

# Upravljanje mobilnim robotom s diferencijalnim pogonom i ultrazvučnim senzorima udaljenosti

---

**Tatalović, Vedran**

**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:235:327590>

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

*Download date / Datum preuzimanja:* **2024-10-03**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Vedran Tatalović**

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

Mentori:

Prof. dr. sc. Josip Kasać, dipl. ing.

Student:

Vedran Tatalović

Zagreb, 2015.

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

Zahvaljujem se mentoru dr. sc. Josipu Kasaću na korisnim savjetima i diskusijama tokom izrade ovog diplomskog rada.

Također se zahvaljujem svojoj obitelji i prijateljima na potpori tokom izrade rada i tokom cijelog studija.

Vedran Tatalović



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: Vedran Tatalović

Mat. br.: 0035172898

Naslov rada na  
hrvatskom jeziku:

UPRAVLJANJE MOBILNIM ROBOTOM S DIFERENCIJALNIM  
POGONOM I ULTRAZVUČNIM SENZORIMA UDALJENOSTI

Naslov rada na  
engleskom jeziku:

CONTROL OF MOBILE ROBOT WITH DIFFERENTIAL DRIVE  
AND ULTRASONIC DISTANCE SENSORS

Opis zadatka:

Autonomno kretanje u okolini s neuređenim preprekama ključno je za primjenu mobilnih robota u realnim industrijskim okruženjima poput automatiziranih skladišta i fleksibilnih proizvodnih sustava. Bitan aspekt autonomnog kretanja mobilnog robota predstavlja reaktivna navigacija gdje se na temelju trenutnih podataka sa senzora udaljenosti generira referentna trajektorija u realnom vremenu kojom se izbjegava detektirana prepreka.

U radu je potrebno:

- Izraditi fizički model mobilnog robota s diferencijalnim pogonom i pet ultrazvučnih senzora udaljenosti.
- Ugraditi odgovarajuće elektroničke sklopove za prikupljanje i obradu podataka sa senzora, te za upravljanje elektromotorima.
- Izraditi biblioteke programa u programskom jeziku Python za upravljanje elektroničkim sklopovima mobilnog robota.
- Implementirati neizraziti analitički regulator za problem pozicioniranja mobilnog robota, te slijeđenja zadane referentne trajektorije.
- Implementirati metodu potencijalnih polja za sintezu referentne trajektorije mobilnog robota u slučaju reaktivne navigacije u okolini s preprekama.
- Testirati implementirani upravljački sustav modela mobilnog robota na primjerima kretanja mobilnog robota prema zadanom cilju kroz neuređenu okolinu.

Zadatak zadan:

13. studenog 2014.

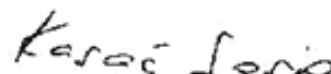
Rok predaje rada:

15. siječnja 2015.

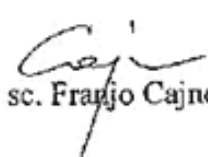
Predvideni datum obrane:

21., 22. i 23. siječnja 2015.

Zadatak zadao:

  
Prof. dr. sc. Josip Kasac

Predsjednik Povjerenstva:

  
Prof. dr. sc. Franjo Cajner

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA .....	V
POPIS OZNAKA .....	VI
SAŽETAK .....	IX
SUMMARY .....	X
1. UVOD .....	1
2. UPRAVLJANJE MOBILNIM ROBOTOM .....	4
2.1. Kinematika mobilnog robota .....	4
2.2 PID regulator .....	6
2.3 Neizraziti sustavi .....	8
2.4 Metoda potencijalnih polja .....	12
2.4.1 Atraktivni i repulzivni potencijal .....	12
2.4.2 Generiranje referentne trajektorije .....	14
3. IZRADA MOBILNOG ROBOTA S DIFERENCIJALNIM POGONOM .....	17
3.1 Trup mobilnog robota .....	18
3.2 Podsustav za kretanje .....	19
3.2.1 Elektromotor .....	19
3.2.2 H-most .....	20
3.2.3 Kotači .....	21
3.3 Senzori .....	22
3.3.1 Inkrementalni optički enkoderi .....	22
3.3.2 Ultrazvučni senzor .....	25
3.4 Podsustav za procesiranje .....	26
3.4.1 Arduino Romeo V2 .....	27
3.4.2 Arduino Due .....	28
3.4.3 Dvosmjerni logički pretvarač .....	30
3.5 Podsustav za komunikaciju .....	32
3.6 Podsustav za napajanje .....	34

---

4. UPRAVLJANJE ELEKTRONIČKIM SKLOPOVIMA.....	35
4.1 Upravljanje enkoderima .....	35
4.2 Upravljanje elektromotorima .....	36
4.3 Komunikacija između mobilnog robota i računala .....	37
4.3.1 Serijska komunikacija - mobilni robot .....	38
4.3.2 Serijska komunikacija - računalo .....	39
4.4 Komunikacija između dva mikrokontrolera.....	40
4.5 Rad ultrazvučnih senzora .....	41
4.5.1 Eksperiment 1 .....	41
4.5.2 Eksperiment 2.....	43
5. REAKTIVNA NAVIGACIJA.....	46
5.1 Testiranje parametara PID regulatora.....	48
5.1.1 Mikrokontroler .....	48
5.1.2 Računalo.....	50
5.1.3 Izvedena testiranja.....	52
5.1.4 Zaključak testiranja .....	61
5.2 Testiranje parametara analitičkog neizrazitog regulatora.....	61
5.2.1 Mikrokontroler .....	61
5.2.2 Računalo.....	63
5.2.3 Izvedena testiranja.....	64
5.2.4 Zaključci testiranja .....	79
5.3 Testiranje parametara metode potencijalnih polja.....	80
5.3.1 Upravljački mikrokontroler .....	81
5.3.2 Računalo.....	82
5.3.3 Izvedeno testiranje.....	83
5.3.4 Zaključci testiranja .....	85
6. ZAKLJUČAK.....	86

**POPIS SLIKA**

Slika 1.	Model mobilnog robota .....	5
Slika 2.	PID regulacijski krug.....	6
Slika 3.	Mobilni robot u prostoru i referentna trajektorija.....	10
Slika 4.	Potencijalno polje .....	13
Slika 5.	Mobilni robot gornji dio .....	17
Slika 6.	Mobilni robot donji dio .....	17
Slika 7.	Donji dio mobilnog robota sa dodatcima .....	18
Slika 8.	DC motori.....	19
Slika 9.	Princip rada H-mosta.....	20
Slika 10.	Dvostruki H-most <i>L298N</i> .....	20
Slika 11.	Kotač enkodera.....	22
Slika 12.	Dvofazni inkrementalni enkoder .....	23
Slika 13.	Jednofazni inkrementalni enkoder.....	24
Slika 14.	<i>HC-SR04</i> .....	25
Slika 15.	<i>Arduino Romeo V2</i> .....	27
Slika 16.	<i>Arduino Due</i> .....	28
Slika 17.	Shema dvosmjernog logičkog pretvarača.....	30
Slika 18.	Prva verzija dvosmjernog logičkog pretvarača .....	31
Slika 19.	Druga verzija dvosmjernog logičkog pretvarača.....	31
Slika 20.	<i>Xbee 1 mW Wire Antenna Series 1</i> .....	33
Slika 21.	<i>Xbee</i> -USB adapter .....	33
Slika 22.	PWM signali.....	37
Slika 23.	Rezultati eksperimenta sa ultrazvučnim sensorima .....	42
Slika 24.	Problem očitavanja stražnjih ultrazvučnih senzora .....	44
Slika 25.	Ispravan rad stražnjih ultrazvučnih senzora .....	45
Slika 26.	Grafičko sučelje za testiranje PID regulatora .....	50
Slika 27.	Uspostava komunikacije.....	51
Slika 28.	Odziv PD regulatora .....	52
Slika 29.	Odziv regulatora sa prefiltracijom ulaznih brzina (Step) .....	54
Slika 30.	Odziv regulatora sa prefiltracijom brzina (Sinusoida) .....	55



---

Slika 31.	Odziv PD regulatora uz uvođenje minimalne promjene ulaznih brzina za restiranje koraka filtracije (Sinusoida) .....	56
Slika 32.	Odziv PD regulatora uz uvođenje minimalne promjene ulaznih brzina za restiranje koraka filtracije (Step) .....	57
Slika 33.	Problem regulacije malih ulaznih brzina .....	58
Slika 34.	Uvođenje minimalnog upravljačkog PWM signala .....	59
Slika 35.	Odziv PD regulatora režim rada enkodera RISING .....	60
Slika 36.	Grafičko sučelje za testiranje analitičkog neizrastog regulatora .....	63
Slika 37.	Rezultati 1. testa .....	66
Slika 38.	Rezultati 2. testa .....	68
Slika 39.	Rezultati 3. testa .....	70
Slika 40.	Rezultati 4. testa .....	72
Slika 41.	Rezultati 5. testa .....	74
Slika 42.	Rezultati 6. testa .....	76
Slika 43.	Rezultati 7. testa .....	78
Slika 44.	Rezultati 8. testa .....	79
Slika 45.	Grafičko sučelje za testiranje metode potencijalnih polja .....	82
Slika 46.	Rezultat testiranja metode potencijalnih polja .....	84

**POPIS TABLICA**

Tablica 1. Osnovni dijelovi dvostrukog H-mosta <i>L298N</i> .....	21
Tablica 2. <i>Pin</i> -ovi jednofaznog inkrementalnog enkodera.....	24
Tablica 3. <i>Pin</i> -ovi ultrazvučnog senzora <i>HC-SR04</i> .....	25
Tablica 4. Usporedba <i>Arduino Due</i> i <i>Arduino Romeo V2</i> .....	28
Tablica 5. Spajanje mikrokontrolera i uređaja preko dvosmjernog pretvarača .....	32
Tablica 6. <i>Pin</i> -ovi <i>Xbee</i> -USB adaptera .....	33
Tablica 7. Popis <i>pin</i> -ova na upravljačkom mikrokontroleru .....	47
Tablica 8. Popis <i>pin</i> -ova na mikrokontroleru sa ultrazvučnim senzrima .....	47

## POPIS OZNAKA

Oznaka	Jedinica	Opis
$\dot{x}$	m/s	Linearna brzina u smjeru osi X
$\dot{y}$	m/s	Linearna brzina u smjeru osi Y
$\dot{\theta}$	rad/s	Promjena kuta zakreta
$x$	M	Položaj robota u prostoru u smjeru osi X
$y$	M	Položaj robota u prostoru u smjeru osi Y
$\theta$	Rad	Kut zakreta robota oko osi X
$x_a$	M	Položaj cilja u prostoru u smjeru osi X
$y_a$	M	Položaj cilja u prostoru u smjeru osi Y
$x_{oL}, x_{oM}, x_{oR}$	M	Položaj prepreke u prostoru u smjeru osi X (pojedini senzor)
$y_{oL}, y_{oM}, y_{oR}$	M	Položaj prepreke u prostoru u smjeru osi Y (pojedini senzor)
$u_1$	m/s	Upravljačka linearna brzina
$u_2$	rad/s	Upravljačka kutna brzina
$u_R$	m/s	Zadana brzina desnog kotača
$u_L$	m/s	Zadana brzina lijevog kotača
$u_{fL}$	m/s	Filtrirana zadana brzina lijevog kotača
$u_{fR}$	m/s	Filtrirana zadana brzina desnog kotača
$v$	m/s	Linearna brzina mobilnog robota
$\omega$	rad/s	Kutna brzina mobilnog robota
$v_L$	m/s	Brzina lijevog kotača
$v_R$	m/s	Brzina desnog kotača
$v_{max}$	m/s	Maksimalna dopuštena linearna brzina
$\omega_{max}$	rad/s	Maksimalna dopuštena kutna brzina
$L$	M	Udajenost između kotača
$D_k$	M	Promjer kotača
$N_L, N_R$		Broj promjene stanja enkodera (lijevi i desni)
$rez$		Rezolucija enkodera
$t_{eL}, t_{eR}$	S	Vrijeme zadnje zabilježene promjene na enkoderu

