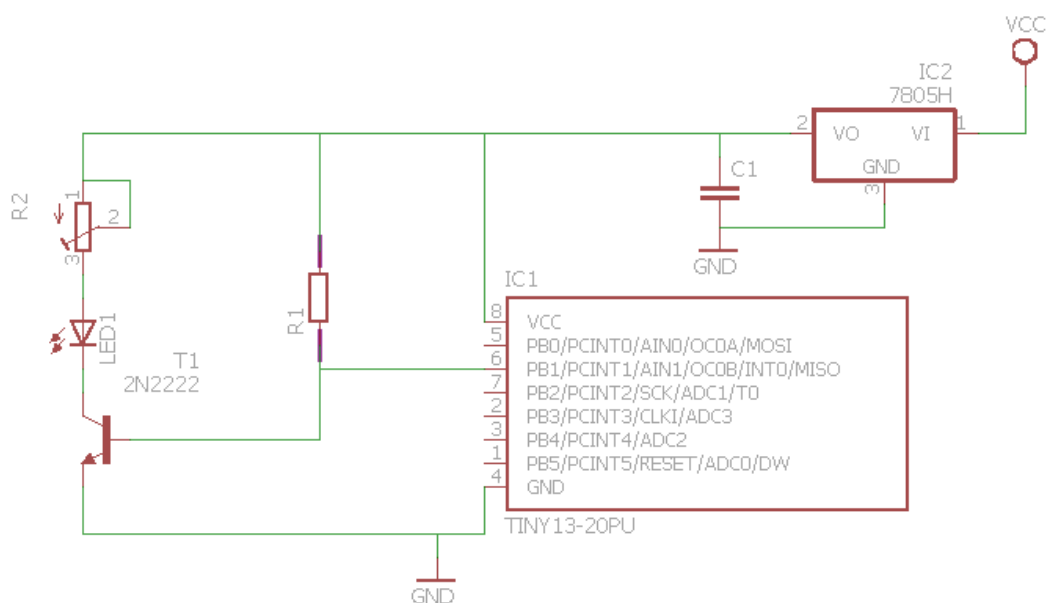
3.2.1. Elektronički sklop predajnika

Predajnik signala za navođenje potreban je kako bi stanica za punjenje na neki način signalizirala robotu svoj položaj u prostoru. Infracrveno svjetlo odabrano je zbog relativno jednostavnog načina emitiranja: usmjerenog snopa signala putem svjetleće diode (IR LED). Kako bi robot mogao razlučiti signal punjača od svih ostalih infracrvenih signala koji se nalaze u prostoru, struja kroz IR LED je modulirana TTL signalom frekvencije 36kHz. S obzirom da je planirani domet uređaja iznosio oko 2m, upotrijebljena je IR svjetleća dioda tipa IR333A. Maksimalni intenzitet zračenja odabrane diode je 940nm, što je važan podatak jer izvor zračenja treba biti kompatibilan s prijemnikom na robotu. Isto tako na slici 8 možemo vidjeti da je maksimalna dopuštena trajna struja oko 150mA. Prema podacima od proizvođača [12], u takvim radnim uvjetima IR svjetleća dioda trebala bi izdržati 1000 radnih sati što je zadovoljavajuće za potrebe ispitivanja u laboratoriju.

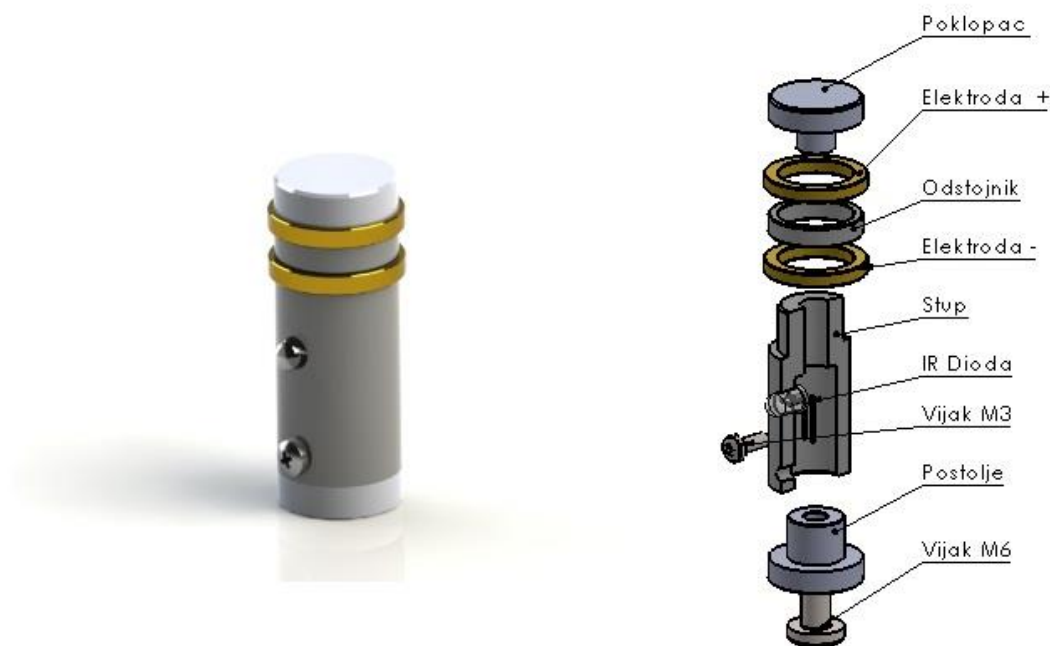


Slika 6 Karakteristike IR svjetleće diode

Slika 9 prikazuje elektroničku shemu predajnika signala za navođenje. Sklop predajnika temelji se na Atmel mikrokontroleru iz porodice AVR Attiny13 koji je svojom izvedbom u 8-pinskom kućištu te internim oscilatorom frekvencije 8MHz pogodan za izvođenje jednostavnih i preciznih oscilatora. Također, upotreba mikrokontrolera osigurava jednostavno podešavanje frekvencije te odabir odgovarajućeg trajanja impulsa budući da se svi navedeni parametri zadaju softverski, tj. računalnim programom. S obzirom da potrebna struja IR diode iznosi 150mA, potrebno je dodati pojačalo u obliku malog signalnog tranzistora 2N2222. Otpornikom R1 ograničena je maksimalna struja baze kako ne bi došlo do oštećenja tranzistora, dok se promjenjivim otpornikom R2 podešava struja kroz IR svjetleću diodu te time upravlja snagom signala za navođenje. Kondenzator C1 ima ulogu filtera napajanja. Kako je napajanje punjača baterija oko 14V, za napajanje sklopa odašiljača potreban je linearni stabilizator napona LM7805.



Slika 7 Elektronička shema predajnika signala za navođenje



Slika 8 Mehanička konstrukcija predajnika

3.2.2. Mehanička konstrukcija predajnika

Osim IR svjetleće diode, na stupu punjača nalaze se i elektrode putem kojih se ostvaruje električki kontakt između punjača i baterije na robotu. Budući da u trenutku spajanja na punjač dolazi do udara robota u stup punjača, njegova mehanička konstrukcija izrađena je od industrijske plastike POM. Svjetleća dioda smještena je na stup punjača, a visina iznad poda poligona određena je položajem prijemnika na robotu. Stup punjača pričvršćen je za postolje vijkom M3, a postolje stupa za poligon vijkom M6. Na slici 10. prikazan je sklop stupa punjača. Iznad IR svjetleće diode nalaze se elektrode punjača koje su izrađene od mjedi te pričvršćene na stup izrađenim vijkom M8. Udaljenost između elektroda određena je na 6mm kako bi se smanjila opasnost od kratkog spoja. Kontakt između vodiča punjača i elektroda postiže se pritezanjem vijka na vrhu stupa. Svi dijelovi mehaničke konstrukcije stupa punjača izrađeni su tehnologijom obrade odvajanjem čestica zbog relativno velikog opterećenja koje podnose prilikom spajanja robota na punjač. Sam elektronički sklop punjača smješten je u posebnu kutiju koja se nalazi na kabelu za napajanje punjača.

3.2.3. Generiranje signala za navođenje

S obzirom da je odabrani infracrveni signal za navođenje robota amplitudno modulirana infracrvena svjetlost, a mikrokontroler je odabran za oscilator, kao što je opisano u odjeljku 3.2.1., potrebno je napisati odgovorajući program za generiranje oscilacija. S obzirom da je generiranje oscilacija relativno jednostavan zadatak, program je napisan korištenjem programskog jezika BascomAVR [13]. Frekvencija signala određena je na 36kHz budući da je prijemnik najosjetljiviji upravo na toj frekvenciji. Širina impulsa određena je kao jedna polovina perioda prema opće poznatom izrazu :

$$f = 36kHz \rightarrow t = \frac{1}{36000} = 2.77 \cdot 10^{-5}[s] \approx 28 \mu s \rightarrow t_h = t_l = 14\mu s$$

Kako bismo generirali oscilacije na izlaznom pinu mikrokontrolera PB.1, korištena je Bascom [13] ugrađena funkcija **waitus** što se mjerenjem na osciloskopu pokazalo kao dovoljno precizno za ovu primjenu. Na slici 11. prikazan je programski kod za generiranje impulsa te osciloskopska snimka ekrana izvedena akvizicijskom karticom Picoscope [14]. Tehnike složenije modulacije IR svjetleće diode detaljno su opisane u radu [15].

```

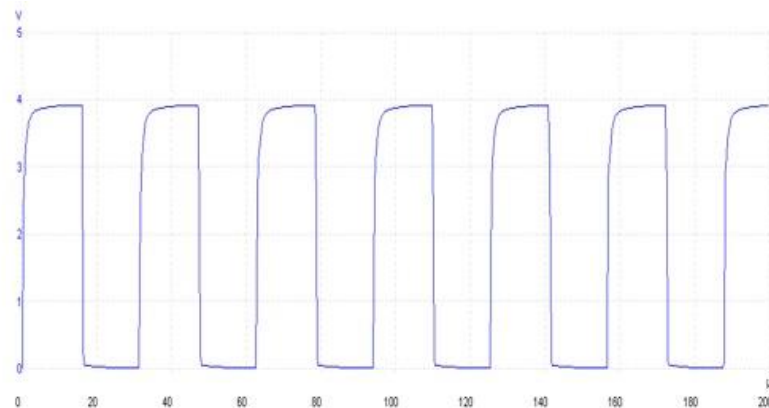
$regfile = "t13def.dat"
$crystal = 8000000
Config Portb = Output

```

```

Do
PortB.1 = 0
waitus 14
PortB.1=1
waitus 14
Loop

```

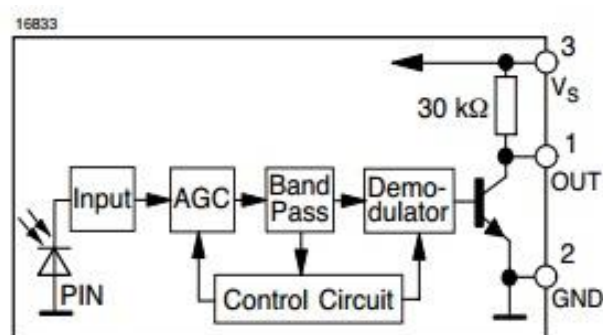


Slika 9 Programski kod i snimka signala za navođenje

3.3. Prijemnik signala za navođenje

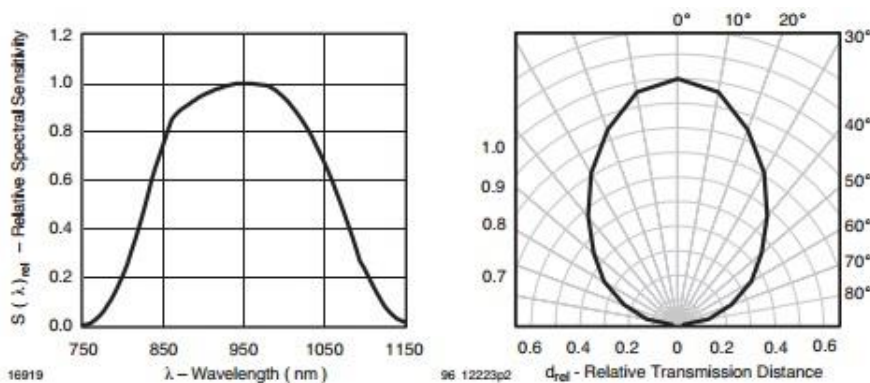
Kako bi robot mogao pronaći položaj predajnika i punjača pomoću opisanog signala za navođenje, na robot je ugrađen prijemnik signala. S obzirom da je glavna funkcija prijemnika određivanje smjera u kojem se nalazi punjač u odnosu na uzdužnu os robota, odabrana su dva IR detektora tipa TSOP 34836.