$t_{epL}, t_{epR}$	S	Vrijeme zadnje zabilježene promjene na enkoderu (prethodni korak programa)
$e$	m/s	Regulacijsko odstupanje
$P$		Proporcionalni član PID regulatora
$K_p$		Pojačanje P člana
$I$		Integracijski član PID regulatora
$K_i$		Pojačanje I člana
$T_i$		Vremenska konstanta I člana
$D$		Derivacijski član PID regulatora
$K_d$		Pojačanje D člana
$T_d$		Vremenska konstanta D člana
$p$		Upravljački PWM signal
$t$	S	Vrijeme
$T$	S	Vremenski period
$N$		Korak
$\varepsilon$		Mala konstanta
$r_m$	M	Položaj mobilnog robota u prostoru
$r$	M	Položaj mobilnog robota u prostoru u odnosu na referentnu trajektoriju
$r_r$	M	Položaj referentne trajektorije u prostoru
$r_e$	M	Orjentacija mobilnog robota
$r_a$	M	Položaj cilja u prostoru
$V_p$		Vektorski produkt
$V_{pn}$		Normirani vektorski produkt
$S_{pn}$		Normirani skalarni produkt
$k_1, k_2, k_3$		Faktori pojačanja analitičkog neizrazitog regulatora
$V$	m <sup>2</sup> /s	Potencijalno polje
$V_a$	m <sup>2</sup> /s	Atraktivni potencijal
$V_r$	m <sup>2</sup> /s	Repulzivni potencijal
$V_{rL}, V_{rM}, V_{rR}$	m <sup>2</sup> /s	Repulzivni potencijal na pojedinom senzoru
$\dot{x}_r$	m/s	Promjena referentne trajektorije u smjeru osi X
$\dot{y}_r$	m/s	Promjena referentne trajektorije u smjeru osi Y
$k_a$		Težinski faktor za atraktivni potencijal
$k_r$		Težinski faktor za repulzivni potencijal

---

$B$	M	Radijus u kojem djeluje repulzivni potencijal
$r_{rob}$	M	Polumjer robota
$N_{tol}$	M	Dopuštena udaljenost mobilnog robota od prepreke
$\mu_{rot}$		Omjer jakosti između rotacijskog i repulzivnog potencijala
$sl$		Težinski faktor u modelu sljeđenja zida
$L_S$	M	Udaljenost prepreke od ultrazvučnog senzora
$L_{SL}, L_{SM}, L_{SR}$	M	Udaljenost prepreke od pojedinog senzora
$t_s$	S	Vrijeme potrebno da se zvuk vrati na senzor
$c$	m/s	Brzina zvuka

## SAŽETAK

U ovom radu je proučavana problematika reaktivne navigacije mobilnog robota u nepoznatom prostoru preko metode potencijalnih polja. U tu svrhu je izrađen mobilni robot na diferencijalni pogon sa 5 ultrazvučnih senzora za određivanje udaljenosti mobilnog robota od prepreke. Izrađeni mobilni robot je opisan, zajedno sa svim svojim važnim dijelovima. Uz njihov opis i način rada, navedeni su i razlozi odabira pojedinih dijelova, kao i načini poboljšanja samog mobilnog robota. Upravljanje mobilnim robotom je izvedeno preko dvaju mikrokontrolera u kojima je jedan korišten za prikupljanje podataka sa ultrazvučnih senzora, dok drugi mikrokontroler upravlja samim mobilnim robotom. Opisan je način na koji mikrokontroler upravlja pojedinim upravljačkim sklopovima mobilnog robota, kao i prikupljanje podataka sa senzora. Izvršen je eksperiment za analizu rada pet ultrazvučnih senzora i načina na koji mikrokontroleri međusobno komuniciraju. Za ostvarenje reaktivne navigacije izrađena su tri programa koji analiziraju rad važnijih upravljačkih funkcija. Prvi program je izrađen za namještanje parametara PID regulatora brzine vrtnje kotača mobilnog robota za ostvarenje željene linearne i kutne brzine. Drugi program omogućava namještanje parametara analitičkog neizrazitog regulatora i omogućava mobilnom robotu da slijedi zadanu trajektoriju. Zadnji izrađeni program omogućava reaktivnu navigaciju mobilnog robota preko metode potencijalnih polja, kao i namještanje njenih parametara. Na sva tri programa su izvedena testiranja na osnovu kojih su doneseni neki zaključci o samom radu mobilnog robota. Mobilni robot je programiran u sučelju *Arduino IDE 1.5.7*, dok je program na računalu izrađen u programskom jeziku *Python 2.6*.

Ključne riječi: Reaktivna navigacija, Metoda potencijalnih polja, PID regulator, Analitički neizraziti regulator, Mobilni robot, Diferencijalni pogon, Ultrazvučni senzor, Arduino, Python

## SUMMARY

The main task of this work was to program the reactive navigation of a mobile robot based on potential field method, as well as creation of programs for testing PID regulator and analytical fuzzy regulator. For the purposes of testing these programs, a differential-drive mobile robot with 5 ultrasonic range finders for determining the distance from the obstacle was constructed. Mobile robot is controlled by two microcontrollers, one of which was used for collection of ultrasonic range finders data, while the second one controls the robot itself. The method of controlling the individual parts of mobile robot, and data collection from sensors, are explained. Experiment was made to check the sensor way of working, as well as their communication. In order to achieve reactive navigation, three programs were made, each analyzing its particular control function. First program sets PID regulator parameters of left and right wheels speeds, in order to achieve the needed translational and rotational speed. Second program allows setting of analytical fuzzy regulator parameters, and enables the mobile robot to follow a set trajectory. Last created program enables the reactive navigation of mobile robot using the potential field method, as well as setting of its parameters. All three programs were tested and their results analyzed in order to draw conclusions about the way that mobile robot works. Mobile robot was programmed in *Arduino IDE 1.5.7*, while the computer program was coded in *Python 2.6* programming language.

Key words: Reactive navigation, Potential field method, PID regulator, Analytical fuzzy regulator, Mobile robot, Differential drive, Ultrasonic range finder , Arduino, Python

## 1. UVOD

Mobilne robote možemo definirati kao mobilan i manipulativan fizički sustav koji se autonomno giba kroz nestrukturirani prostor, ostvarujući pritom interakciju s ljudskim bićima ili autonomno obavljajući neki posao umjesto njih [1].

Stroj možemo smatrati mobilnim ako ispunjava sljedeće uvjete [2]:

- Mobilnost: Robot mora imati sposobnost slobodnog gibanja po prostoru (ne smije imati stacionarnu točku)
- Autonomnost: Ljudska interakcija sa robotom je minimalna i ograničena na zadatak. U ovom radu to je zadavanje željenog cilja
- Percepciju: Kako bi se uspješno mogao snalaziti u okolini, mobilni robot mora posjedovati senzore pomoću kojih zapaža promjene u njoj. Mobilni robot izrađen u sklopu ovog rada koristi ultrazvučne senzore koji služe za određivanje udaljenosti od prepreka na koje može naići tokom dolaska do cilja.

Mobilni roboti se dijele na više načina. Najpopularnije podjele su prema vrsti medija po kojem se gibaju (kopno, voda, zrak) i po načinu na koji ostvaruju svoje gibanje (kotači, noge, zmijoliki, itd) [3].

U sklopu ovog rada je izrađen mobilni robot sa diferencijalnim pogonom. Takav mobilni robot se sastoji od dva kotača pokretanih uz pomoć elektromotora i slobodnog kotača koji služi za odražavanje ravnoteže mobilnog robota. Prednost mobilnog robota s diferencijalnim pogonom je u jednostavnom upravljanju i načinu gibanja koji mu omogućava zakretanje u prostoru. Glavna mana ovakvih mobilnih robota je u osjetljivosti slobodnog kotača, zbog čega kod nepravilnih prepreka može doći do izbacivanja mobilnog robota iz smjera putanje. Također, pošto se gibanje ostvaruje samo preko dva kotača, za ostvarivanje stabilnog gibanja (npr. ravno gibanje) potrebna je dobra regulacija brzine vrtnje elektromotora.

Glavni cilj ovog rada je izraditi program koji bi mogao omogućiti izrađenom mobilnom robotu da dođe do zadanog cilja. Mobilni robot se giba po nepoznatom prostoru, pa mora imati sposobnost izbjegavanja prepreka koje će mu se naći na putu, tj. mora imati sposobnost promjene svoje putanje ovisno o podacima dobivenim sa 5 ultrazvučnih senzora. Takav način rada mobilnog robota se zove reaktivna navigacija, a on je ključan za realnu primjenu mobilnih robota.



Sam način dolaska mobilnog robota do cilja reaktivnom navigacijom može se opisati u tri koraka:

- Mobilni robot generira referentnu trajektoriju ovisno o položaju cilja i prepreka u prostoru. To je u sklopu ovog rada izvedeno metodom potencijalnih polja.
- Mobilni robot određuje linearnu i kutnu brzinu pomoću koje može doći do zadane trajektorije, preko analitičkog neizrazitog regulatora.
- Mobilni robot pretvara linearnu i kutnu brzinu u brzinu vrtnje lijevog i desnog kotača. Kako bi se osiguralo da se kotači gibaju zadanom brzinom potrebno ih je regulirati. To je osigurano PD regulatorom.

Za uspješno ostvarenje reaktivne navigacije potrebno je da ta tri koraka rade optimalno. S time na umu, izrađena su tri programa koja služe za analizu rada pojedinog od ta tri koraka i namještanje njihovih parametara. Valja napomenuti da se programi trebaju optimizirati obrnuto od koraka u kojima se pojavljuju u reaktivnoj navigaciji. Tako je prvo potrebno optimizirati rad PD regulatora i takav program ugraditi u program koji testira rad analitičkog neizrazitog regulatora, itd. Na sva tri programa su provedena testiranja i analizirani rezultati.

#### Pregled po poglavljima:

U drugom poglavlju prikazana je teorija na osnovu koje je izrađen program. Prvo je rečeno nešto o samoj kinematici mobilnog robota sa diferencijalnim pogonom. Zatim je opisana sinteza PID regulatora za brzinu vrtnje kotača. Nakon toga je rečeno nešto o analitičkom neizrazitom regulatoru na osnovu kojeg mobilni robot prati referentnu trajektoriju. I za kraj, prikazana je metoda potencijalnih polja na osnovu koje mobilni robot generira referentnu trajektoriju.

U trećem poglavlju opisan je izrađeni mobilni robot i njegove komponente podjeljene na podsustave. Podsustavi koji su detaljnije opisani su trup mobilnog robota, podsustav za kretanje, razumjevanje, komunikaciju i napajanje.

Četvrto poglavlje se bavi upravljanjem nekih važnijih elektronskih uređaja korištenih u mobilnom robotu. Prvo je opisan način na koji mjere enkoderi i kako se preko njih određuje brzina vrtnje motora. Zatim je naveden način kako mikrokontroler namješta napon na elektromotorima. Također je objašnjeno kako mobilni robot komunicira sa računalom, te kako dva mikrokontrolera na mobilnom robotu međusobno komuniciraju. Za kraj je objašnjen rad ultrazvučnih senzora te je proveden eksperiment za proučavanje rada istih.

U petom poglavlju su opisani programi za namještanje parametara za ostvarenje reaktivne navigacije mobilnog robota, te su opisani rezultati ispitivanja rada pojedog programa. Prvo je opisan program za namještanje parametara PID regulatora (tj. odabranog PD regulatora). Također su predložena neka poboljšanja za rad samog regulatora. Nakon toga je opisan program za namještanje parametara analitičkog regulatora, te su opisani rezultati ispitivanja u kojima mobilni robot pokušava doći do zadanog cilja. Na kraju je opisan program za namještanje parametara metode potencijalnih polja i provedena testiranja rada samog programa.

## 2. UPRAVLJANJE MOBILNIM ROBOTOM

Tema ovog rada je reaktivna navigacija diferencijalnog mobilnog robota u nepoznatoj okolini.

Za uspješno ostvarenje reaktivne navigacije potrebno je izvršiti korake:

- Odrediti kinematički model mobilnog robota
- Napraviti sintezu regulatora brzine vrtnje kotača (PID regulator)
- Izraditi analitički neizraziti regulator za sljeđenje referentne trajektorije
- Izraditi algoritam za generiranje referentne trajektorije pomoću metode potencijalnih polja.

Za poboljšanu izvedbu reaktivne navigacije potrebno je poznavati i dinamiku mobilnog robota, no ona nije razmatrana u sklopu ovog rada.

Teoretska osnova za ostvarenje navedenih koraka je izrađena pomoću [4] i [5].

### 2.1. Kinematika mobilnog robota

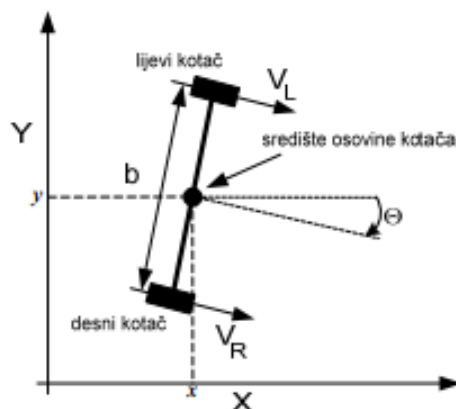
Gibanje mobilnog robota u prostoru (Kartezijev koordinatni sustav) može se opisati preko sljedećih diferencijalnih jednažbi:

$$\dot{x} = u_1 \cos \theta \quad (1)$$

$$\dot{y} = u_1 \sin \theta \quad (2)$$

$$\dot{\theta} = u_2 \quad (3)$$

gdje je  $u_1$  linearna brzina,  $u_2$  kutna brzina, a  $\theta$  je kut zakreta mobilnog robota u odnosu na os  $x$  [Slika 1].



Slika 1. Model mobilnog robota

Ovakva kinematička struktura omogućava mobilnom robotu okretanje na mjestu, što je vrlo korisno kod prolaska mobilnog robota kroz uske prostore. Takva struktura omogućava da se mobilni robot reprezentira poput kružnice koja opisuje isti, što je vrlo korisno za potrebne reaktivne navigacije.

U simulacijama ovakav opis kinematike mobilnog robota je dovoljan za daljnju izradu programa. No, u praktičnoj izvedbi mora se uzeti u obzir da se upravljački sustav mobilnog robota sastoji od dva elektromotora kojima se upravlja odvojeno.

Mobilni robot ostvaruje kutnu i linearnu brzinu preko rada lijevog i desnog kotača čiji se odnos može prikazati jednadžbama:

$$u_1 = \frac{u_R + u_L}{2} \quad (4)$$

$$u_2 = \frac{u_R - u_L}{L} \quad (5)$$

gdje  $L$  predstavlja razmak između lijevog i desnog kotača.

U radu program će prvo odrediti  $u_1$  i  $u_2$ , pa iz njih izraziti željene brzine lijevog  $u_L$  i desnog  $u_R$  kotača.

$$u_L = \frac{2u_1 - u_2L}{2} \quad (6)$$

$$u_R = u_2L + u_L \quad (7)$$

Također valja napomenuti kako su brzine  $u_L$  i  $u_R$  željene brzine vrtnje kotača, no one ne moraju odgovarati realnim brzinama kotača. Kako bi se osiguralo da željena brzina odgovara realnoj brzini kotača potrebno je izraditi regulator za lijevi i desni kotač. U ovom radu je

korišten PD regulator, no prije nego što je takva vrsta regulatora bila odabrana, provedeno je testiranje rada na PID regulatoru.

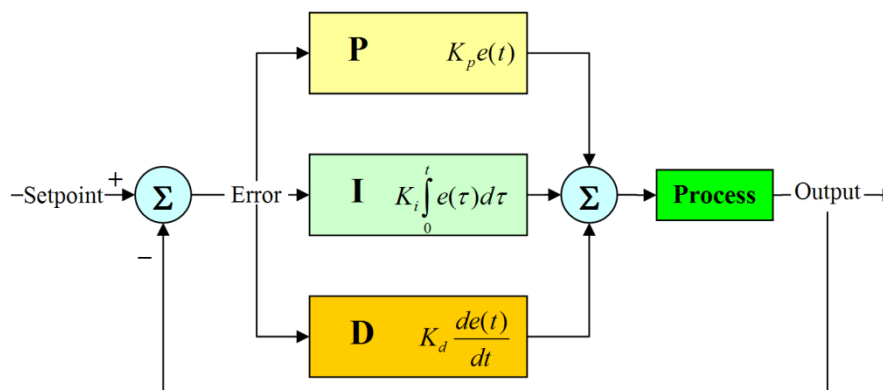
## 2.2 PID regulator

Za uspješno upravljanje mobilnim robotom potrebno je osigurati da kotači mobilnog robota mogu ostvariti i održavati željenu brzinu vrtnje. U idealnom slučaju mobilni robot bi u trenutku zadavanja željene brzine kotača ( $u_L$  ili  $u_R$ ) odmah ostvario traženu brzinu. U praksi realna brzina  $v$  neće uvijek odgovarati traženoj brzini. Razliku između željene brzine i izmjerene brzine nazivamo regulacijsko odstupanje  $e$ .

$$e(t) = u(t) - v(t) \quad (8)$$

Svrha regulacije je osigurati da regulacijsko odstupanje bude 0 ili što bliže nuli. Jedna od najzastupljenijih metoda regulacije je uz pomoć PID regulatora.

PID regulator je vrsta zatvorenog regulacijskog kruga koji računa pogrešku između izmjerene vrijednosti i određuje vrijednost i korigira upravljačku varijablu. [Slika 2] prikazuje klasični krug sa PID regulatorom.



Slika 2. PID regulacijski krug

Djelovanje PID regulatora se može podijeliti na  $P$  (Proporcionalno),  $I$  (Integracijsko) i  $D$  (Derivacijsko). Svaki od tih parametara drugačije utječe na odziv sustava.

- $P$  djelovanje: Proporcionalno djelovanje dodaje linearni odziv na pogrešku.

Proporcionalno djelovanje može se izraziti jednadžbom:

$$P = K_p e \quad (9)$$

$P$  član može stvoriti oscilacije u sustavu (nestabilan sustav), dok premali član može dovesti do presporog odziva sustava kod velikih pogrešaka (npr. kod promjene vrijednosti stanja  $i$  u početnom trenutku).

$P$  se koristi za stabilizaciju sustava prvog reda (kao npr. elektromotor) gdje nije bitna točnost sustava (zbog potencijalne pogreške u ustaljenom stanju).

- $I$  djelovanje: Integracijski član definiran je jednažbom:

$$I = K_i \int_0^t e(\tau) d\tau \quad (10)$$

gdje je  $K_i$  najčešće jednak:

$$K_i = \frac{K_p}{T_i} \quad (11)$$

Integracijsko djelovanje uklanja regulacijsko odstupanje, no zato djeluje destabilizirajuće na sustav i povećava početno odskakivanje od željenog stanja, čime usporava sustav (nepovoljno za slučaj mobilnog robota).

- $D$  djelovanje: Derivacijski član je definiran jednažbom:

$$D = K_d \frac{d}{dt} e(t) \quad (12)$$

Gdje je  $K_d$  najčešće jednak:

$$K_d = K_p T_d \quad (13)$$

$D$  član ubrzava vrijeme stabilizacije sustava, no uvodi i šum u sustav, pa je često potrebno implementirati niskopropusne filtre kako ne bi došlo do visokofrekvencijskog pojačanja.

Upravljačka varijabla  $p(t)$  PID regulatora dobiva se zbrajanjem tih triju djelovanja.

$$p(t) = K_p e(t) + K_p T_d \frac{d}{dt} e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (14)$$

Vrlo važan dio izrade PID regulatora za sustav je određivanje parametara koji se moraju optimizirati prema zahtjevima sustava.

Za podešavanje parametara postoji više pristupa:

- Eksperimentalno,
- Zeiger-Nichols metoda,
- Analitičko podešavanje,
- Podešavanje na temelju simulacije,
- *Autotuning*,
- *Selftuning*.

U ovom radu je izrađeno sučelje za eksperimentalno podešavanje parametara.

### PID regulator mobilni robot

PID regulator kod mobilnog robota mora osigurati željenu brzinu vrtnje lijevog i desnog kotača. Kod podešavanja parametara potrebno je voditi računa o nekoliko činjenica:

- Kako bi se mobilni robot mogao stabilno gibati, i lijevi i desni kotač moraju doći do željenih brzina u istom trenutku (vrijeme stabilizacije sustava mora biti jednako).
- Željene brzine vrtnje mobilnog robota će se konstantno mijenjati, pa je zato potreban brz odziv regulatora.
- Ne smiju se pojavljivati oscilacije koje bi mogle dodatno destabilizirati mobilni robot.
- Ne smije postojati pretjerano odskakivanje (*overshoot*) sustava, osim u slučaju pokretanja motora (potrebno je poslati dovoljno velik napon na motor za njegovo pokretanje).

Iako se u regulatoru računa razlika između željene brzine i dobivene brzine (8), od velike je važnosti razumjeti da sama upravljačka varijabla koja regulira brzinu jest PWM signal, pomoću kojeg se postavlja napon na elektromotore. Raspon PWM signala se kreće od 0 do 255. Više o tome će biti rečeno kasnije.

## **2.3 Neizraziti sustavi**

Neizraziti sustav predstavlja skupinu sustava (neizrazito upravljanje, neizraziti ekspertni sustavi, itd.) koji koriste neizrazitu logiku i pripadajući matematiku.

Osnovu svih neizrazitih sustava čini [4]:

- Omekšavanje ili fazifikacija: Proces transformiranja ulaznih izrazitih vrijednosti u stupnjeve pripadnosti ulaznih neizrazitih skupova. Taj postupak se može podijeliti na tri koraka:

1. Određivanje ulaznih i izlaznih varijabli,
  2. Određivanje područja definicije navedenih varijabli,
  3. Određivanje broja i oblika funkcija pripadnosti.
- Inferencija ili proces odlučivanja: Proces transformiranja ulaznih neizrazitih skupova u neizrazite izlazne skupove. Postupak se sastoji od određivanja baze pravila ponašanja.
  - Izoštavanje ili defazifikacija: Proces transformiranja neizrazitih izlaznih skupova u izrazite izlazne vrijednosti. Taj proces daje vrijednosti koje se šalju na aktuator.

Glavna ideja neizrazitih sustava je izrada baze pravila ponašanja pomoću kojih će neki sustav pokušavati imitirati rad čovjeka, eksperta.

### Neizraziti analitički regulator

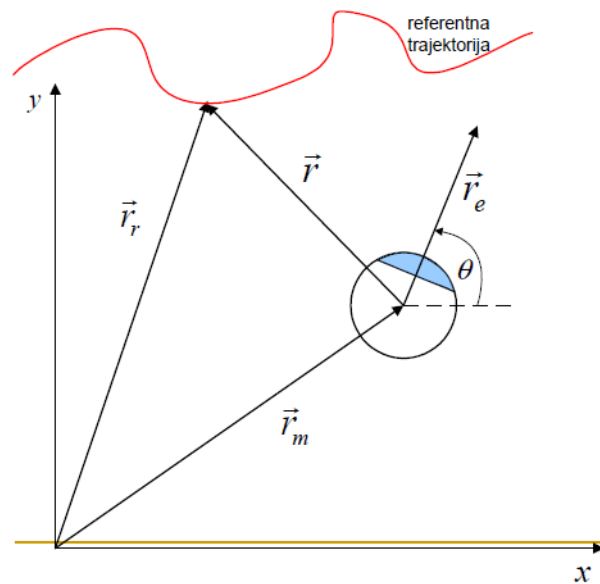
Glavni nedostatak neizrazitih sustava je u bazi pravila ponašanja koja eksponencijalno raste sa brojem ulazno/izlaznih varijabli. Baza pravila ponašanja se sastoji od niza *IF-THEN* logike koja sama po sebi zahtjeva relativno dugo vrijeme procesiranja, pa kod velike baze podataka dolazi do velikog usporenja rada sustava. Takav način rada zahtjeva snažan procesor i nije dobar kod sustava koji je sklon brzim promjenama stanja (kao što je slučaj mobilnog robota).

Da bi se taj problem izbjegao kod upravljanja mobilnim robotom može se koristiti analitički neizraziti regulator. Analitički neizraziti regulator ne posjeduje bazu podataka već su pravila ponašanja opisana preko analitičke formule koja daje izlazne varijable regulatora.

Kod mobilnog robota u radu neizraziti regulator se koristi kod navigacije mobilnog robota tako da pokušava slijediti generiranu trajektoriju nastalu metodom potencijalnih polja, o čemu će više biti rečeno kasnije.