3.3.1. Detektor infracrvenog signala



Slika 10 Shema detektora infracrvenog signala

Na tržištu postoji cijeli niz detektora infracrvenog zračenja. Glavne razlike među njima su razlike u: frekvenciji prijema, ugradbenim dimenzijama, spektralnim karakteristikama te ovisnosti osjetljivosti o kutu upada infracrvenog svjetla. Odabrani detektor TSOP 34836 pokazao se kao najpogodniji zbog malih ugradbenih dimenzija, jednostavnog korištenja,

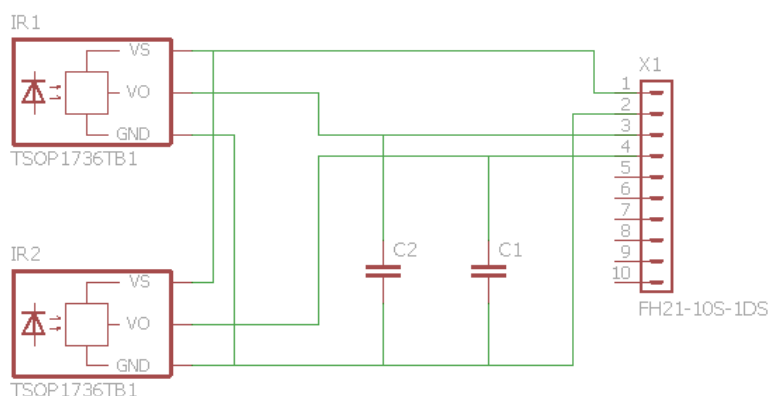


Slika 11 Karakteristike detektora infracrvenog svjetla

ugrađenog pojasno propusnog filtera te pogodne spektralne karakteristike. Ugrađeni filter omogućava selektivan prijem isključivo amplitudno moduliranog infracrvenog svjetla frekvencije 36kHz, a maksimum osjetljivosti nalazi se na 950nm što je kompatibilno s odabranom IR svjetlećom diodom, opisanom u odjeljku 3.2. Isto tako je moguće vidjeti iz Grafa kutne raspodjele osjetljivosti da osjetljivost senzora padne na jednu polovinu maksimalne pri kutu upada od 60° u odnosu na detektor što je također pogodno za ugradnju u prijemnik na robotu jer olakšava konstrukciju sjenila. Kada detektor prima signal, izlazni pin odlazi u stanje logičke nule, a kada je detektor bez prijema, odlazi u stanje logičke jedinice. Da bi se stanje izlaza promijenilo potreban je prijem od barem tri impulsa 36kHz što uzrokuje određeno kašnjenje u sustavu upravljanja, koje se za potrebe vođenja mobilnog robota može zanemariti. Slika 13. prikazuje spektralnu i kutnu karakteristiku osjetljivosti.

3.3.2. *Elektronički sklop prijemnika signala za navođenje*

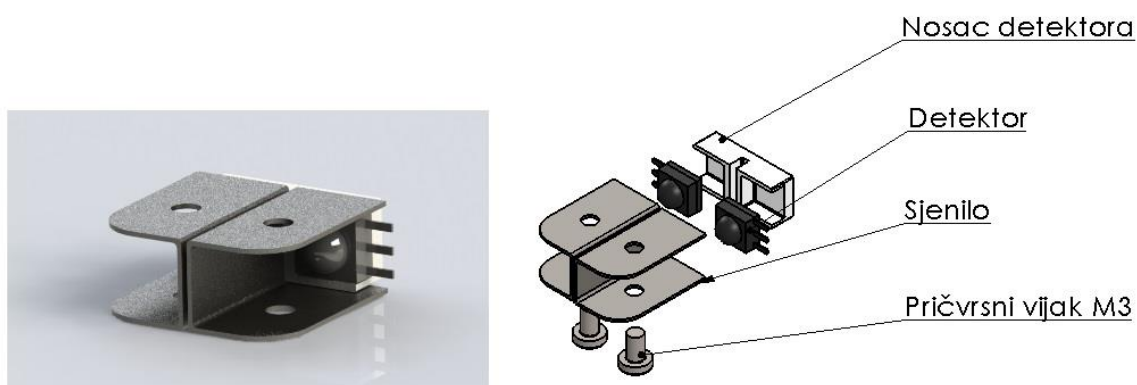
Kao što je već rečeno za određivanje smjera prema punjaču, robotu je potreban prijemnik s dva detektora infracrvenog signala. Jedan detektor javlja informaciju o položaju svjetla desno, a drugi lijevo od uzdužne osi robota. Oba detektora spojena su na upravljačku jedinicu robota eMiR preko ulazno-izlazne sabirnice. Na slici 14. prikazana je shema sklopa prijemnika. Osim dva detektora signala, dodana su u krug i dva kondenzatora C1 i C2 kapaciteta 100nF. Uloga kondenzatora je smanjivanje visokofrekventnog šuma na izlazu detektora.



Slika 12 Elektronička shema prijemnika

Na upravljačkoj jedinici robota eMiR ugrađen je konektor tipa IDC 10 putem kojega je moguće spojiti sklop na mikrokontroler unutar upravljačke jedinice. Shema spajanja opisana je u radu [10]. Signali podržavaju standardne TTL 5V signale.

3.3.3. Mehanička konstrukcija prijemnika

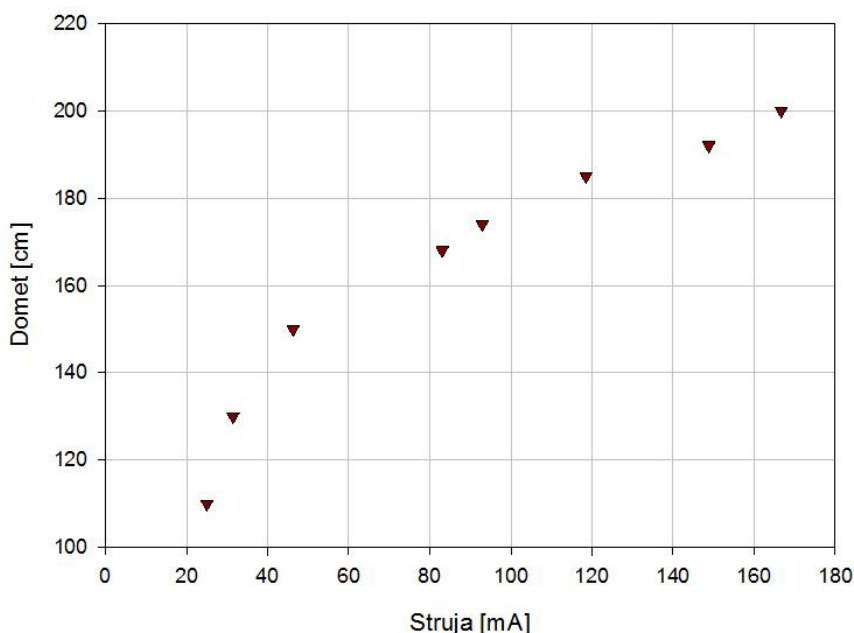


Slika 13 Mehanička konstrukcija prijemnika

Kako bismo prijemnikom mogli uspješno određivati položaj punjača, osim elektroničkog dijela, potrebna je i mehanička konstrukcija kojom se detektori montiraju na robota te sjenilo pomoću kojega se ograničava vidno polje detektora. Mehanički sklop sastoji se od detektora infracrvenog svjetla, nosača detektora, te sjenila. Kao što je vidljivo na slici 12, detektori su

postavljeni paralelno, a sjenilo se nalazi između detektora. Glavna uloga sjenila je spriječiti neželjene detekcije signala. Naime, senzor postavljen na desnu stranu robota treba detektirati infracrveni signal koji dolazi s desna ili srijeda, a nipošto signal koji dolazi s lijeva, odozdo ili odozgo. Kako infracrveno svjetlo prolazi kroz polimerne dijelove proizvedene aditivnim tehnologijama (3D print), sjenilo je izrađeno od čeličnog lima debljine 0.2mm. Čitav sklop prijemnika montira se pričvrstnim vijcima M3 na uređaj za spajanje robota na punjač.

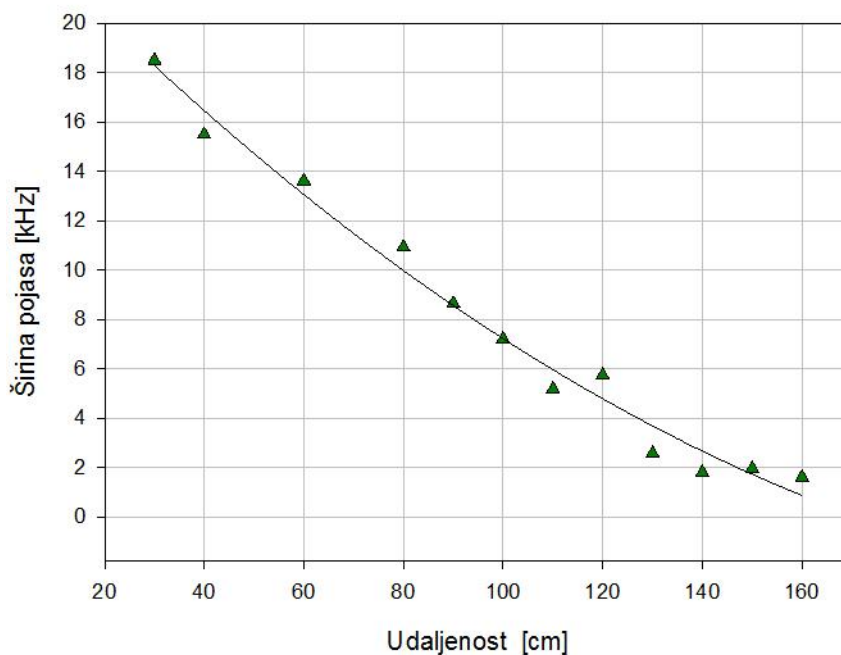
3.4. Ispitivanje dometa sustava prijemnik – predajnik



Slika 14 Ovisnost dometa o struje IR diode

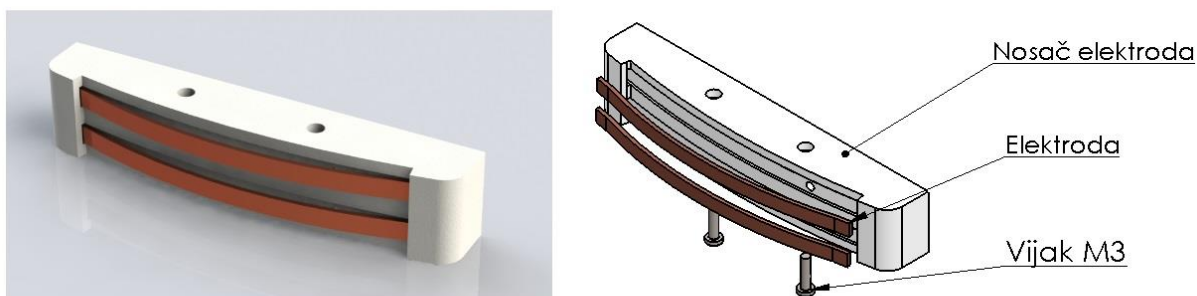
Prije ispitivanja sustava na robotu, bilo je potrebno ispitati rad sustava prijemnik-predajnik u kontroliranim uvjetima u kojima prijemnik miruje. Cilj ovakvog ispitivanja bio je odrediti najveću udaljenost na kojoj prijemnik još uvijek prima signal predajnika, provjeriti podešenost oscilatora predajnika te ispitati rad sjenila. U sklopu pokusa, također je određena ovisnost dometa predajnika o struji kroz IR svjetleću diodu koja se podešava promjenjivim otpornikom prikazanom na slici 6. Domet predajnika definiran je kao udaljenost na kojoj prijemnik prvi put primi signal. Kod velikih struja IR svjetleće diode, domet je iznosio 10m na otvorenom, a unutar prostorija bio je gotovo neograničen zbog višestrukih refleksija signala na zidovima prostorije.