Željene upravljačke veličine (linearna i kutna brzina) se dobivaju preko norme vektora i vektorskog produkta. Za bolje pojašnjenje ideje analitičkog neizrazitog regulatora potrebno je prikazati mobilni robot u Kartezijevom koordinatnom sustavu i cilj (ili referentnu trajektoriju) [Slika 3].





**Slika 3. Mobilni robot u prostoru i referentna trajektorija**

$\vec{r}_m$  je pozicija mobilnog robota u prostoru,  $\vec{r}_r$  je referentna pozicija cilja,  $\vec{r}$  je udaljenost robota od cilja dok je  $\vec{r}_e$  orijentacija mobilnog robota u prostoru.

$$\vec{r}_m = x\vec{i} + y\vec{j} \quad (15)$$

$$\vec{r}_r = x_r(t)\vec{i} + y_r(t)\vec{j} \quad (16)$$

$$\vec{r} = \vec{r}_r - \vec{r}_m = (x_r - x)\vec{i} + (y_r - y)\vec{j} \quad (17)$$

$$\vec{r}_e = \cos \theta \vec{i} + \sin \theta \vec{j} \quad (18)$$

Za izradu analitičkog neizrazitog regulatora potrebno nam je poznavanje norme vektora robota u prostoru  $\|\vec{r}\|$  (19), kao i vektorski produkt pozicije robota u prostoru i njene orijentacije  $V_p$  (20).

$$\|\vec{r}\| = \sqrt{(x_r - x)^2 + (y_r - y)^2} \quad (19)$$

$$V_p = \vec{r}_e \times \vec{r} = (y_r - y) \cos \theta - (x_r - x) \sin \theta \quad (20)$$

Vektorski produkt predstavlja orijentaciju mobilnog robota u odnosu na referentnu trajektoriju. Kako bi vektorski produkt mogli koristiti za izračun kutne i linearne brzine,

vektorski produkt (20) je potrebno normirati, čime dobivamo normirani vektorski produkt  $V_{pn}$ .

$$V_{pn} = \frac{\vec{r}_e \times \vec{r}}{\|\vec{r}\| + \varepsilon} v(t) \quad (21)$$

Gdje je  $\varepsilon$  neka mala konstantna vrijednost (0.001 ili slično) koja sprječava djeljenje sa nulom. Roatacijsku brzinu  $u_2$  dobivamo iz analitičke formule koja je u ovisnosti sa vektorskim produktom  $V_{pn}$ .

$$u_2 = \omega_{max} \tanh(k_1 V_{pn}) \quad (22)$$

Za izračun linearne brzine potrebno je poznavati kako mobilni robot dobiva informacije o svojoj okolini. Ako mobilni robot ima senzore samo na prednjem kraju, poželjno je limitirati linearnu brzinu samo na pozitivne vrijednosti. U tom slučaju jednažba za izračun glasi:

$$u_1 = v_{max} \|\vec{r}\| e^{-k_3 |V_{pn}|} \quad (23)$$

gdje je  $v_{max}$  maksimalna dopuštena linearna brzina. Ograničavanje brzine je potrebno kako mobilni robot ne bi dobivao vrijednosti koje ne može ostvariti, čime bi se sustav učinio nestabilnim.

U slučaju da imamo senzore i na stražnjem dijelu linearnu brzinu određujemo iz jednažbe:

$$u_1 = v_{max} \tanh(k_2 S_{pn}) e^{-k_3 |V_{pn}|} \quad (24)$$

gdje je  $S_{pn}$  normirani skalarni produkt.

$$S_{pn} = \frac{\vec{r}_e \cdot \vec{r}}{\|\vec{r}\| + \varepsilon} = \frac{(x_r - x) \cos \theta - (y_r - y) \sin \theta}{\|\vec{r}\| + \varepsilon} \quad (25)$$

Generalna ideja analitičkog neizrazitog regulatora je da se mobilni robot prvo pravilno orijentira u odnosu na referentnu trajektoriju (dominantnija  $u_2$ ), a zatim krene prema njoj (dominantnija  $u_1$ ).

## 2.4 Metoda potencijalnih polja

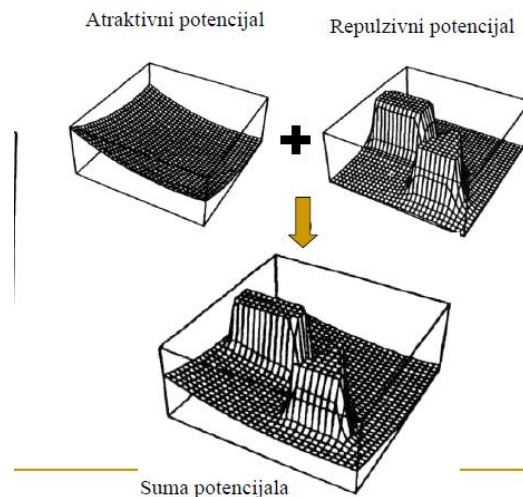
Metoda potencijalnih polja je jedan od načina generiranja referentne trajektorije (putanje) u navigaciji mobilnog robota u okolini. Iako se metoda može koristiti i u poznatoj okolini, njena glavna snaga je u navigaciji u okolini koja je nepoznata ili u kojoj je mobilni robot u interakciji sa ljudima i drugim robotima. Ta metoda se sastoji od dodjeljivanja atraktivnog potencijala cilju kojem mobilni robot teži i repulzivnog potencijala prepreci. Glavna ideja je u tome da cilj generira polje koje privlači mobilni robot prema sebi, dok prepreke generiraju polje koje odbija mobilni robot od sebe. U slučaju u kojem se robot giba u nepoznatom prostoru referentna trajektorija se mijenja ovisno o novim informacijama.

### 2.4.1 Atraktivni i repulzivni potencijal

U metodi potencijalnih polja mobilni robot se promatra poput kružnice koju opisuje površina mobilnog robota. Središte te kružnice predstavlja trenutnu poziciju mobilnog robota u prostoru. Kod praktične izvedbe od velike je važnosti da se centar kružnice nalazi na mjestu gdje su elektromotori kako ne bi bila potrebna dodatna pretvorba koordinata. Na točku središta kružnice djeluje potencijalno polje  $V$  koje je jednako sumi atraktivnog  $V_a$  i repulzivnog  $V_r$  potencijala.

$$V = V_a + V_r \quad (26)$$

Kada bismo potencijalna polja htjeli ilustrirati u  $3D$  prostoru, za atraktivni potencijal bi mogli reći da je poput nizbrdice na čijem najnižem dijelu je cilj. Repulzivni potencijal bi se mogao promatrati poput strmine koja otežava mobilnom robotu gibanje po njoj. Zbrajanjem ta dva potencijala dobiva se potencijalno polje prostora. Za potencijalno polje bi se, dakle, moglo reći da predstavlja prikaz trodimenzionalnog prostora kako ga je mobilni robot izgradio. [Slika 4] prikazuje potencijalno polje.



**Slika 4. Potencijalno polje**

Mobilni robot se tokom rada giba po svojem izgrađenom prostoru tako da se giba po putanji sa najnižom mogućom sumom potencijala.

Atraktivni potencijal se može računati na više načina, no u sklopu ovog rada se računa preko formule:

$$V_a = k_a \|r_r - r_a\| \quad (27)$$

gdje je  $k_a$  pozitivni skalirajući faktor čija veličina mora biti veća od  $v_{max}$ .

Repulzivna se također može izračunati na više načina no u ovom radu je ona jednaka:

$$V_r = (V_{rL} + V_{rM} + V_{rR})/3 \quad (28)$$

gdje je  $V_{ri}$  repulzivna sila izmjerena na sensorima, čija formula glasi:

$$V_{ri} = \begin{cases} k_r \|r_r - r_{oi}\| + V_{rmi} & \text{ako } \|r_r - r_{oi}\| \leq B \\ 0 & \text{ako } \|r_r - r_{oi}\| > B \end{cases} \quad (29)$$

pri čemu je  $k_r$  skalirajući faktor za repulzivnu silu koji mora biti veći od  $v_{max}$ , a  $B$  odabrani radijus u kojem djeluje repulzivna sila. Odabrani radijus je jednak radijusu mobilnog robota i danoj toleranciji (30). U ovom radu je pretpostavljeno da je  $r_{rob}$  jednak 0,154 (iako to nije istina, radi olakšanja izračuna se stavlja da je centar mase jednak udaljenosti mobilnog robota od prednjih senzora).

$$B = (r_{rob} + N_{tol}) \quad (30)$$

Pošto se u ovom radu trajektorija generira za realan mobilan robot čiju dinamiku ne znamo, važno je da je  $N_{tol} > 0.1$  m.

## 2.4.2 Generiranje referentne trajektorije

Pošto se mobilni robot giba po  $x$ - $y$  koordinatnom prostoru, atraktivni i repulzivni potencijal se moraju vektorski rastaviti u  $x$  i  $y$  koordinate.

Tako se promjena referentne trajektorija u metodi potencijalnih polja računa preko jednadžbi:

$$\dot{x}_r = -\frac{\partial V_a}{\partial x_r} - \frac{1}{3} \left( \frac{\partial V_{rL}}{\partial x_r} + \frac{\partial V_{rM}}{\partial x_r} + \frac{\partial V_{rR}}{\partial x_r} \right) \quad (31)$$

$$\dot{y}_r = -\frac{\partial V_a}{\partial y_r} - \frac{1}{3} \left( \frac{\partial V_{rL}}{\partial y_r} + \frac{\partial V_{rM}}{\partial y_r} + \frac{\partial V_{rR}}{\partial y_r} \right) \quad (32)$$

gdje su promjene atraktivnih i repulzivnih sila opisane jednadžbama:

$$\frac{\partial V_a}{\partial x_r} = k_a \frac{x_r - x_a}{\|r_r - r_a\| + \varepsilon} \quad (33)$$

$$\frac{\partial V_a}{\partial y_r} = k_a \frac{y_r - y_a}{\|r_r - r_a\| + \varepsilon} \quad (34)$$

$$\frac{\partial V_{ri}}{\partial x_r} = \begin{cases} -k_r \frac{x_r - x_{oi}}{\|r_r - r_{oi}\| + \varepsilon} & \text{ako } \|r_r - r_{oi}\| \leq B \\ 0 & \text{ako } \|r_r - r_{oi}\| > B \end{cases} \quad (35)$$

$$\frac{\partial V_{ri}}{\partial y_r} = \begin{cases} -k_r \frac{y_r - y_{oi}}{\|r_r - r_{oi}\| + \varepsilon} & \text{ako } \|r_r - r_{oi}\| \leq B \\ 0 & \text{ako } \|r_r - r_{oi}\| > B \end{cases} \quad (36)$$

Sam proces generiranja referentne trajektorije se odvija iterativno, ponavlja se sve dok mobilni robot ne dođe do cilja.

Također valja napomenuti da kod metode potencijalnih polja postoji problem zapinjanja robota u točki lokalnog minimuma ako dođe do izjednačavanja potencijalne i repulzivne. Taj problem je posebice u situacijama u kojima mobilni robot prolazi blizu zida ili kroz uski prolaz.

U ovom radu će po uzoru na literaturu [4] i [5] biti izrađena tri modela za izbjegavanje lokalnog minimuma:

1. Uvođenje rotacijskog potencijala
2. Slijeđenje zida
3. Prolaz kroz uski prolaz

Iako će programsko rješenje za odabir metode praćenja zida i prolaska kroz uske prolaze biti uvedeni u program, oni će biti zakomentirani, a testiranja će se vršiti samo za klasičnu metodu sa dodatkom rotacijskog potencijala.

### 2.4.2.1 Uvođenje rotacijskog potencijala

Rotacijski potencijal je potencijal koji dijeluje okomito na repulzivni potencijal, čime pomaže robot u izbjegavanju lokalnog minimuma.