Problem refleksije onemogućava rad sustava pa je struja IR svjetleće diode smanjena na 150mA pri čemu domet iznosi 2m, što je dovoljno za ispitivanje na poligonu. Ovisnost dometa o struji IR svjetleće diode mjerena je za struje manje od 150mA što odgovara udaljenostima manjim od 2m. U početnom trenutku mjerenja, prijemnik je postavljen na 2m uod predajnika, struja je postupno smanjivana, a prijemnik približavan. Zabilježene su i uzete za domet predajnika na zadanoj struji udaljenosti na kojima bi prijemnik prvi puta osjetio signal. Utjecaj sjenila ispitivan je na svim udaljenostima kako bi se vidjelo je li zasjenjivanje uspješno i na malim udaljenostima kada je signal relativno jak. Prva ispitivanja s polimernim sjenilima pokazala su preveliku propusnost IR svjetla te je kasnije odabrano metalno sjenilo. S metalnim sjenilom provedeno je ispitivanje ovisnosti izlaza prijemnika o kutu zakreta u odnosu na predajnik. U drugom dijelu ispitivanja, ustanovljeno je da dopušteno odstupanje frekvencije oscilatora također utječe na domet i provedeno je mjerenje ovisnosti dopuštenog odstupanja od nominalne frekvencije, koje je pokazalo da je domet od 2m moguće dobiti samo ako je frekvencija oscilatora stabilna unutar 1kHz, što je dodatni argument za uporabu mikrokontrolera. Slike 16. i 17. prikazuju rezultate mjerenja.



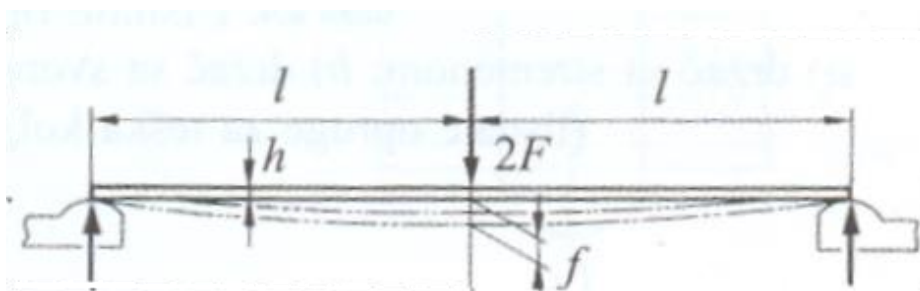
Slika 15 Dopuštena odstupanja od nominalne frekvencije

3.5. Uređaj za spajanje robota na punjač



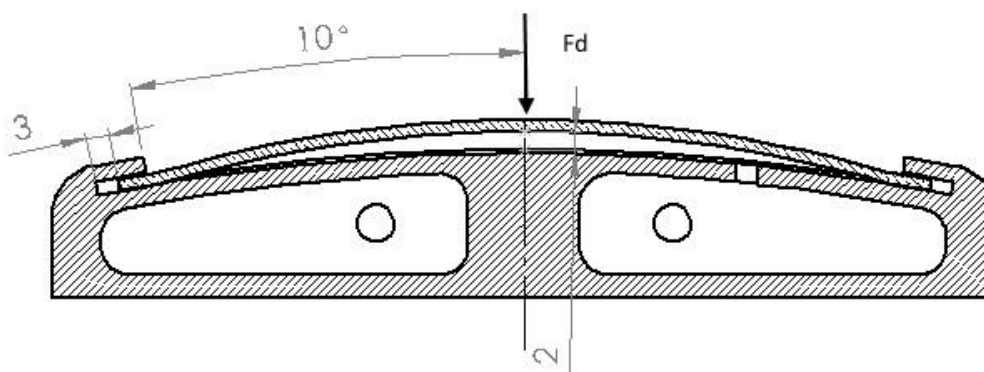
Slika 16 Uređaj za spajanje robota na punjač

Problem spajanja robota na punjač moguće je riješiti na razne načine pa tako, npr. Kim et al.



Slika 17 Princip djelovanja lisnate opruge [17]

[6] predlažu relativno složenu samopodešavajuću napravu, koja u trenutku nailaska, sustavom opruga „hvata“ robota te tako uspostavlja električni kontakt između elektroda punjača i elektroda robota. Kako bismo izbjegli izradu takvih složenih mehaničkih konstrukcija, odabran je način spajanja pomoću dvije jednostavne dvokrake lisnate opruge [17]. Naime kako bi se robot održavao u električnom kontaktu s punjačem potrebna je određena sila između elektrodi na robotu i elektrodi na punjaču. Elektrode u obliku lisnatih opruga omogućavaju konstantnu silu pritiska na elektrode za vrijeme mirovanja robota, te ujedno imaju i amortizirajuće djelovanje prilikom spajanja robota na stup punjača.

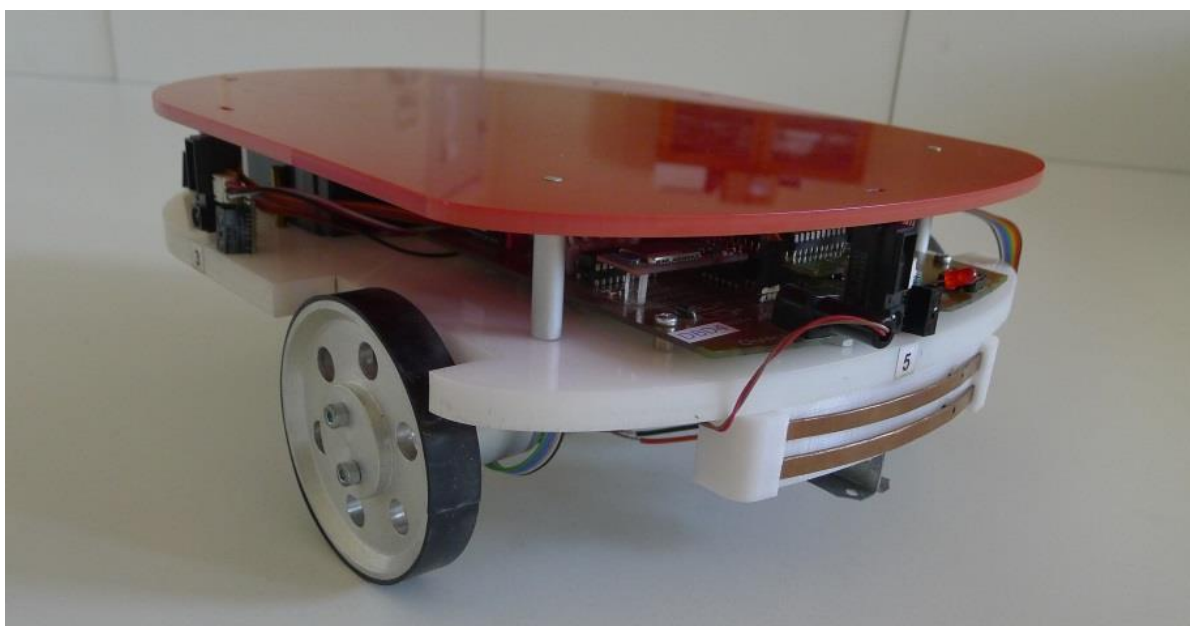


Slika 18 Skica ugradnje elektrode na nosač

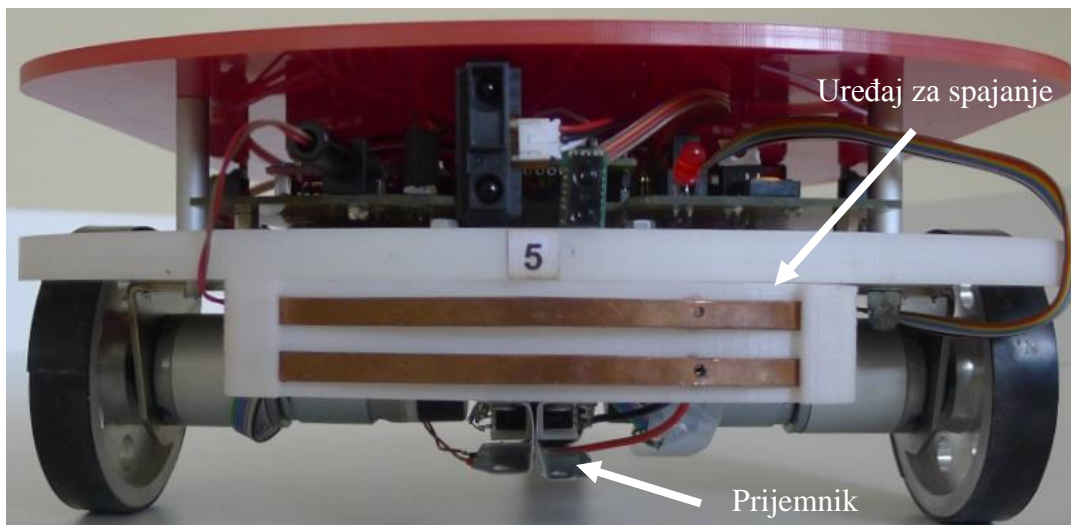
Isto tako konstruirani uređaj za spajanje robota kompenzira pogreške u navođenju, jer zauzima velik dio stražnjeg dijela robota, a istovremeno je kružnog oblika što omogućava spajanje na bilo kojem dijelu elektroda. Kako bi elektrode mogle djelovati kao lisnate opruge potrebno ih je pričvrstiti tako da se krajevima ostavi dovoljno stupnjeva slobode potrebnih za pomicanje prilikom progibanja opruge. Na slici 16. prikazan je princip djelovanja lisnate opruge iz čega vidimo da je krajevima potrebno ograničiti gibanje u X smjeru (po dužini robota), dok se gibanje u Y smjeru može ograničiti samo djelomično kako pri progibanju opruge za iznos f ne bi došlo do zaglavljivanja. Zbog takvih zahtjeva nosač elektroda izrađen je aditivnom tehnologijom FDM (3D print) budući da ona omogućava izrađivanje geometrijskih oblika koji su problematični u konvencionalnoj tehnologiji. Aditivna tehnologija FDM jedna je od najrasprostranjenijih tehnologija za brzu izradu prototipa, a posebno je pogodna za istraživačke projekte zbog male cijene opreme i materijala te dobrih mehaničkih svojstava izradaka. Više detalja o izradi elemenata robota aditivnom tehnologijom FDM, moguće je pronaći u radu [18]. Elektrode za spajanje robota na punjač izrađene su od bakrene trake debljine 1.5mm te su umetnute unutar nosača kako je prikazano na slici 20. Vodiči za spajanje elektroda i baterije na elektrode su spojeni lemljenjem. Slika 18. prikazuje dijelove uređaja za spajanje na punjač te pričvrstne vijke kojima se spaja na robota.

3.6. Montaža sklopa za automatsko punjenje na robota eMiR

Svi dijelovi potrebni za ostvarivanje automatskog prilaza i spajanja na punjač kroz faze opisane u odjeljku 3.1. montirani su na stražnju stranu robota. Stražnja strana odabrana je kako se ugradnjom sklopa za automatsko punjenje ne bi ometala funkcija zakretanja prednjeg kotača te onemogućila daljnja nadogradnja robota s prednje strane, na kojoj se nalazi veći broj infracrvenih senzora udaljenosti potrebnih za izbjegavanje prepreka. Za potrebe ugradnje sklopa na robotu su probušena dva provrta promjera 3mm za pričvršne vijke prikazane na slici 15. Zbog relativno velikih sila koje se pojavljuju prikol spajanja robota na punjač, sklop mora biti pričvršćen vijcima. Visina sklopa za punjenje nad tлом iznosi 30mm što ne ograničava robota prilikom gibanja po poligonu. Slike 21. i 22. prikazuju robota s montiranim sklopom za automatsko punjenje.



Slika 19 Robot eMiR s montiranim ujedajem za spajanje na punjač



Slika 20 Dijelovi sustava za automatsko punjenje na robotu eMiR

3.7. Montaža stupa punjača na poligon

Kako bi se mogao ispitati rad sustava za automatsko punjenje na poligon za mobilne robote u laboratoriju za inteligentne montažne sustave, ugrađen je stup punjača opisan u odjeljku 3.2.2. Kako bi se stup mogao montirati izbušen je provrt 6mm te je stup pričvršćen vijkom. Vodiči za punjenje i napajenje svjetleće diode dovedeni su izvana. Slika 23. prikazuje stup punjača montiran na poligonu.



Slika 21. Stup punjača na poligonu

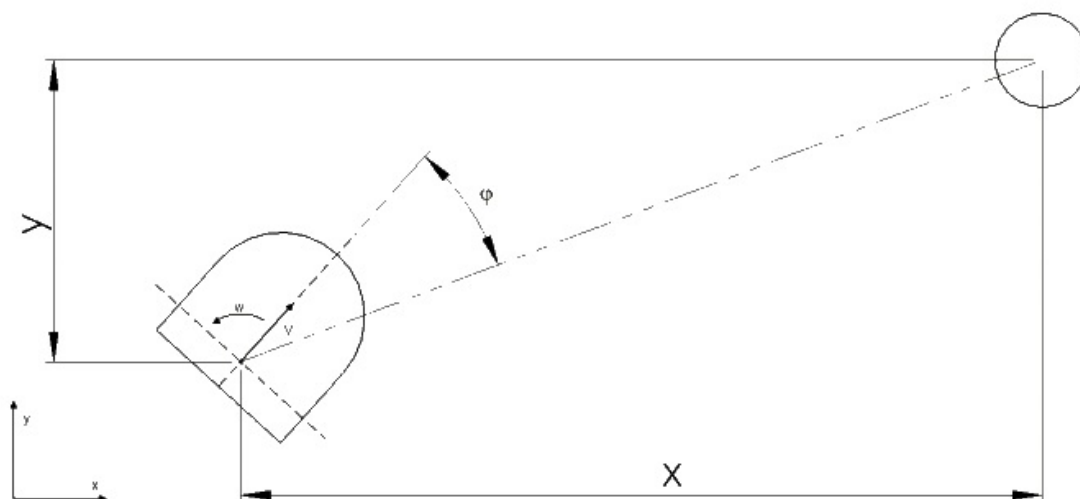
4. Vođenje robota do punjača

4.1. Strategija vođenja robota do punjača

Kako bismo doveli robota u kontakt s punjačem koji se nalazi na nekoj udaljenosti u horizontalnoj ravnini potrebno je razviti strategiju vođenja. S obzirom da između robota i stupa punjača ne postoje nikakve prepreke dovoljno je osigurati gibanje robota po pravcu prema lopti. Strategiju vođenja robota do punjača temeljenu na ideji opisanoj u odjeljku 3.1. možemo podijeliti u 4 faze :

1. **Pronalazak i identifikacija signala za navođenje**
2. **Okretanje prema punjaču**
3. **Gibanje prema punjaču**
4. **Spajanje robota na punjač**

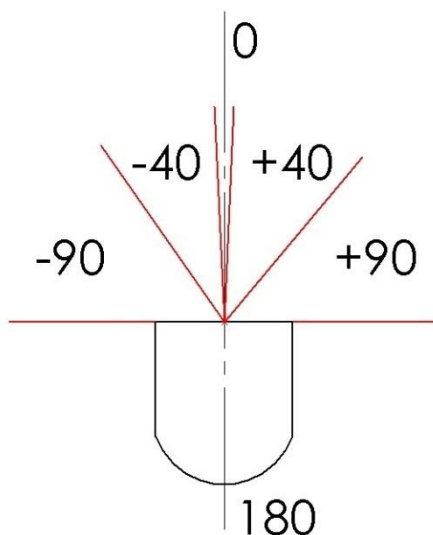
Slika 24 Prikazuje robota i stup punjača u ravnini. Prva faza je pronalazak i identifikacija signala za navođenje. Robot u ovoj fazi rotira konstantnom kutnom brzinom ω i faza traje sve dok jedan od detektora signala za navođenje na prijemniku na primi signal za navođenje. Druga faza je regulacija kutne brzine u svrhu minimizacije kuta objekta u odnosu na uzdužnu os robota



Slika 22 Robot i punjač u horizontalnoj ravnini

φ do zadane granice. Tijekom druge faze kutna brzina robota mijenja se s obinom da signale s prijemnika te se pojavljuje i translacijska brzina čime počinje treća faza. U trećoj fazi koja počinje kada je kut objekta u odnosu na uzdužnu os robota unutar zadanih granica, započinje translatorno gibanje robota konstantnom brzinom v . Kutna brzina se tijekom treće faze regulira pomoću povratne veze sa prijemnika kako bi se kut φ držao što je moguće bliže iznosu 0° , a time i robot gibao prema stupu punjača. Kada robot dotakne stup punjača, upravljačka jedinica osjeti električni napon na elektrodama te se zaustavljaju sva gibanja. Kako robot prilazi punjaču s određenom brzinom u trenutku kontakta s punjačem dolazi do udara robota u punjač koji se amortizira opisanim uređajem z aspajanje. Poslije spajanja na stup punjača robot prati napon na punjaču te u slučaju gubitka električnog kontakta ponovno se pomakne kako bi se kontakt uspostavio. U daljnjem tekstu biti će detaljno opisan način ostvarivanja gibanja u svim fazama.

4.2. Mjerenje kuta punjača u odnosu na robota



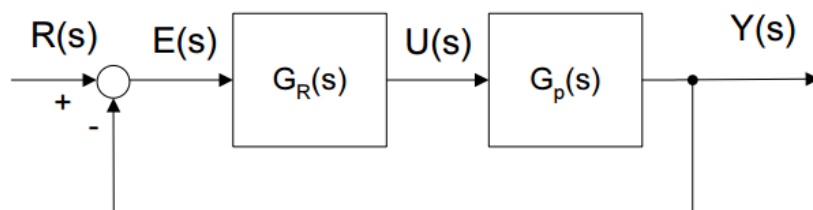
Slika 23 Područje mjerenja prijemnika

Kako bi se mogla ostvariti regulacija gibanja tj. zatvoriti povratna veza potrebno je na neki način mjeriti kut punjača u odnosu na os robota. Mjerenje kuta moguće je izvoditi na razne načine no zbog pristupačnosti, male potrošnje struje te jednostavne obrade podataka odabran je

prijemnik čija je konstrukcija opisana u odjeljku 3.3. Budući da prijemnik posjeduje dva digitalna infracrvena detektora puni kut iznosa 360° moguće je razlučiti u 4 bita. Za uspješno spajanje robota na punjač potrebno je ostvariti kontakt robota i punjača unutar kuta određenog konstrukcijom uređaja za spajanje robota na punjač te zbog toga puni kut nije jednoliko podjeljen na četiri moguća stanja senzora. Tablica 3. prikazuje vezu između kuta pod kojim se nalazi stup punjača i logičkih stanja izlaza prijemnika, a slika 25 opisane kuteve u horizontalnoj ravnini.

Kut između osi i punjača	Detektor 1	Detektor 2
$-2,5^\circ$ do $+2,5^\circ$	0	0
$+2,5^\circ$ do $+40^\circ$	0	1
$-2,5^\circ$ do -40°	1	0
$-40^\circ / +40^\circ$ do 180°	1	1

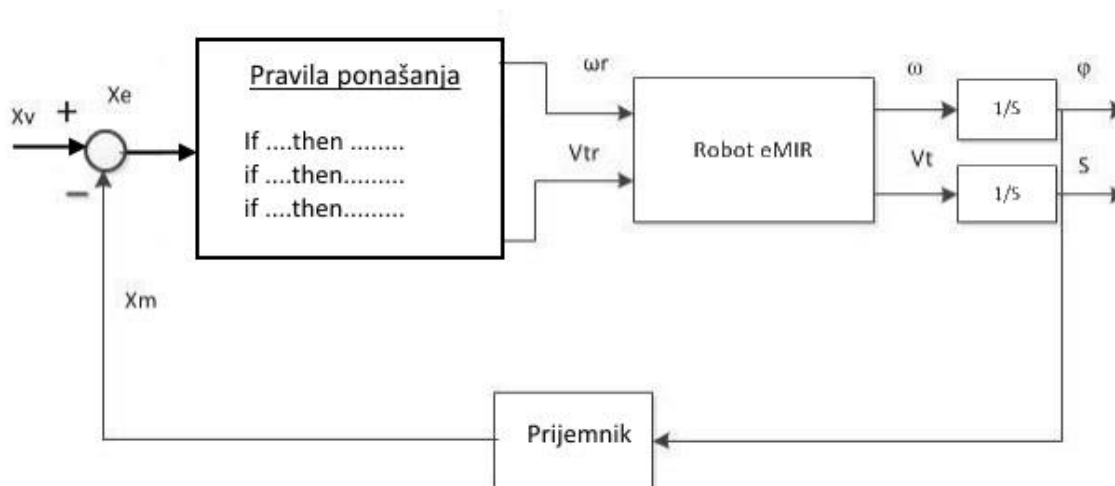
4.3. Regulacija gibanja robota eMiR



Slika 24 Klasični regulacijski krug

Kao što je vidljivo iz prethodnog odjeljka, odabranim mjernim članom nije moguće kontinuirano mjeriti kut punjača u odnosu na robota te zbog toga nije moguće primjeniti niti jednu od klasičnih kontinuiranih regulacijskih metoda. Prema tome jedina regulacijska struktura koja se može primjeniti je diskontinuirani više položajni regulator kojim se s obzirom na dobiveni binarni signal s prijemnika određuju pripadajuće reference kutne i translacijske

brzine robota. Struktura diskontinuiranog regulatora slična je klasičnom regulacijskom krugu poznatom iz osnova automatike osim što sam zakon upravljanja nije kontinuirat tj. opisan kontinuiranom funkcijom $u(t)$. Kod diskontinuiranog regulatora umjesto kontinuiranog zakona upravljanja pojavljuje se skup „pravila ponašanja“ temeljem kojih se određuje odgovarajuća vrijednost upravljačke varijable. Sličan pristup pojavljuje se i u neizravnom regulatoru no isti se također u ovom slučaju ne može primjeniti u potpunosti zbog prirode mjernog člana tj. načina mjerenja kuta punjača u odnosu na os robota. Kako je cilj regulacije robota voditi prema punjaču zadatak regulatora je minimizirati kut φ prikazan na slici 24 te se suprostavljati vanjskim poremećajima te poremećajima koji se pojavljuju zbog nesavršenosti u prijenosnom mehanizmu servo motora robota. Na slici 27. vidimo diskontinuirani regulacijski krug za vođenje robota prema punjaču. Vodeća veličina X_v iznosi $\varphi_v=0$ dok mjerena veličina poprima rezultate opisane u 4.2. Nakon oduzimanja regulator se ponaša prema određenom pravilu kako bi se minimizirao X_e te time i kut punjača u odnosu na os robota. Nakon što X_e padne unutar zadanih granica, zadaje se i V_{tr} tj. robot se počinje gibati translatorno prema punjaču.



Slika 25 Diskontinuirani regulacijski krug

4.3.1. Pravila ponašanja

Kod diskontinuiranih i neizrazitih regulatora s pravilima ponašanja, regulatoru se ugrađuje znanje operatera, a proces zaključivanja je sličniji čovjekovom načinu razmišljanja nego klasičnom kontinuiranom regulatoru. Jedan od najvećih nedostataka regulatora s pravilima ponašanja je brzi rast broja uvjeta s brojem ulaza kod neizrazitih regulatora [19], tj. broja mogućih stanja diskontinuiranog mjernog člana kod diskontinuiranog regulatora. S obzirom da mjerni član – prijemnik signala za navođenje, ima četiri moguća stanja, definirana su pravila ponašanja kako slijedi :

1. Ako X_e pripada $[-40,-180]$ $\rightarrow \omega_r = \omega_1 \quad v_r = 0$
2. Ako X_e pripada $[-40,-2.5]$ $\rightarrow \omega_r = \omega_2 \quad v_r = v_2$
3. Ako X_e pripada $[-2.5,+2.5]$ $\rightarrow \omega_r = 0 \quad v_r = v_1$
4. Ako X_e pripada $[+2.5,+40]$ $\rightarrow \omega_r = -\omega_2 \quad v_r = v_2$
5. Ako X_e pripada $[+40,+180]$ $\rightarrow \omega_r = \omega_1 \quad v_r = 0$

Možemo vidjeti da pravila 1. i 5. imaju isto ponašanje zbog toga što robot ne može odrediti položaj stupa punjača ukoliko se on nalazi s prednje strane robota. Veličine kutnih brzina ω_1 i ω_2 imaju isti iznos te suprotan predznak kako bi se zatvorila negativna povratna veza. Isto tako vidimo da se robot ne giba translacijski ako je veličina signala pogreške X_e prevelika što odgovara opisanoj strategiji gibanja prema punjaču iz odjeljka 4.1. Određivanje vrijednosti upravljačkih veličina, tj. željene kutne i translacijske brzine koje se šalju robotu ω_1 , ω_2 te v_1 , v_2 , sinteza je opisanog regulatora. Iste konstante pronađene su pokusima izvedenim na poligonu te su opisane u odjeljku „Podešavanje i ispitivanje rada sustava za automatsko punjenje“.