Uvođenjem rotacijskog potencijala dobivaju se nove jednadžbe za generiranje promjene referentne trajektorije mobilnog robota:

$$\dot{x}_r = -\frac{\partial V_a}{\partial x_r} - \mu_{rot} \frac{\partial V_r}{\partial x_r} - (1 - \mu_{rot}) \frac{\partial V_r}{\partial y_r} \quad (37)$$

$$\dot{y}_r = -\frac{\partial V_a}{\partial y_r} - \mu_{rot} \frac{\partial V_r}{\partial y_r} + (1 - \mu_{rot}) \frac{\partial V_r}{\partial x_r} \quad (38)$$

gdje su  $\frac{\partial V_r}{\partial x_r}$  i  $\frac{\partial V_r}{\partial y_r}$  jednaki.

$$\frac{\partial V_r}{\partial x_r} = \frac{1}{3} \left( \frac{\partial V_{rL}}{\partial x_r} + \frac{\partial V_{rM}}{\partial x_r} + \frac{\partial V_{rR}}{\partial x_r} \right) \quad (39)$$

$$\frac{\partial V_r}{\partial y_r} = \frac{1}{3} \left( \frac{\partial V_{rL}}{\partial y_r} + \frac{\partial V_{rM}}{\partial y_r} + \frac{\partial V_{rR}}{\partial y_r} \right) \quad (40)$$

$\mu_{rot}$  je broj između 0 i 1 definira omjer jakosti između rotacijskog i repulzivnog potencijala.

### 2.4.2.2 Uvođenje modela praćenja zida

Model praćenja zida se aktivira u situacijama kad se mobilni robot nađe ispred dugačke prepreke koju pokušava zaobići. Za određivanje modela praćenja zida potrebno je odrediti dominantan senzor (lijevi ili desni) koji prati prepreku. Uvjet za paljenje modela praćenja zida bi mogao biti kada sva tri senzora daju približne vrijednosti. Kada se upali model praćenja zida gasi se atraktivni potencijal, a promjena referentne trajektorije se računa preko jednadžbi:

$$\dot{x}_r = -\mu_{rot} \frac{\partial V_r}{\partial x_r} - (1 - \mu_{rot}) \frac{\partial V_r}{\partial y_r} \quad (41)$$

$$\dot{y}_r = -\mu_{rot} \frac{\partial V_r}{\partial y_r} + (1 - \mu_{rot}) \frac{\partial V_r}{\partial x_r} \quad (42)$$

S time da se promjena repulzivne sile sada računa preko formula:

$$\frac{\partial V_{ri}}{\partial x_r} = \begin{cases} -k_r \frac{x_r - x_{oi}}{\|r_r - r_{oi}\| + \varepsilon} & \text{ako } \|r_r - r_{oi}\| \leq B \\ k_r \frac{x_r - x_{oi}}{sl \|r_r - r_{oi}\| + \varepsilon} & \text{ako } \|r_r - r_{oi}\| \leq 2B \\ 0 & \text{ako } \|r_r - r_{oi}\| > 2B \end{cases} \quad (43)$$

$$\frac{\partial V_{ri}}{\partial y_r} = \begin{cases} -k_r \frac{y_r - y_{oi}}{\|r_r - r_{oi}\| + \varepsilon} & \text{ako } \|r_r - r_{oi}\| \leq B \\ k_r \frac{y_r - y_{oi}}{sl \|r_r - r_{oi}\| + \varepsilon} & \text{ako } \|r_r - r_{oi}\| \leq 2B \\ 0 & \text{ako } \|r_r - r_{oi}\| > 2B \end{cases} \quad (44)$$

gdje je  $sl$  faktor koji smanjuje repulzivnu silu u području gibanja između  $B$  i  $2B$ . U tom području je također poželjno da se promjeni  $\mu_{rot}$ .

#### 2.4.2.3 Model prolaska kroz uske prolaze

Mod prolaska kroz uski prolaz se aktivira kada lijevi i desni senzor pokazuju slične vrijednosti, a srednji senzor ne zamjećuje prepreku. U modelu prolaska kroz uske prolaze se promjena referentne trajektorije računa preko jednadžbi (37) i (38), s time da se jednadžbe (39) i (40) mijenjaju u jednadžbe.

$$\frac{\partial V_r}{\partial x_r} = \frac{1}{3} \left( \frac{\partial V_{rL}}{\partial x_r} + \frac{\partial V_{rM}}{\partial x_r} - \frac{\partial V_{rR}}{\partial x_r} \right) \quad (45)$$

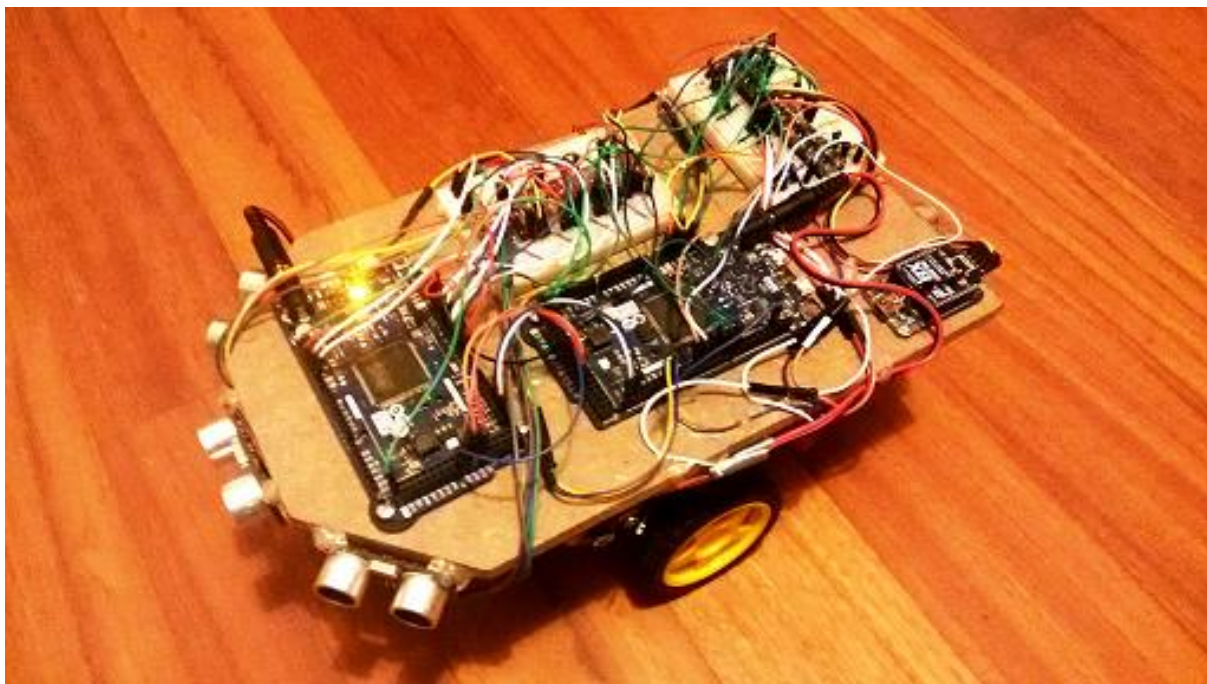
$$\frac{\partial V_r}{\partial y_r} = \frac{1}{3} \left( \frac{\partial V_{rL}}{\partial y_r} + \frac{\partial V_{rM}}{\partial y_r} - \frac{\partial V_{rR}}{\partial y_r} \right) \quad (46)$$

Navedene jednadžbe bi se teoretski mogle aktivirati ako mobilni robot padne u točku lokalnog minimuma, a nije upaljen niti jedan od dva moda rada.

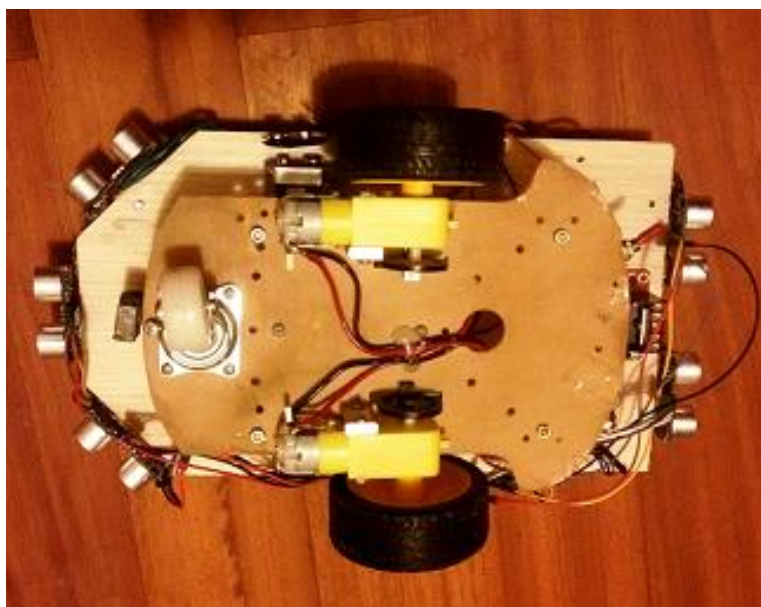
Osnovna ideja metode potencijalnih polja je u tome da integracijom promjene referentne trajektorije se konstantno generira referentna trajektorija, koju mobilni robot pomoću analitičkog neizrazitog regulatora dostigne, s time da se brzine lijevog i desnog kotača postižu preko PID regulatora.

### 3. IZRADA MOBILNOG ROBOTA S DIFERENCIJALNIM POGONOM

U prethodnom poglavlju je dana osnovna teoretska podloga za reaktivnu navigaciju mobilnog robota. Glavna tema ovog rada je realna izvedba reaktivne navigacije u praksi. Za te potrebe izrađen je mobilni robot na diferencijalni pogon sa 5 ultrazvučnih senzora [Slika 5] i [Slika 6].



Slika 5. Mobilni robot gornji dio



Slika 6. Mobilni robot donji dio





problem sanirao, ugrađen je mali "uteg" netom iznad slobodnog kotača koji djeluje kao protuteža gornjem dijelu trupa.

U idealnom slučaju trup bi trebao biti dizajniran tako da je donji dio veći od gornjeg. Cijeli mobilni robot bi trebao biti dizajniran tako da, kada bi opisali oko njega kružnicu, svi kutevi mobilnog robota bi doticali tu kružnicu. Zbog manjka odgovarajućeg alata i relativno kratkog vremena izrade trup mobilnog robota je izrađen dosta loše, no dovoljno dobro da se može testirati rad programa.

### 3.2 Podsustav za kretanje

Podsustav za kretanje u mobilnom robotu se sastoji od:

- Dva DC elektromotora,
- *H*-mosta,
- Tri kotača.

#### 3.2.1 Elektromotor

Kao aktuator mobilnog robota korišten je DC motor [Slika 8] sa prijenosnim omjerom 1:48 i radnim naponom između 3 i 12 V (idealno 3 do 8).



**Slika 8. DC motori**

Ovi elektromotori su dio kompleta [Slika 7] i odabrani su zato što je njihova osovina dimenzija jednakih kotaču enkoera i kotačima mobilnog robota, te zbog jednostavne ugradnje na donji dio trupa. Veliki nedostatak ovih elektromotora je to što je osovina plastična, pa ima malu savojnu čvrstoću. Zbog toga su kotači u izrađenom mobilnom robotu malo zakrivljeni, što dovodi do otežanog upravljanja istima.

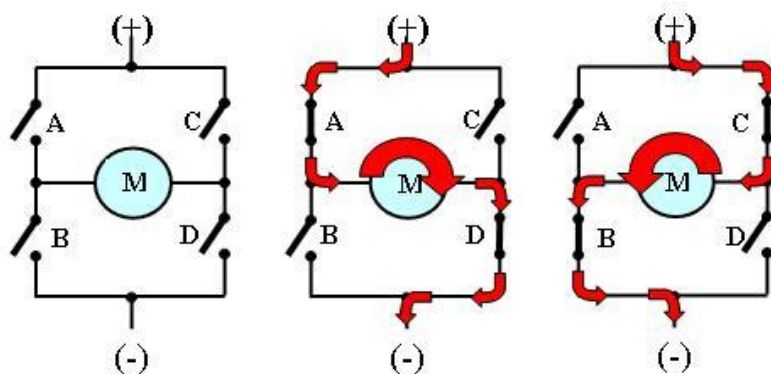
Također valja napomenuti da je za uspješno upravljanje mobilnim robotima potrebno između konektora motora zalemiti kondenzator od 0,1  $\mu$ F kako bi se prigušio električni šum.

Prigušenje električnog šuma je vrlo važno za regulaciju mobilnog robota, jer sprječava neočekivne promjene u vrtnji kotača.

### 3.2.2 H-most

Većina mikorkontrolera može na uređaje kojima upravlja slati napone od 0 V do 3,3 V, ili od 0 V do 5 V. Većina elektromotora u mobilnoj robotici rade na naponima većima od tih. Kako bi mikrokontroler mogao uspješno upravljati takvim elektromotorima, potrebno je koristiti upravljačku jedinicu koja će pojačati napon iz mikrokontrolera na odgovarajući napon u elektromotorima.

Svaki H-most radi poput 4 sklopke koje određuju smjer rotacije motora [Slika 9][6].



**Slika 9. Princip rada H-mosta**

Kada su sve sklopke otvorene motor miruje. Smjer rotacije mobilnog robota se određuje zatvaranjem kombinacija sklopki A i D ili B i C. Također je od velike važnosti napomenuti da se ne smiju istovremeno upaliti sklopke A i B ili C i D, pošto bi to dovelo do kratkog spoja.

Izrađeni mobilni robot koristiti dvostruki H-most *L298N* [Slika 10].



**Slika 10. Dvostruki H-most *L298N***

*L298N* u sebi sadrži dva H-mosta, što omogućava da se preko njega upravlja sa dva elektromotora. Ovo svojstvo je korisno zbog uštede prostora na mobilnom robotu. U [Tablica

1] dan je ispis važnijih dijelova H-mosta i mjesto spajanja na mobilnom robotu i mikrokontroleru [7]. Od velike je važnosti za napomenuti da ako ne spojimo *GND* na H-mostu sa uzemljenjem od mikrokontrolera, elektromotori se neće moći pokretati.

**Tablica 1. Osnovni dijelovi dvostrukog H-mosta L298N**

<i>L298N</i>	Mobilni robot	Mikrokontroler
<i>OUT 1</i>	Pozitivni kontakt lijevog kotača	
<i>OUT 2</i>	Negativni kontakt lijevog kotača	
<i>OUT 3</i>	Pozitivni kontakt desnog kotača	
<i>OUT 4</i>	Negativni kontakt desnog kotača	
<i>+12V</i>	Pozitivan pol baterije	
<i>GND</i>	Negativan pol baterije i uzemljenje na mikrokontroleru	Uzemljenje mikrokontrolera ( <i>GND</i> )
<i>+5V</i>		5V na mikrokontroleru samo ako je napon elektromotora iznad 12 V
<i>ENA</i>		PWM signal lijevog kotača
<i>IN1</i>		Gibanje natrag lijevi kotač
<i>IN2</i>		Gibanje naprijed lijevi kotač
<i>IN3</i>		Gibanje natrag desni kotač
<i>IN4</i>		Gibanje naprijed desni kotač
<i>ENB</i>		PWM signal desnog kotača

### 3.2.3 Kotači

Mobilni robot ima tri kotača, od kojih su dva pogonska kotača promjera  $D_k$  65 mm i jedan pomoćni kotač [Slika 7]. Trup mobilnog robota je dizajniran tako da je razmak između dvaju pogonskih kotača  $L$  147 mm, što je važno za izračun kutna brzine (5).

Pomoćni kotač mobilnog robota je jako osjetljiv, što zahtjeva dodatno opterećenje na njemu kako bi se postiglo relativno stabilno gibanje. Također valja napomenuti da su pogonski

kotači dizajnirani od materijala koji je pogodniji za gibanje po betonu nego po glatkoj podlozi, što je dovelo do velikog problema sa proklizavanjem.

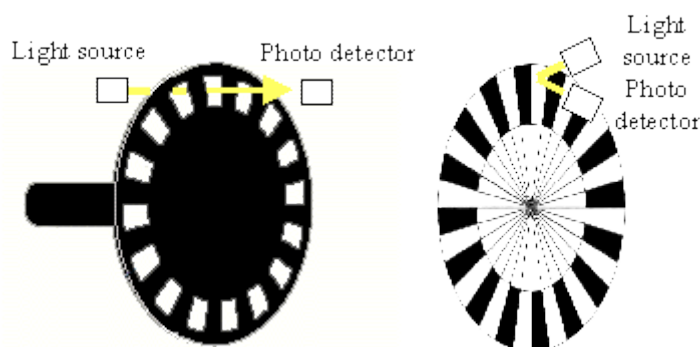
### 3.3 Senzori

U navedeni sustav spadaju svi senzori pomoću kojih mobilni robot dolazi do saznanja o vlastitim parametrima i o okolini u kojoj se nalazi. Taj podsustav se može smatrati drugim najvažnijim podsustavom, pošto bez njega mobilni robot ne može imati autonomnost. Ovaj podsustav se sastoji od dvije vrste senzora:

- Inkrementalnog optičkog enkodera,
- Ultrazvučnog senzora.

#### 3.3.1 Inkrementalni optički enkoderi

Inkrementalni optički enkoderi pripadaju vrsti odometrijskih senzora pomoću kojih se može odrediti položaj mobilnog robota (tj. elektromotora) [8]. Optički enkoderi se sastoje od odašiljača svjetlosti i osjetila svjetlosti. Inkrementalni optički enkoderi također sadrže kotač koji se postavlja na osovinu elektromotora. Kotači enkodera mogu biti izrađeni na više načina, no za precizno određivanje brzine vrtnje motora važno je da se u jednakim rasponima pojavljuju bijeli (prorezi) i crni (puni) dijelovi [Slika 11]. Broj proreza utječe na rezoluciju enkodera, čime raste preciznost mjerenja, no isto tako se povećava i cijena samog enkodera, pošto je potrebno imati dovoljno uzak snop svjetlosti koji bi mogao prolaziti samo kroz jedan prorez u svakom trenutku.



Slika 11. Kotač enkodera

Inkrementalni enkoderi rade tako da prate promjene stanja na osjetilu svjetlosti, tako da u trenutku kada osjetilo prima svjetlost šalje 1 (visoki napon) na mikrokontroler, a dok ne prima svjetlost šalje 0 (niski napon). Mikrokontroler određuje brzinu vrtnje elektromotora na

dva načina. Prvi način je da broji prijelaze iz niskog u visoko stanje enkodera i zabilježuje ih u određenom vremenskom periodu. Drugi način rada je da mikorkontroler bilježi promjene stanja iz niskog u visoko, pa zatim iz visokog u nisko čime se dobiva dvostruka rezolucija enkodera. Takav način rada ima svoje probleme koji će biti prokomentirani kasnije.

Također valja napomenuti da postoje dvije vrste inkrementalnih enkodera:

- Jednofazni inkrementalni enkoder: Sadrži jedan svjetlosni izvor i osjetilo svjetlosti i može očitavati samo brzinu vrtnje motora.
- Dvofazni inkrementalni enkoder: Sadrži dva izvora svjetlosti i osjetila koji su međusobno pomaknuti u fazi za  $90^\circ$  što omogućava određivanje smjera vrtnje motora.

U radu je pokušano ugrađivanje dvofaznog inkrementalnog enkodera rezolucije 100 [Slika 12]. No, tokom eksperimenta sa radom enkodera, došlo je do pojave problema koji je ukazao na glavni nedostatak elektromotora. Enkoder nije uspevao pravilno očitavati vrijednosti tokom vrtnje kotača. Problem se javio u tome što je enkoder sa rezolucijom 100 linija jako osjetljiv na promjene na osovini na kojoj se nalazi. Pošto je kotač sklon vibracijama, tijekom gibanja je došlo do velikog šuma tokom očitavanja. Također je zamiječeno da elektromotori nisu potpuno u ravnini sa podlogom, što je dovodilo do toga da kotač ulazi između senzora pod malim kutom, što dovodi do refleksije svjetlosti.



**Slika 12. Dvofazni inkrementalni enkoder**

Iz eksperimenta sa dvofaznim enkoderom može se zaključiti da je jako važno odvojiti osovinu elektromotora od kotača kako bi enkoder koji se nalazi na osovini elektromotora mogao biti što stabilniji.

Za daljnji rad je iskorišten jednofazni inkrementalni enkoder rezolucije 20 [Slika 13]. Jedna od glavnih karakteristika kotača korištenog enkodera je u tome da širina punih i praznih dijelova nije ista što ograničava preciznost mjerenja u režimu rada u kojem se mjeri promjena stanja

enkodera. [Tablica 2] daje popis *pin*-ova u korištenom enkoderu i mjesta u mikrokontroleru gdje se isti moraju spojiti.



Slika 13. Jednofazni inkrementalni enkoder

Tablica 2. *Pin*-ovi jednofaznog inkrementalnog enkodera

Enkoder	Mikrokontroler
5V	5V
GND	GND
OUT	<i>Pin sa mogućnošću External interrupta (Važno).</i>

Kod korištenog enkodera važno je napomenuti da se često pojavljuje jako veliki šum. Kako bi se taj šum sanirao, potrebno je implementirati niskopropusni filtar. To se može izvesti tako da se između *GND* i *OUT* izlaza zalemi kondenzator od 1,1  $\mu\text{F}$ , a u mikrokontroleru postavi otpor na korištenom *pin*-u. Time se stvara niskopropusni RC-filtar koji prigušuje sve visoke frekvencije (šum) na enkoderu.

Najlakši način ispravnosti enkodera je da se ručno pokuša zavrtiti enkoder za točno jedan krug. Ako mikrokontroler izbroji promjene jednake rezoluciji enkodera, znači da enkoder radi zadovoljavajuće.

Enkoderi mogu biti korišteni za određivanje položaja mobilnog robota ili brzine. U radu enkoderima mjerimo brzinu lijevog i desnog kotača.

$$v_L = \frac{D_k \pi N_L}{rez \cdot (t_{eL} - t_{epL} + \varepsilon)} \quad (47)$$

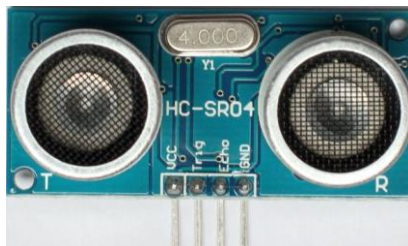
$$v_R = \frac{D_k \pi N_R}{rez \cdot (t_{eR} - t_{epR} + \varepsilon)} \quad (48)$$

Vrijednost *rez* predstavlja rezoluciju koja je jednaka 20 ili 40, ovisno o načinu inkrementacije enkodera.

### 3.3.2 Ultrazvučni senzor

Ultrazvučni senzori su vrsta senzora za određivanje udaljenosti, koji udaljenost od prepreke određuju preko vremena potrebnog da se odaslani zvuk vrati od prepreke.

Ultrazvučni senzor korišten u sklopu rada je *HC-SR04* [Slika 14]. Popis važnih *pin*-ova ultrazvučnog senzora i mjesta spajanja na mikrokontroleru prikazuje [Tablica 3].



Slika 14. *HC-SR04*

Tablica 3. *Pin*-ovi ultrazvučnog senzora *HC-SR04*

HC-SR04	Mikrokontroler
VCC	5V
TRIG	Izlazni <i>pin</i> na mikrokontroleru ( <i>OUTPUT</i> ) sa PWM
ECHO	Ulazni <i>pin</i> na mikrokontroleru ( <i>INPUT</i> )
GND	GND

Ultrazvučni senzor *HC-SR04* radi tako da prvo pošalje ultrazvučni val, te zatim čeka da se taj val vrati i pritom mjeri vrijeme trajanja puta vala. Iz izmjenog vremena se može odrediti udaljenost ultrazvučnog senzora od prepreke  $L_s$ .

$$L_s = \frac{t_s c}{2 \cdot 10^6} \approx \frac{t_s}{5880} \quad (49)$$

Vrijeme trajanja mjerenja  $t_s$  se mjeri u  $\mu s$  zbog definirane funkcije u mikrokontroleru (*Pulsein*), koja na taj način bilježi vrijeme trajanja mjerenja.

U radu je za određivanje udaljenosti prepreka od mobilnog robota korišteno 5 ultrazvučnih senzora, 3 sprijeda i 2 straga. Prednja tri senzora su postavljena tako da je srednji senzor postavljen u smjeru gibanja mobilnog robota, dok su lijevi i desni senzor zakrenuti pod kutem od  $45^\circ$  u odnosu na njega [Slika 5]. U idealnom slučaju bi se u centralnoj točki između dva pogonska kotača mogle sjeći tri dužine jednake duljine do svakog senzora. U ovom radu, to



nažalost nije slučaj zbog loše izrezanog gornjeg trupa, ali je aproksimacija dovoljno dobra za potrebe rada.