5. Programaska podrška za vođenje robota prema punjaču

5.1. Programski jezik Python

Opisana strategija gibanja na robotu se izvršava putem programskog koda u računalu. Program je napisan korištenjem programskog jezika Python 2.7 i razvojnog okruženja Py scripter. Navedeni programski jezik odabran je zbog svoje objektne strukture koja omogućava te podržava velik broj vanjskih modula i paketa za serijsku komunikaciju i ostatale radnje potrebne za vođenje robota. Osim toga programiranje u jeziku Python sastavni je dio nastavnog programa na smjeru Mehatronike i robotike Fakulteta strojarstva i brodogradnje što će olakšati budući rad na ovoj problematici. Osim navedenog razloga, jedan od prednosti uporabe programskog jezika Python je jednostavan prijelaz na operativni sustav Lynux te time i ugradbena računala kao što su Raspberry Pi ili BeagleBone.

5.2. Skripta za upravljanje robotom

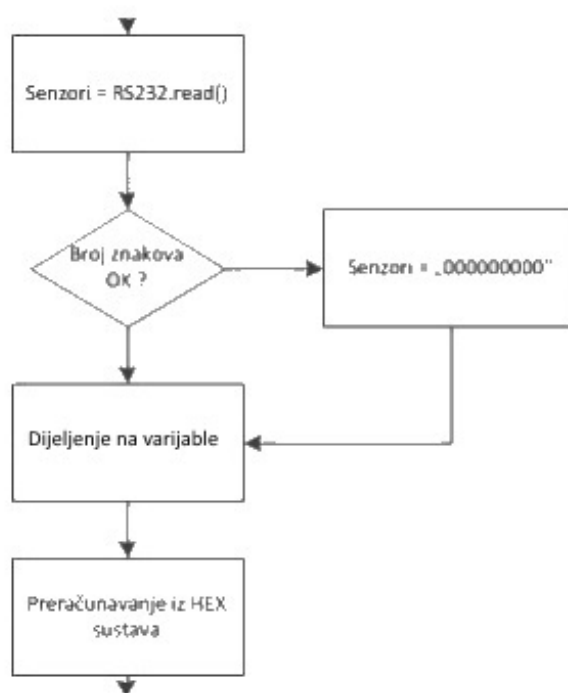
S obzirom da je Python skriptni interpreterski jezik, umjesto .dll datoteka ili idrivera u klasičnom smislu koriste se obične Python skripte koje u sebi zadrže isključivo definicije funkcija. Za potrebe ovog rada napisana je skripta emircon.py koja, po uzoru na postojeći jezik za upravljanje robotom eMiR opisanom u radu [9], posjeduje funkcije za upravljanje robotom. U tablici su prikazane funkcije koje sadrži skripta te pripadajuće radnje koje robot obavlja nakon njihovog pozivanja.

Funkcija	Ponašanje robota
eMiR_init(port_no)	Otvora komunikaciju na serijskom portu „port_no“
eMiR_move(v,w)	Robot se giba translacijskom brzinom v te rotacijskom w
eMiR_citaj()	Vraća string servisnih podataka
eMiR_docking('log.txt')	Izvodi automatsko spajanje na punjač

Programski kod svih funkcija nalazi se u prilogu 1. (Skripta eMiRcon.py) te ćemo ovdje detaljnije opisati funkcije eMiR_citaj i eMiR_docking() jer se pomoću eMiR_citaj() ostvaruje mjerenje kuta robota prema punjaču, a u funkciji eMiR_docking se nalazi opisani regulator gibanja.

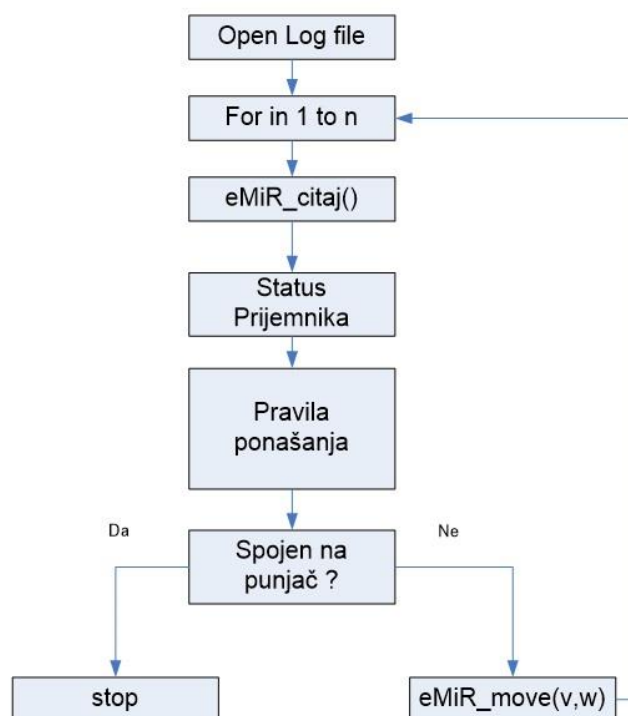
5.2.1. Funkcija *eMiR_citaj()*

Senzori robota očitavaju se putem serijske veze na način koji je opisan u odjeljku o komunikaciji sa robotom. S obzirom da je serijska komunikacija putem Bluetootha relativno nepouzdana, potrebno je svaki paket informacija provjeravati. Iako robot u paketu informacija šalje kontrolnu vrijednost CS (Check Sum), u ovomu radu ona nije korištena. Dovoljno se pokazalo samo provjeriti broj znakova u paketu te pakete s premalim ili prevelikim brojem znakova odbaciti. U slučaju prijema neispravnog paketa, program u servisnu datoteku na mjesto primljenih podataka ispisuje „000000“ tako da je lako uočiti pogreške u komunikaciji. Nakon prijema, ispravan paket dijeli se na petnaest veličina od čega prvih šest predstavljaju mjerene vrijednosti IC daljnómjera, sedma predstavlja napon baterije, osma i deveta kutnu i translacijsku brzinu, a na trinaestom mjestu se nalaze stanja digitalnih ulaza upravljačke jedinice na koje je spojen prijemnik signala za navođenje. Detaljne upute za čitanje servisnih podataka s robota nalaze se u Prilogu 2. [eMiR protokol komunikacije]. Slika 28. prikazuje dijagram toka funkcije *eMiR_citaj()*.



Slika 26 Dijagram toka funkcije *citaj()*

5.2.2. Funkcija *eMiR_dock()*

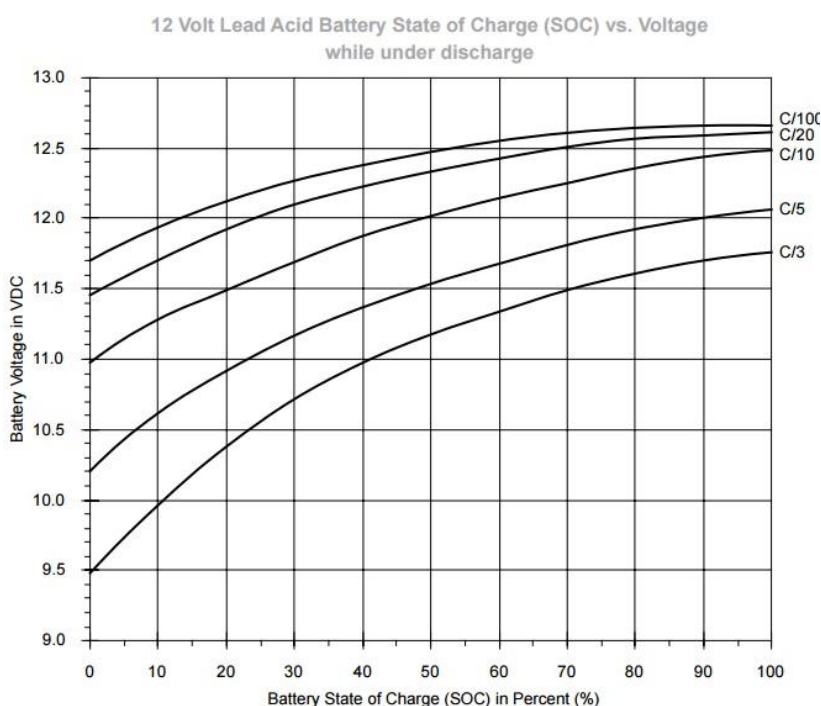


Slika 27 Dijagram toka emir_dock

Unutar funkcije `emir_dock()` nalazi se cijela strategija i regulacija gibanja opisana u odjeljku 4. Funkcija se može pozvati s argumentom koji se uzima kao ime izlazne servisne datoteke. Kako se program ne bi zaglavio u beskonačnoj petlji, moguće je podešavati vrijeme koje se program izvršava. Prema dijagramu toka funkcije možemo vidjeti da se na samom početku otvara servisna datoteka 'log.txt' u koju se pohranjuju podaci o gibanju robota pomoću kojih se kasnije može analizirati ponašanje robota te izvršiti sinteza regulatora gibanja. Kada je otvorena izlazna datoteka, program ulazi u konačnu petlju koja se izvršava n puta. Nakon toga poziva se funkcija `emir_citaj()` pomoću koje se čitaju podaci koje robot šalje računalu. Kako bi se izbjegla pojava pogrešnih podataka ili zaustavljanje programa zbog primanja nepotpunog paketa, operacije nad primljenim podacima izvode se pod funkcijama „try: except:“, tzv. Python exception handler opisanim u [20]. Ukoliko je primljen ispravan paket izvlače se potrebni podaci i pretvaraju se iz heksadecimalnog u dekadski brojevni sustav da bi se mjerenja senzora mogla upotrijebiti u pravilima ponašanja regulatora opisanim u odjeljku 4.3.1. Poslije pretvorbe slijedi zaključivanje na temelju pravila ponašanja te na pozivanje funkcije `eMiR_mov(v,w)` kojom se robotu šalju željena kutna i translacijska brzina. Prije kraja petlje, tj. prelaska u novi ciklus, provjerava se uvjet spojenosti robota na punjač, a ukoliko je robot spojen petlja se prekida. Na samom kraju

programske petlje u servisnu datoteku ispisuju se podaci o gibanju robota. Vrijeme potrebno za izvođenje jednog ciklusa petlje iznosi 0.2s. Nakon završetka svih zadanih ciklusa petlje zatvara se servisna datoteka i gibanje robota prestaje. Slika 29. prikazuje dijagram toka funkcije eMiR_docking().

5.2.3. Funkcije BattStatus() i Chgcon()



Slika 28 Napon u ovisnosti o stupnju napunjenosti [20]

Funkcije BattStatus i Chgcon su jednostavne funkcije koje iz primljenih podataka od robota očitavaju napon baterije i napon punjača. S obzirom da se u izlaznom stringu opisanom u Prilogu 2. podaci napona baterije izmjenjuju s podacima napona punjača, na istim pozicijama funkcije moraju provjeravati koji je napon primljen te ga preračunati korištenjem izraza također opisanih u prilogu. Funkcija battstatus vraća napon baterije i na temelju njezinog izlaza zaključuje se o napunjenosti baterije, tj. potrebi za pokretanjem procedure spajanja na punjač. U radu [21] predložene su razine napona baterije prema kojima se može odrediti količina dostupne energije u bateriji. Dijagram na slici 30. prikazuje napone baterije u ovisnosti o napunjenosti. Svaka krivulja prikazuje opterećenje baterije norminarno na kapacitet. Za naše potrebe možemo odabrati krivulju C/10 jer je baterija u prosjeku opterećena strujom od jedne

desetine kapaciteta. S obzirom da se spajanje na punjač neće izvršiti trenutno, potrebno je pokrenuti proceduru spajanja na punjač dok baterija posjeduje dovoljno energije za traženje punjača. S obzirom na veličinu poligona za ispitivanje odabrano je da se spajanje započinje kada napon padne na 11.5V tj. kada je u bateriji preostalo još 20% deklariranog kapaciteta. Funkcija Chgcon vrlo je slična funkciji Battstatus samo što umjesto napona baterije vraća napon punjača.

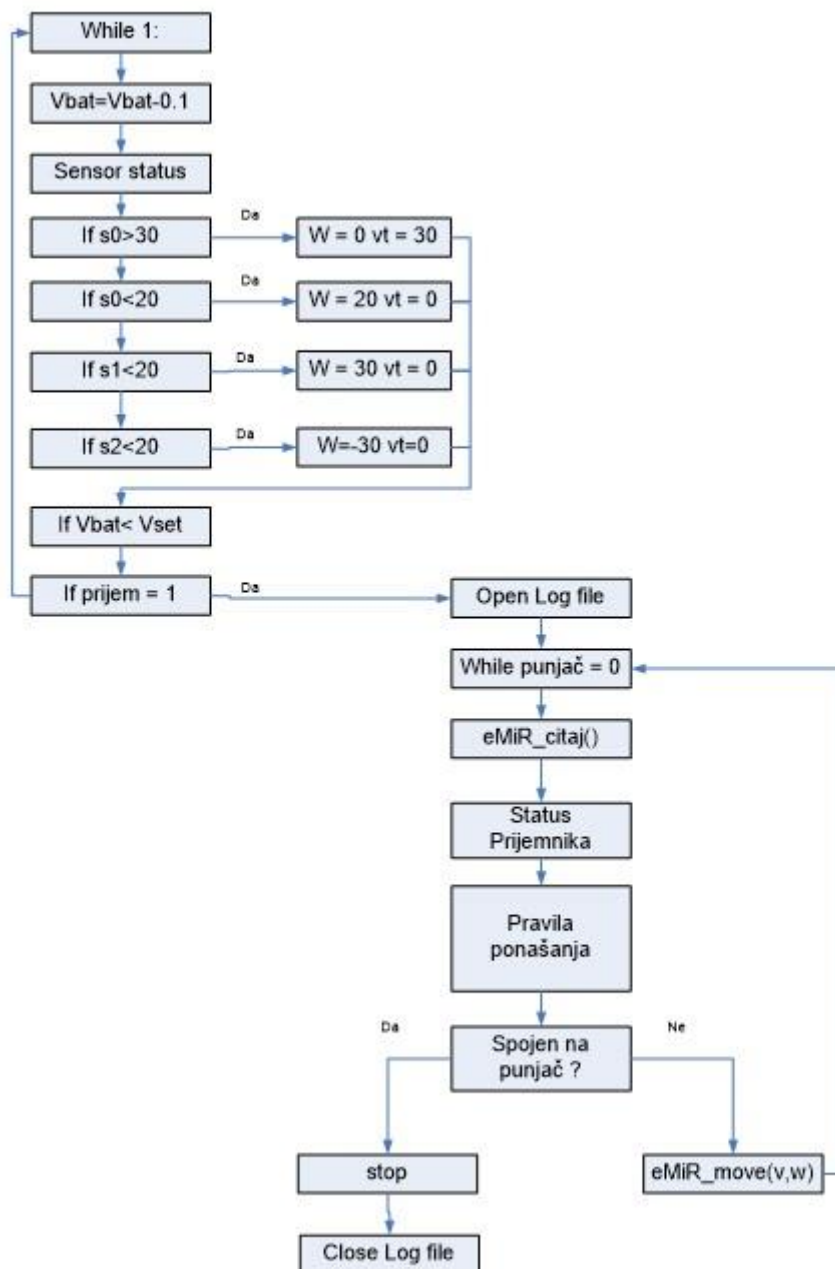
5.3. Program za ispitivanje rada sustava

Za ispitivanje opisanog sustava za automatsko punjenje robota napisana je funkcija koja se nalazi u skripti emircon. Osim opisane funkcije emir_docking() koja pokreće proces spajanja na punjač, funkcija emir_testchg() sadrži dio koda koji simulira radni zadatak zadan robotu. Kako bismo ispitali rad sustava za automatsko punjenje, robotu je zadan jednostavan zadatak proizvoljnog gibanja s izbjegavanjem prepreka. Prepreke se izbjegavaju pomoću infracrvenih senzora na prednjoj strani robota. Algoritam za vođenje robota prilikom izvršavanja tog zadatka je prilično jednostavan te se sastoji od četiri logička uvjeta koja određuju ponašanje robota na temelju očitavanja stanja infracrvenih daljinomjera na robotu. Pravila ponašanja za generiranje gibanja s izbjegavanjem prepreka zadana su kao:

1. If $\text{senzor0} > 30$ $vt = 30$ $w = 0$
2. If $\text{senzor0} < 20$ $vt = 0$ $w = 20$
3. If $\text{senzor1} < 20$ $vt = 0$ $w = -30$
4. If $\text{senzor2} < 20$ $vt = 0$ $w = 30$

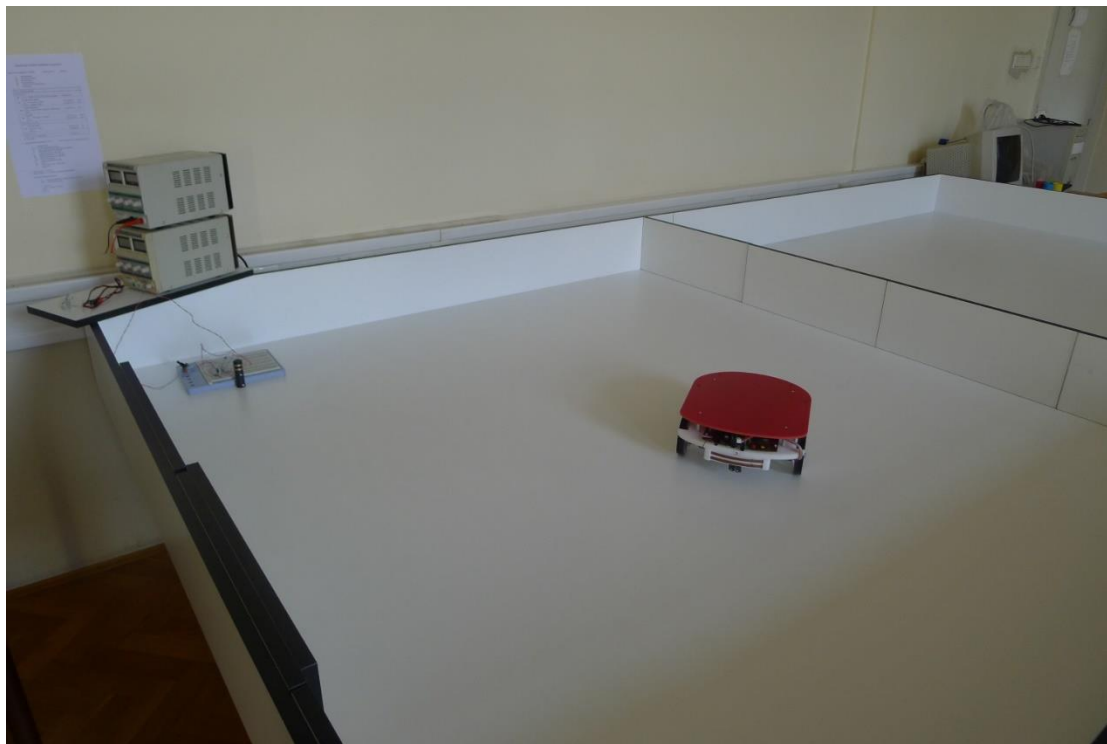
Senzor0, senzor1 i senzor2 su očitavanja senzora postavljenih na prednjoj strani robota kao što je opisano u radu [8], a vt i w su kutna i translacijska brzina koja se zadaje robotu. Istovremeno s gibanjem prema zadanim pravilima ponašanja, smanjuje se vrijednost varijable Vbat koja simulira pražnjenje baterije kako bi se sustav mogao ispitivati s napunjenim baterijama. Nakon što vrijednost varijable Vbat padne ispod zadane vrijednosti, robot se i dalje giba nasumično sve dok jedan prijemnik za navođenje prema stupu punjača ne uhvati signal za navođenje barem jednim detektorom. Nakon detekcije signala za navođenje pokreće se funkcija docking te se

robot navodi prema punjaču sve do spajanja na isti što se provjerava opisanom funkcijom Chgcon(). Dijagram toka programa za ispitivanje rada sustava prikazan je na slici 31.