Stražnja dva senzora promatraju prema natrag i u sklopu ovog rada će služiti samo kao indikacija ako je mobilni robot se sudario sa preprekom gibajući se prema natrag.

Ultrazvučni senzori izračunavaju udaljenost detektirane prepreke u odnosu na njih same, no za potrebe metode potencijalnih polja potrebno je tu udaljenost pretvoriti u točku u prostoru koja bi djelovala kao centar repulzivnog potencijala. Za prednja tri ultrazvučna senzora to je izvedeno preko jednadžbi:

$$\begin{aligned}x_{oL} &= x + (L_{sL} + 0,154) \cos\left(\theta + \frac{\pi}{4}\right) \\y_{oL} &= y + (L_{sL} + 0,154) \sin\left(\theta + \frac{\pi}{4}\right)\end{aligned}\tag{50}$$

$$\begin{aligned}x_{oM} &= x + (L_{IM} + 0,154) \cos(\theta) \\y_{oM} &= y + (L_{sM} + 0,154) \sin(\theta)\end{aligned}\tag{51}$$

$$\begin{aligned}x_{oR} &= x + (L_{sR} + 0,154) \cos\left(\theta - \frac{\pi}{4}\right) \\y_{oR} &= y + (L_{sR} + 0,154) \sin\left(\theta - \frac{\pi}{4}\right)\end{aligned}\tag{52}$$

gdje je 0,154 udaljenost senzora od središnje točke između dva pogonska kotača mobilnog robota izražena u metrima.

Jedna od glavih mana određivanja udaljenosti mobilnog robota od prepreke preko ultrazvučnih senzora je u tome da nam taj senzor javlja udaljenost od prepreke no ne i gdje je ona pozicionirana u odnosu na njega. Ultrazvučni senzori također imaju i druge mane koje će biti opisane kasnije.

### 3.4 Podsustav za procesiranje

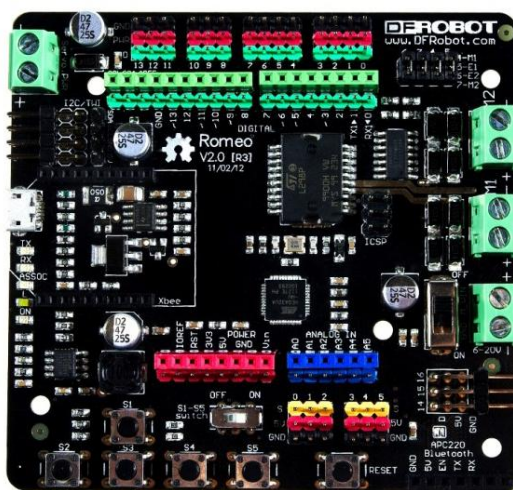
Najvažniji podsustav u mobilnom robotu, a sastoji se od mikrokontrolera kojim se upravlja sa svim ostalim komponentama i u kojem je uprogramiran sam način rada mobilnog robota. Za mikrokontrolere bi se moglo reći da su "mozak" mobilnog robota.

U početku se kao mikrokontrolerska ploča za upravljanje mobilnim robotom koristio *Arduino Romeo V2*. U završnoj realizaciji je odlučeno upravljati mobilnim robotom pomoću dva *Arduino Due* mikrokontrolera.

### 3.4.1 Arduino Romeo V2

*Arduino Romeo* je mikrokontrolerska ploča koja sadrži 8-bitni mikrokontroler *ATmega32u4* i specijalizirana je za jednostavne mobilne robote sa DC pogonom kotača [Slika 15]. Glavna prednost ove mikrokontrolerske ploče je u tome da u sebi ima integrirane [11]:

- H-most za upravljanje sa 2 DC elektromotora,
- Utor za *Xbee* uređaje,
- Utor za klasični *Bluetooth* uređaj i *Bluetooth shield*,
- Puno "muških" i "ženskih" utičnica za senzore,
- Puno 5V i GND utičnica koje su uz priključak za digitalni *pin* mikrokontrolera.



Slika 15. *Arduino Romeo V2*

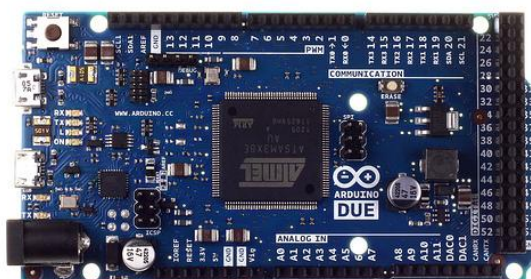
*Arduino Romeo V2* je preporučljiv za jednostavne mobilne robote pošto se na njemu nalaze priključci za spajanje svih potrebnih komponenti i dovoljan broj *GND* i *5V pin*-ova, što je važno kod korištenja velike količine senzora i komponenti općenito. Iako se mikrokontroler može programirati preko mikro USB-a, dobro je nabaviti i programator pošto je mikro USB utor vrlo osjetljiv.

Iako je *Arduino Romeo V2* vrlo dobra pločica za mobilne robote, u ovom radu nije korištena zbog tri razloga:

1. *Arduino Romeo* ima 5 *pin*-a koji imaju *External Interrupt*, što nije dovoljno za izradu ovog rada
2. Kao senzor za određivanje udaljenosti od prepreke se koriste ultrazvučni senzori,
3. Mikro USB utor se odlomio tokom rada sa pločicom.

### 3.4.2 Arduino Due

*Arduino Due* je mikrokontrolerska pločica sa 32-bitnim mikrokontrolerom *Atmel SAM3X8E ARM Cortex-M3 CPU* [Slika 16]. Radi boljeg prikaza prednosti *Arduino Due*-a, njegove karakteristike su uspoređene sa karakteristikama *Arduino Romeo V2* [Tablica 4] [11] [12] [13].



Slika 16. *Arduino Due*

Tablica 4. Usporedba *Arduino Due* i *Arduino Romeo V2*

Karakteristika	<i>Arduino Due</i>	<i>Arduino Romeo V2</i>
Procesor	32 bit	8 bit
Radni napon	3,3 V	5 V
Brzina procesora	84 MHz	16 MHz
Napon napajanja	6-16 V	6-23 V
Flash memorija	512 kB	32 kB (28 kB iskoristivo)
SRAM	96 kB	2,5 kB
Digitalni I/O	54	14
Analogni I	12	6 (+6 na Digitalnim I/O)
PWM	12	7
Serijska komunikacija	4	2
External interrupt	Svi <i>pin</i> -ovi	5 (3 realno iskoristiva)
5V <i>pin</i>	3	21
GND <i>pin</i>	5	22

Kao što se vidi, *Arduino Due* ima puno bolje karakteristike procesora i više *pin*-ova, što nam omogućava izradu mobilnih robota sa više dijelova, koji mogu obavljati složene zadatke. Također *Arduino Due* ima više portova za serijsku komunikaciju, što nam omogućava komunikaciju sa korisnikom, ali i sa drugim uređajima (npr. više mobilnih robota).

Iako se *Arduino Due* čini kao očiti pobjednik usporedbe, važno je napomenuti nekoliko razloga zbog kojih bi bilo bolje koristiti *Arduino Romeo V2*:

1. *Arduino Due* ima radni napon od 3,3 V, dok većina jeftinijih uređaja korištenih za izradu jednostavnih mobilnih robota (korisnih za isprobavanje programa i istraživanja) imaju radni napon od 5 V. Kako bi osigurali uspješan rad uređaja i izbjegli oštećenje mikrokontrolera koji je vrlo osjetljiv, potrebno je koristiti dvosmjerne logičke pretvarače koji će skalirati napon u 3,3 V u 5 V i obrnuto.
2. Zbog manje količine *5V-pin* i *GND pin*-ova, kao i zbog dvosmjernih logičkih pretvarača potrebno je koristiti eksperimentalnu pločicu za povezivanje komponenata sa mobilnim robotom, čime se povećava broj žica.
3. *Arduino Due* nema u sebi ukomponirane priključke za komunikaciju i upravljanje elektromotorima, što rezultira potrebom za dodatnim komponentama.
4. Veliki broj dodatnih komponenti i eksperimentalne pločice zauzimaju dodatan prostor čime se povećava minimalna moguća veličina mobilnog robota.
5. Iako je brzina procesora *Arduino Romeo*-a mnogo manja od *Arduino Due*-a, važno je napomenuti da su funkcije koje usporavaju vrijeme upravljačkog sustava funkcije koje su ukomponirane u programsko sučelje *Arduino IDE*. Te funkcije se mogu izbjeći ako se isprogramiraju same uz pomoć *port* manipulacije. Kako je *Arduino Romeo* 8-bitni sustav, takvo programiranje je mnogo lakše od programiranja 32-bitnih sustava poput *Arduino Due*-a. Također, *port* manipulacija *Arduino Romeo*-a je skoro identična onoj od *Arduino Uno*-a za kojeg ima puno više literature i gotovih rješenja nego za *Arduino Due*, koji je relativno nova mikroprocesorska pločica.

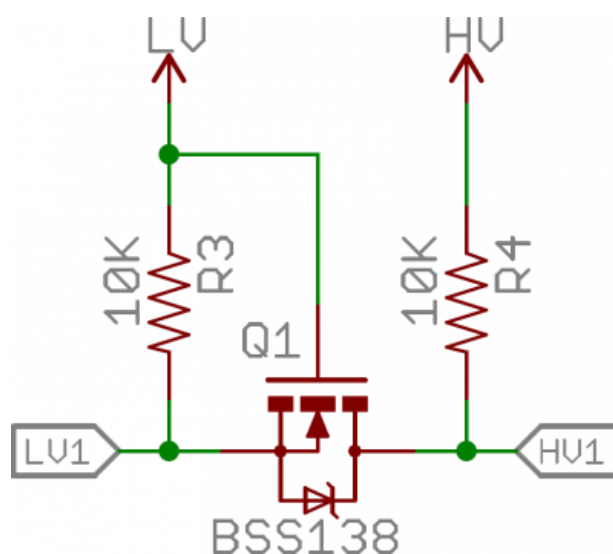
U sklopu ovog rada su korištena dva *Arduino Due*-a, s time da se jedan *Arduino Due* koristio za upravljanje mobilnim robotom, dok drugi služi za rad ultrazvučnih senzora. Sve komponente su povezane sa mikrokontrolerima preko dvosmjernih logičkih pretvarača, dok su mikrokontroleri međusobno povezani direktno. Dvosmjerni logički pretvarači su postavljeni na dvije eksperimentalne ploče tako da su svi koji spajaju ultrazvučne senzore postavljeni na jednu eksperimentalnu pločicu, dok su ostali postavljeni na drugu pločicu. Jedan od razloga za to je da se lakše može pratiti spajanje žica. Drugi razlog je da posao

napajanja logičkih pretvarača ne bude samo na jednoj bateriji što može rezultirati padom napona na mikrokontroleru i nemogućnošću rada programa. Na ovaj način su sve komponente podjeljene na mikrokontroler koji njima upravlja.

### 3.4.3 Dvosmjerni logički pretvarač

Kako bi se uređaji koji rade na naponima 5 V mogli upravljati pomoću mikrokontrolera sa radnim naponom 3,3 V i obrnuto, potrebno je između njih postaviti sklop koji skalira nadolazeće napone na ulazu u prikladne napone na izlazu. To se može izvesti pomoću elektroničkog sklopa koji se naziva dvosmjerni logički pretvarač.

Ima više načina izvedbe takvog sklopa, no svaki se sastoji od kombinacije MOSFET-a za pojačavanje napona i otpornika za smanjivanje napona. Na ulaze uređaja također moramo poslati referentni visoki napon (u ovom slučaju 5 V) i referentni niski napon (u ovom slučaju 3,3 V) pomoću kojih će sklop provoditi izmjenu napona. Primjer sheme jedne vrste izvedbe dvosmjernog logičkog pretvarača daje [Slika 17] [14]. Taj sklop se ponavlja u uređaju toliko puta koliko ima slobodnih priključaka za uređaje.