Slika 29 Dijagram toka programa za ispitivanje

6. Podešavanje i ispitivanje sustava za automatsko punjenje

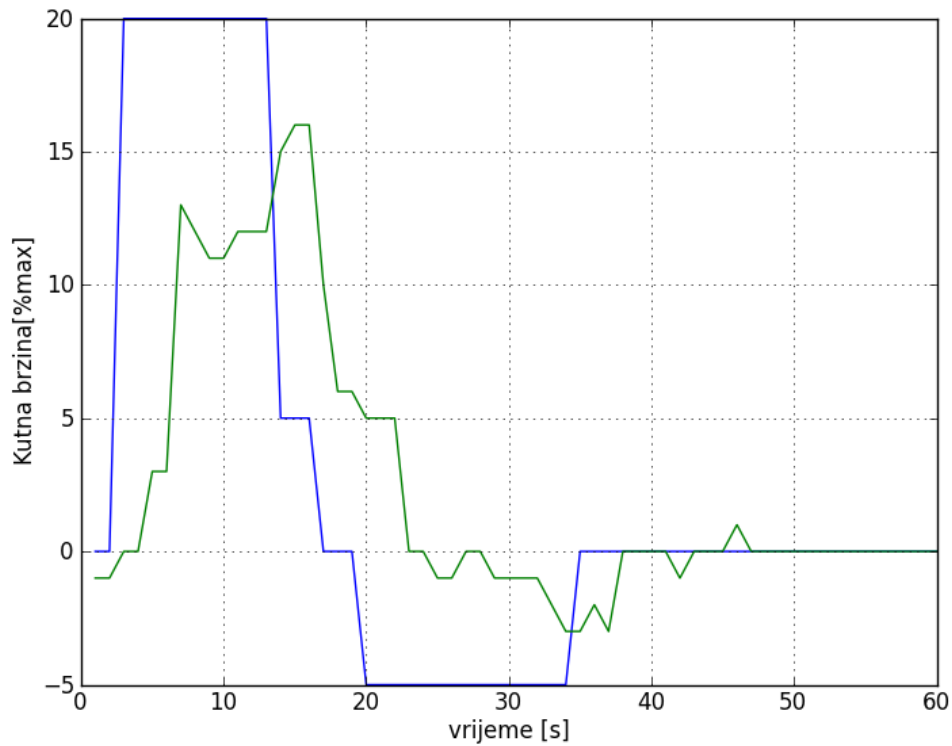


Slika 30 Robot na poligonu za vrijeme ispitivanja

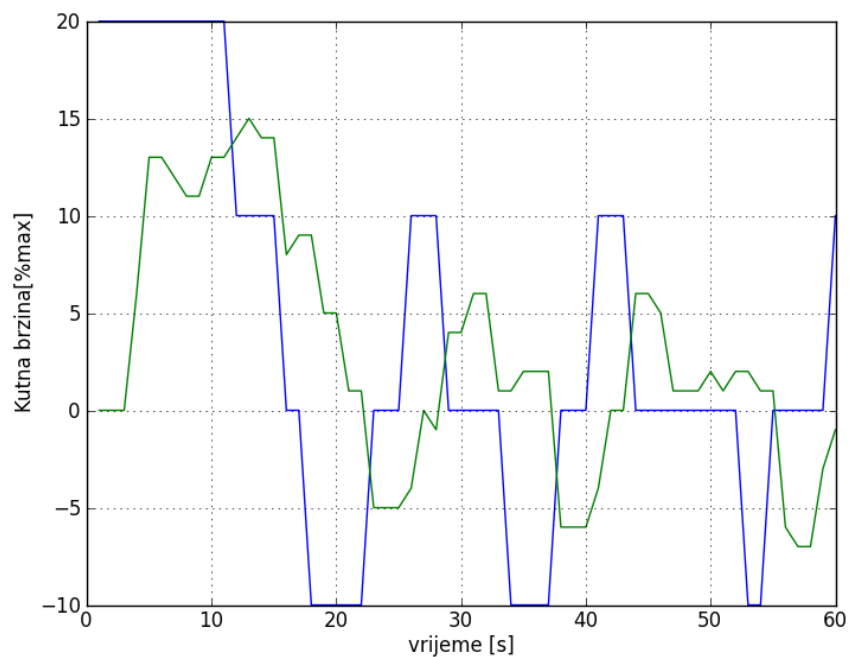
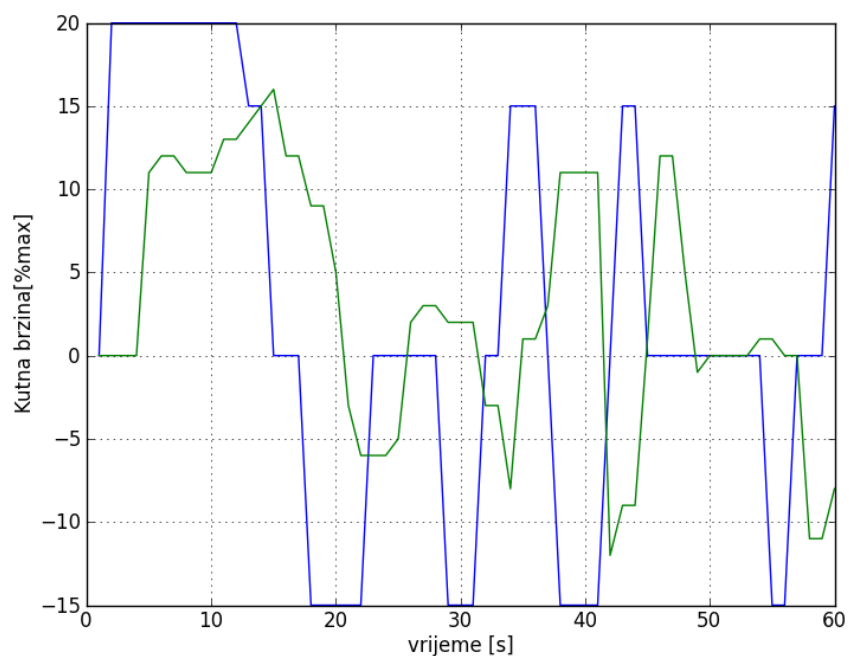
6.1. Podešavanje konstanti regulatora gibanja

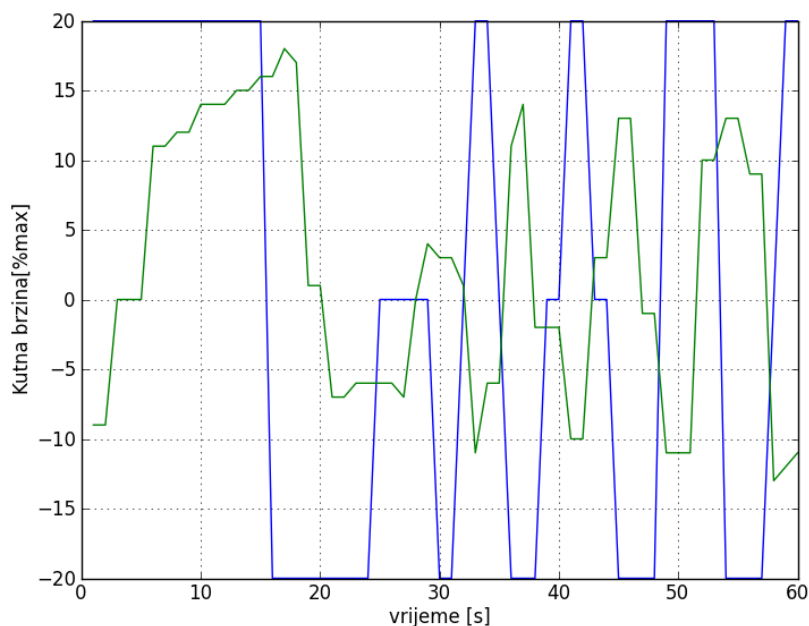
Kao što je izneseno u odjeljku 4. konstante regulacije, tj. veličine v_1 i v_2 ω_1 i ω_2 , određene su eksperimentalno. Pokus za određivanje konstanti sastojao se od izvršavanja funkcije docking s nasumično postavljenim konstantama te praćenjem ponašanja robota preko izlazne servisne datoteke. Kako bi se promatrao isključivo utjecaj konstanti na gibanje robota, sva gibanja počinjala su na istoj točki u poligonu uz istu početnu orijentaciju robota. Na početku pokusa robot je postavljen na udaljenost 1500mm od stupa punjača na poligonu okrenut tako da prijemnik ne detektira signal za navođenje kako bi se aktiviralo pravilo ponašanja 1. iz odjeljka 4. Nakon toga pokrenuta je funkcija docking te je program zaustavljan ručno nakon vremena od 60s. Kao što je uobičajeno u tehničkoj praksi, ispitivanje je započeto s vrlo malim vrijednostima za zadane brzine. Pokazalo se da, za male kutne brzine, 1-2% maksimalne, PI regulator kutne brzine unutar upravljačke jedinice robota, zbog trenja u prijenosnom mehanizmu, ne uspijeva regulirati kutnu brzinu te dolazi do nepravilnih oscilacija. Za referentnu

kutnu brzinu $\omega_1=5$ te $\omega_2=-5$ prihvatljiv aperiodični odziv. Grafovi na slikama 33.-36. prikazuju ponašanje kutne brzine robota u ovisnosti o vremenu za različite vrijednosti ω_1 i ω_2 . Plavom linijom prikazan je izlaz iz regulatora gibanja opisanog u odjeljku 4, a zelenom mjerenje kutne brzine primljeno s robota putem servisnih podataka.



Slika 31 Ponašanje robota za $w = 5$

**Slika 33 Ponašanje robota za $w = 10$** **Slika 32 Ponašanje robota za $w = 15$**



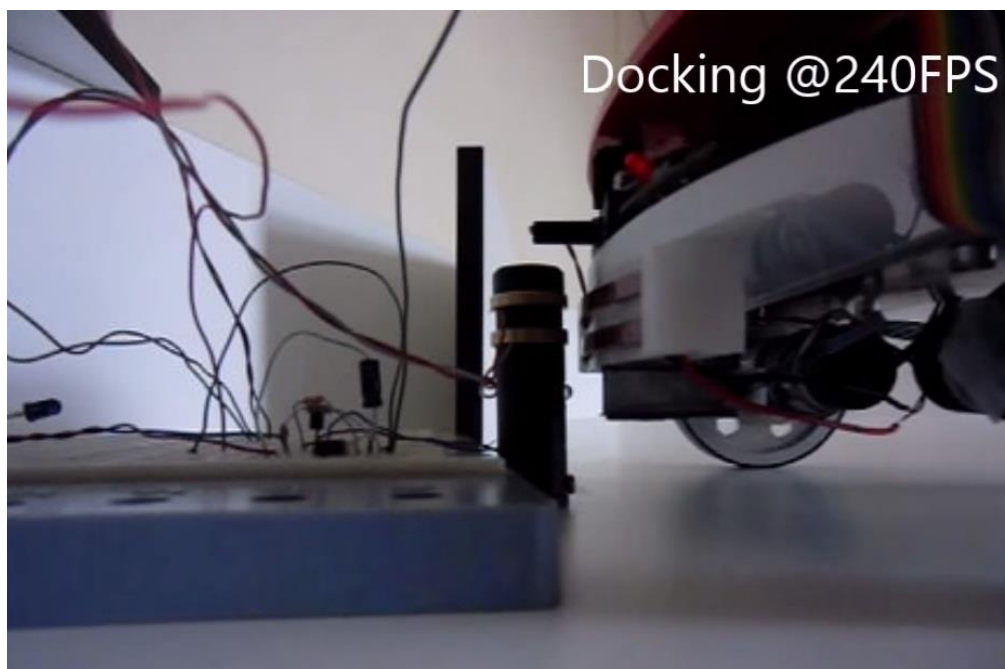
Slika 34 Ponašanje robota za $w = 20$

Iz priloženih mjerenja možemo vidjeti da s povećanjem konstante ω_1 , odziv robota postaje sve više oscilatoran te za $\omega_1 = 20$ imamo gotovo periodičke oscilacije. Isto tako utvrđeno je da zadana translacijska brzina ne utječe na ponašanje kutne brzine, no kod vrlo malih translacijskih gibanja robot ne uspijeva stići to punjača u zadanom vremenu od 60s. Kod prevelikih translacijskih brzina reda $v_t > 50\%$ maksimalne brzine, robot prilikom spajanja može oštetiti stup punjača. Kao optimalne vrijednosti za ω_1 i v_t odabrane su $\omega_1 = 5$ te $v_t = 30$.

6.2. Analiza spajanja robota na stup punjača

Kako bi se provjerila funkcija opisanog uređaja za spajanje robota na punjač izvršena je analiza gibanja robota prilikom sudara sa stupom punjača. Sudar robota i punjača snimljen je brzom kamerom s periodom snimanja $1/240$ s. Promjena udaljenosti između stupa punjača i robota izmjerena je sa fotografija. Analizom je utvrđeno da uređaj za spajanje posjeduje predviđeno amortizirajuće djelovanje te da su bakrene elektrode na uređaju ispravno dimenzionirane. Stup punjača također se pomiče, no uzrok toga je otežana montaža na drvenom poligonu. Slika 37.

prikazuje robota i stup punjača u trenutku spajanja. Snimku spajanja robota na punjač moguće je vidjeti u digitalnom prilogu.

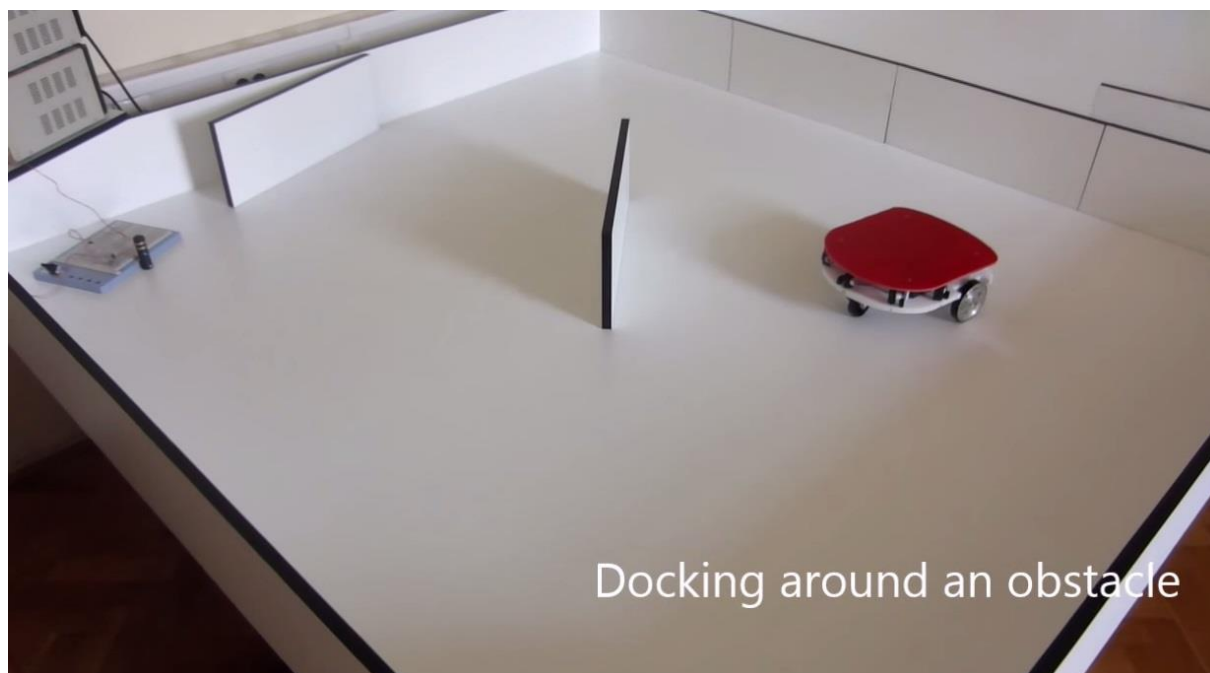


Slika 35 Robot u trenutku spajanja na punjač

6.3. Ispitivanje rada sustava za automatsko punjenje

Za potrebe ispitivanja cjelokupnog sustava za automatsko punjenje nakon podešavanja konstanti regulatora, provedena su dva pokusa. U prvom pokusu pokrenut je program za ispitivanje opisan u odjeljku 3.5. dok se robot nalazio na poligonu bez prepreka dimenzija 2x2m. Nakon pada vrijednosti varijable V_{bat} robot je započeo traženje punjača i uspješno se spojio na stup punjača za 10 sekundi. Drugi pokus odnosio se na ispitivanje rada sustava uz jednu prepreku na poligonu. Pokrenut je isti program na poligonu dimenzija 2x2m s jednom preprekom duljine 50cm na sredini poligona. S obzirom da je prepreka onemogućavala optičku vidljivost između stupa punjača i robota, vrijeme potrebno za spajanja iznosilo je 20 sekundi. Slika 38. prikazuje robota na poligonu za vrijeme ispitivanja spajanja na punjač s jednom

preprekom. Ispitivanje sustava moguće je vidjeti na videu u digitalnom prilogu ili na poveznici [21].



Slika 36 Ispitivanje spajanja na punjač uz prepreke

7. Zaključak

Prema podacima međunarodnog udruženja za robotiku IFR (International Federation of Robotics) [1], potrebe tržišta za mobilnim robotima sve su veće. Roboti u kućanstvu bitno su različiti od industrijskih robota kako po mehaničkoj konstrukciji tako i po načinu vođenja. U svrhu istraživanja ponašanja mobilnih robota, na Fakultetu strojarstva i brodogradnje razvijena je mobilna robotska platforma eMiR. Postoje dvije inačice robota eMiR: prva namjenjena isključivo za komunikaciju s računalom putem Bluetooth veze i druga nadograđena računalom Raspberry Pi. Nadograđena druga inačica podržava komunikaciju putem LAN-a, stoga je robotom moguće upravljati preko VNC sučelja. U svrhu daljnjeg razvoja robota eMiR, konstruiran je i izrađen sustav za automatsko punjenje. Za navođenje robota do punjača koristi se infracrveni signal za navođenje frekvencije 36kHz. Na stupu punjača postavljen je jedan izvor infracrvenog zračenja dometa 2m te širine snopa 20°. Ovakav izvor signala za navođenje pokazao se kao dovoljno dobar za navođenje robota na poligonu dimenzija 2x2m. Prijemnik signala na robotu sadrži dva infracrvena detektora, sjenilo i nosač izrađen aditivnom tehnologijom taložnog očvršćivanja (FDM). Korištenjem samo dva detektora, postoje četiri različita stanja na izlazu prijemnika. Mjerenje kuta pod kojim prijemnik vidi signal za navođenje je relativno male rezolucije, no u ispitivanjima se pokazalo da sustav radi bez prevelikih oscilacija. Uređaj za spajanje robota na punjač je izrađen aditivnom tehnologijom što se pokazalo kao dobar izbor zbog mogućnosti lake ugradnje elastičnih bakrenih elektroda. Analizom spajanja robota na punjač utvrđeno je da postoji amortizirajuće djelovanje elektroda, no isto tako primjećene su i deformacije stupa punjača što bi moglo izazvati probleme u trajnom pogonu. Električni kontakt između elektroda na robotu i elektroda na stupu punjača dovoljno je dobar za relativno male struje punjenja olovnih akumulatora. U slučaju primjene litijskih baterija koje se pune znatno većim strujama, trebalo bi provjeriti zadovoljava li površina kontakta uvjete za prijenos većih struja. Programska podrška razvijena za podešavanje i ispitivanje sustava napisana je u programskom jeziku Python u obliku skripte `emircon_20.py`. Funkcije koje skripta sadrži mogu se pozivati iz ostalih Python skripti što omogućava lak nastavak rada te korištenje na raznim operativnim sustavima i računalima. U daljnjem radu trebalo bi razmotriti postavljanje više izvora infracrvenog zračenja na stup punjača kako bi se povećala širina snopa signala za navođenje. Isto tako mehanička konstrukcija stupa punjača deformira se prilikom spajanja robota, stoga bi je trebalo pojačati. Veći stup punjača omogućio bi smještaj elektroničkog sklopa za generiranje signala za navođenje unutar stupa punjača što

je pogodnije za praktičnu primjenu. Ispitivanje rada sustava trebalo bi nastaviti kako bi se utvrdila pouzdanost sustava te odredila vjerojatnost uspješnog spajanja robota na punjač. U nastavku ispitivanja gibanje robota trebalo bi snimati kamerom sa stropa laboratorija te analizirati putanje kojima se robot giba prema punjaču nakon početka traženja. Ovim ispitivanjem utvrdila bi se efikasnost načina spajanja robota na punjač. Rad cijelog sustava moguće je vidjeti na poveznici: <https://www.youtube.com/watch?v=eeMn97G3OWg>.

8. Literatura

- [1] internet izvor : www.IFR.org
- [2] Amarasinghe et. al. Vision-based hybrid control strategy for autonomus docking of mobile robot Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on
- [3] Jeong M. et. al. Autonomous Docking for an eROSI Robot Based on a Vision System with Points Clustering, Control & Automation, 2007. MED '07. Mediterranean Conference on
- [4] Cooperstock et. al. A neural Network operated vision-guided robot arm for docking and reaching. Diplomski rad University of Toronto
- [5] Cassinis et. al Docking and Charging System for Autonomus Mobile Robots, 1st European conference on mobile robots ECMR'03
- [6] Kim et al. Development of Docking System for Mobile Robots Using Cheap Infrared Sensors, 1st International Conference on Sensing Technology November 21-23, 2005 Palmerston North, New Zealand
- [7] Varun et al. A beacon-based Docking system for an autonomus Mobile Robot Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on; 12/1996
- [8] M. Crneković, D. Zorc, Z. Kunica. Research of mobile robot behavior with eMIR, International Conference on Innovative Technologies, IN-TECH 2012
- [9] M. Crneković, T. Pavlic, M. Lukas, PROGRAMMING LANGUAGE FOR THE eMIR MOBILE ROBOT, Conference on production engineering CIM, Vodice 2015.
- [10] M. Lukas, Projekt diplomskog studija, Nadogradnja mobilnog robota eMiR, FSB 2014.
- [11] M. Lukas, Završni rad, Loptanje mobilnog robota eMiR. FSB 2013.
- [12] internet izvor: http://www.adafruit.com/datasheets/IR333_A_datasheet.pdf
- [13] internet izvor
http://www.mcselec.com/index.php?option=com_content&task=category§ionid=3&id=67&Itemid=41
- [14] internet izvor : <https://www.picotech.com/>
- [15] M. Lukas, Seminarski rad Mikroprocesorsko upravljanje, FSB 2011.
- [16] Internet izvor: <http://www.datasheetarchive.com/TSSOP-38-datasheet.html>
- [17] Decker. K Elementi strojeva, Golden Marketing Tehnička knjiga

[18] T. Pavlic, M. Lukas, M. Crneković, DESIGN AND CONTROL OF ROBOTIC ARM FOR EDUCATIONAL MOBILE ROBOT. Conference on production engineering CIM, Vodice 2015.

[19] J. Kasać, Skripta za kolegij Neizrazito i digitalno upravljanje. FSB 2014.

[20] internet izvor: <https://www.python.org/>

[21] internet izvor: http://www.arttec.net/Solar_Mower/4_Electrical/Battery%20Charging.pdf

9. Prilog – programski kod

Skripta Emircon_20.py

```
def eMiR_init(c):
    global RS232
    import time
    import serial
    import io
    import sys
    RS232 = serial.Serial(c)
    RS232.baudrate = 57600
    RS232.timeout = 1
    RS232.flushInput()
    # Ukljucuje slanje podataka o senzora
    RS232.write('#100100EF/')
    RS232.flushOutput()

def eMiR_citaj():
    global RS232

    i=0
    znak=''
    string=''

    if RS232.inWaiting()>0:
        znak=RS232.readline(1)
        if znak=='*':
            while znak<>'/' and i<32:
                string=string+znak
                znak=RS232.readline(1)
                i=i+1
            if i>32:
                string=''
            else:
                string=string[0:31]
                RS232.flushInput()
        return string
    RS232.flushInput()
    return ""

def eMiR_stop():
    global RS232
    RS232.write('#010000FF/')
    RS232.flushOutput()
    RS232.close()

#Ocitavanje sa senzora
def eMiR_sensor_status(lista):
    global RS232

#Cita sa senzora
    Senzori=''
    Senzori=eMiR_citaj()

    if len(Senzori)==24:
```

```
#sen0
sen0=Senzori[1:3]
sen0=int(sen0,16)
#sen1
sen1=Senzori[3:5]
sen1=int(sen1,16)
#sen2
sen2=Senzori[5:7]
sen2=int(sen2,16)
#sen3
sen3=Senzori[7:9]
sen3=int(sen3,16)
#sen4
sen4=Senzori[9:11]
sen4=int(sen4,16)
#sen5
sen5=Senzori[11:13]
sen5=int(sen5,16)
#sen6
sen6=Senzori[13:15]
sen6=int(sen6,16)
#bat
bat=Senzori[15:17]
bat=int(bat,16)
#vtrans
vtrans=Senzori[17:19]
vtrans=int(vtrans,16)
if vtrans > 127 :
    vtrans = vtrans - 256
#vrot
vrot=Senzori[19:21]
vrot=int(vrot,16)
#if vrot > 127 :
    # vrot = vrot - 256
izlaz=[]
for i in range(0,len(lista)):
    if lista[i] == 0:
        izlaz.append(sen0)
    if lista[i] == 1:
        izlaz.append(sen1)
    if lista[i] == 2:
        izlaz.append(sen2)
    if lista[i] == 3:
        izlaz.append(sen3)
    if lista[i] == 4:
        izlaz.append(sen4)
    if lista[i] == 5:
        izlaz.append(sen5)
    if lista[i] == 6:
        izlaz.append(sen6)
    if lista[i] == 7:
        izlaz.append(vtrans)
    if lista[i] == 8:
        izlaz.append(vrot)
return izlaz
else:
    return []

def eMiR_move(b,h):
```

```
global RS232
paket = ''
Vt=''
if b<0 :
    b = b+256
Vt = hex(b)
d = Vt.split('x')
if b<16 :
    V='0'+d[1]
else:
    V=d[1]

W=''
if h<0 :
    h = h+256
W = hex(h)
d = W.split('x')
if h<16 :
    W1='0'+d[1]
else:
    W1=d[1]
cs=256-1-h-b
if cs<0:
    cs=256+cs
#checksum to hex
CS = hex(cs)
g = CS.split('x')
if cs<16:
    CS='0'+g[1]
else:
    CS=g[1]

paket = '#01'+V+W1+CS+'/'
paket = paket.swapcase()
RS232.write(paket)
RS232.flushOutput()

def Docking(a):
    import time
    logfile=open(a,"w")
    t=0

    for i in range(60):
        status = eMiR_citaj()
        try:
            ulazi = int(status[25:27],16)
                #vtrans
            vtrans=status[15:17]
            vtrans=int(vtrans,16)
            if vtrans > 127 :
                vtrans = vtrans - 256

            vrot=status[17:19]
            vrot=int(vrot,16)
            if vrot > 127 :
                vrot = vrot - 256

            #sen0
            sen5=status[11:13]
            sen5=int(sen5,16)
```

```
except ValueError:
    print 'nema prijema'
print ulazi
if ulazi == 240:
    v=0
    w=20
elif ulazi == 48:
    v=-30
    w=0
elif ulazi == 112:
    v=-30
    w=6
elif ulazi == 176:
    v=-30
    w=-6
eMiR_move(v,w)
t=t+1
vout=str(v)
wout=str(w)
vtrout=str(vtrans)
wrout=str(vrot)
senout=str(sen5)
ulazio=str(ulazi)
time1 = str(t)
logfile.write(ulazio+', '+vout+', '+wout+', '+vtrout+', '+wrout+', '+sen
out+', '+time1+'\n')

    time.sleep(0.2)

logfile.close()
```