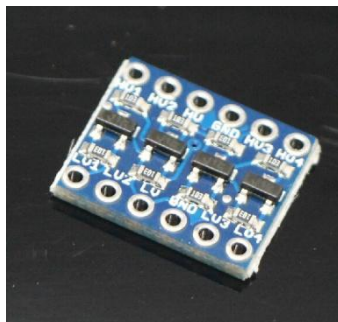


Slika 17. Shema dvosmjernog logičkog pretvarača

U sklopu rada su korištena dva tipa dvosmjernih logičkih pretvarača koji se razlikuju po načinu rada.

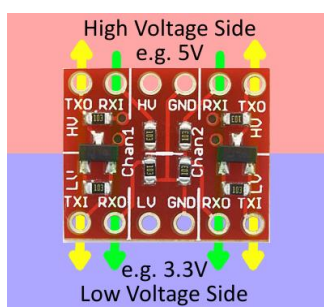
Prva verzija dvosmjernog logičkog pretvarača [Slika 18] radi na isti način poput sheme prikazane na prethodnoj slici [Slika 17] i kao takav je vrlo jednostavan za spajanje. U

mobilnom robotu ta verzija je korištena za spajanje *Arduino Due*-a sa H-mostom, *Xbee* uređajem i sa digitalnim enkoderima.



**Slika 18. Prva verzija dvosmjernog logičkog pretvarača**

Druga verzija pretvarača se razlikuje po tome da jedan priključak može samo smanjiti napon sa 5 V na 3,3 V (oznaka *RI* ili *RO*), dok je drugi priključak MOSFET koji može raditi u oba smjera [Slika 19] [15]. Kod te verzije važno je spajati priključke ovisno o tome ulazi li signal u mikrokontroler ili izlazi iz njega. U mobilnom robotu ta je verzija korištena za spajanje mikrokontrolera sa ultrazvučnim sensorima. *ECHO pin* šalje signal u mikrokontroler i kao takav se spaja na *RXI* dok *TRIG pin* prima signal iz mikrokontrolera, pa se spaja na *TXO*.



**Slika 19. Druga verzija dvosmjernog logičkog pretvarača**

Iako se prva verzija dvosmjernog logičkog pretvarača čini bolja, pošto se ne mora razmišljati je li *pin* koji povezuje ulaz ili izlaz, valja napomenuti da druga verzija dolazi sa već zalemljenim nogama za spajanje na eksperimentalnu ploču. Prva verzija dolazi sa odvojenim nožicama koje je potrebno zalemiti za pretvarač što zahtjeva lemilicu sa oštrim vrhom i spretnu ruku, pošto su razmaci između rupa vrlo mali. Iako se uspješno zalemiti 5 takvih pretvarača sa klasičnom lemilicom, dva su se pokvarila pošto je došlo do rastapanja otpornika tokom skidanja lema koji je povezao dvije susjedne rupe.

Princip povezivanja mikrokontrolera i uređaja preko dvosmjernog logičkog pretvarača za slučaj ultrazvučnih senzora [Tablica 3] i druge verzije pretvarača [Slika 19] prikazuje tablica [Tablica 5]. Taj princip spajanja se primjenjuje na sve ostale uređaje.

**Tablica 5. Spajanje mikrokontrolera i uređaja preko dvosmjernog pretvarača**

Pretvarača	Mikrokontroler	Ultrazvučni senzor
<i>HV</i>	<i>5V</i>	<i>VCC</i>
<i>GND</i> (uz <i>HV</i> )		<i>GND</i>
<i>TXO</i>		<i>TRIG</i>
<i>RXI</i>		<i>ECHO</i>
<i>LV</i>	<i>3,3V</i>	
<i>GND</i> (uz <i>LV</i> )	<i>GND</i>	
<i>TXI</i>	<i>Digital output pin</i>	
<i>RXO</i>	<i>Digital input pin</i>	

Važno je napomenuti kako žica koja spaja 5V na mikrokontroleru mora biti na eksperimentalnoj ploči ispred žice koja spaja VCC na ultrazvučnom senzoru. Iako *Arduino Due* ima referentne napone i 5 V i 3,3 V, sami digitalni *pin*-ovi rade na 3,3 V, pa ih se zato ne smije direktno spajati na senzore koji imaju radni napon 5 V.

Mikrokontroler nema dovoljno 5V i *GND pin*-ova za spajanje sa svim dvosmjernim logičkim pretvaračima, pa se *GND* i 5V spaja na + i - dio na eksperimentalnoj ploči. *HV* i *GND* se spajaju preko žica povezanih sa + i - dijelom na eksperimentalnoj ploči.

### 3.5 Podsustav za komunikaciju

U podsustav za komunikaciju spadaju uređaji koji služe za razmjenu informacija između mobilnog robota i nekog drugog uređaja. U ovom radu se kao veza između kompjutera i mobilnog robota koristi *Xbee 1 mW Wire Antenna Series 1* [Slika 20]. Sam uređaj šalje podatke uz pomoć radio frekvencija, a glavna prednost mu je što ima doomet od 100 m na otvorenom. U unarnjem prostoru jako dobro radi unutar 10 m, no moguća je pojava gubitka informacije kada *Arduino* prolazi kroz natkriven prostor (npr. stolice ili stolove).

Slika 20. *Xbee 1 mW Wire Antenna Series 1*

Kako bismo *Xbee* uređaj mogli spojiti na računalo potrebno je koristiti *Xbee-USB* adapter [Slika 21]. Korišteni adapter također ima pinove koji omogućavaju spajanje sa mikrokontrolerom, pa je taj adapter također korišten kao veza između *Xbee*-a i mikrokontrolera.

Način spajanja *Xbee* adaptera sa mikrokontrolerom prikazuje tablica [Tablica 6]. Kao što tablica pokazuje, spajanje je vrlo jednostavno, što je glavna prednost adaptera.

Slika 21. *Xbee-USB* adapterTablica 6. *Pin-ovi Xbee-USB* adaptera

Adapter	Mikrokontroler
<i>RST</i>	<i>RESET</i>
<i>RX</i>	<i>RX</i>
<i>TX</i>	<i>TX</i>
<i>GND</i>	<i>GND</i>
<i>5V</i>	<i>5V</i>

*Xbee* uređaj se također može spojiti preko eksperimentalne pločice sa mikrokontrolerom. U tom slučaju nije potrebno koristiti dvosmjerni logički pretvarač, pošto je radni napon samog *Xbee* uređaja 3,3 V. Razlika u spajanju je ta što se mora točno znati koja noga *Xbee*-a predstavlja koji *pin* i što se *RX pin* na *Xbee*-u treba spojiti na *TX port* na mikrokontroleru.



Također valja napomenuti da sama serijska komunikacija troši relativno puno energije i vremenski je zahtjevnija, pa ona treba biti minimalna. U programu minimizacija komunikacije neće biti izvedena, no dati će se osnovna ideja za njenu izvedbu.

### 3.6 Podsustav za napajanje

Vrlo važan dio projektiranja mobilnog robota je odabir izvora napajanja za mobilni robot. Baterija bi morala biti dovoljno velikog napona da može osigurati rad mikrokontrolera i korištenih uređaja u zadovoljavajućem vremenskom periodu.

Mobilni robot ima tri izvora napajanja:

1. Izvor od 8 1,2 V punjivih NiMh baterija koji služi za rad elektromotora. Što je napon ove baterije veći to će biti veći maksimalni napon na elektromotoru. Za korišteni napon baterije, maksimalni napon koji elektromotori mogu primiti je oko 8 V, što je zadovoljavajuće za potrebe rada. Punjive baterije, osim što su dugoročno jeftinije, su bolje za napajanje elektromotora jer duže zadržavaju istu razinu napona nego nepunjive 1,5 V baterije (kojima napon linearno pada sa uporabom), što je za upravljanje mobilnim robotom jako važno.
2. 9 V nepunjiva baterija za napajanje mikrokontrolera koji radi sa ultrazvučnim senzorima.
3. 9 V nepunjiva baterija za napajanje mikrokontrolera koji izvodi program i upravlja sa ostalim uređajima.

Iako je korištenje 9 V nepunjivih baterija vrlo loše rješenje pošto klasične 9 V baterije imaju vrlo kratak rok uporabe, pa brzo postaje skupo i loše za okoliš, izrađeni mobilni robot ih koristi zato jer ne zauzimaju veliki prostor i zato jer su dovoljnog napona da omoguće uspješan rad mikrokontrolera. Važno je napomenuti da potreban napon za uspješan rad mikrokontrolera raste sa količinom uređaja koji su priključeni na njemu.

Idealno rješenje za napajanje mobilnog robota bi bilo sljedeće: i H-most i mikrokontroleri bi se napajali iz istog punjivog izvora napajanja relativno velikog napona. Prije ulaza na pojedine uređaje bi bio spojen sklop koji bi osiguravao da na ulaz uređaja dolazi konstantan napon (npr. 10 V H-most, 9 V mikrokontroleri). Time bi se riješio problem koji otežava upravljanje mobilnim robotom zbog promjene u naponu u H-mostu, o čemu će biti više rečeno kasnije. Takva verzija nije izvedena zbog nedostatka vremena u izradi rada.

## 4. UPRAVLJANJE ELEKTRONIČKIM SKLOPOVIMA

U prethodnom poglavlju je dan popis komponenti mobilnog robota i način na koji te komponente rade. Prije same izrade programa za navigaciju mobilnog robota potrebno je prvo proučiti kako mikrokontroler upravlja elektroničkim sklopovima mobilnog robota.

U ovom programu će biti objašnjeni principi rada algoritama za:

- Upravljanje enkoderima,
- Upravljanje elektromotorima,
- Komunikaciju između mobilnog robota i računala,
- Komunikaciju između dva mikrokontrolera,
- Rad ultrazvučnih senzora.

Mikrokontroler je programiran u sučelju *Arduino IDE* 1.5.7, dok je program na računalu izrađen u programskom jeziku *Python* 2.6.

### 4.1 Upravljanje enkoderima

U prethodnom poglavlju je objašnjeno kako enkoder povećava svoju vrijednost na dva načina. Prvi je način da zabilježi kada se na osjetilu svjetlosti pojavi svjetlost, dok je drugi način da zabilježi promjenu stanja.

Kako sam mikrokontroler radi tako da se programi vrte liniju po liniju, a promjene stanja na enkoderu se događaju ovisno o brzini vrtnje kotača, nemoguće je uprogramirati rad enkodera u glavnom dijelu programa.

Kako bi se uspješno zabilježivale promjene u enkoderu potrebno je koristiti protokol *External Interrupt*. *External Interrupt* je protokol koji zaustavlja rad programa uzrokovan vanjskim učinkom. *External Interrupt* također pokreće događaj (*event*) koji radi promjene na programu ovisno o funkciji koja se izvršava.

U *Arduino IDE* *External Interrupt* se koristi pomoću funkcije *AttachInterrupt()*[16]. Ta funkcija pokreće *event* ako se na definiranom *pin*-u dogodi određeni uvjet.

U ovom radu se *AttachInterrupt()* koristi kako bi, svaki put kad se dogodi željena promjena stanja na enkoderu, pokrenuo funkciju koja povećava vrijednost na enkoderu  $N_L$  ili  $N_R$  i bilježio trenutak kada se ta funkcija dogodila.

Kod rada enkodera koji bilježi promjenu sa niskog na visoko uvjet za pokretanje *AttachInterrupt()* označava se sa *RISING*, dok se kod rada enkodera koji bilježi bilo kakvu promjenu uvjet označava sa *CHANGE*. Događaj koji *AttachInterrupt()* pokreće je izvršavanje funkcije koja povećava vrijednost enkodera i bilježi vrijeme u milisekundama u kojem se događaj dogodio. Pošto se brzine lijevog i desnog kotača mogu razlikovati, za pojedini kotač mora biti definirani vlastiti *AttachInterrupt()*.

Neke od karakteristika *AttachInterrupt()* su sljedeće:

- Pokreće se svaki put kada se zadani uvjet ispuni, pa je zato važno da događaj koji poziva bude kratak kako program ne bi zapeo u beskonačnom krugu
- Unutar njega je vrijeme konstantno pa se ne mogu koristiti vremenske funkcije tipa *delay()*.
- *Interrupt*-i se izvode po trenutku događaja, pa ako se promjena stanja na lijevom enkoderu dogodila dok se još vršila funkcija uzrokovana promjenom stanja na desnom enkoderu, ona će se pokrenuti nakon što završi rad prethodne funkcije

## 4.2 Upravljanje elektromotorima

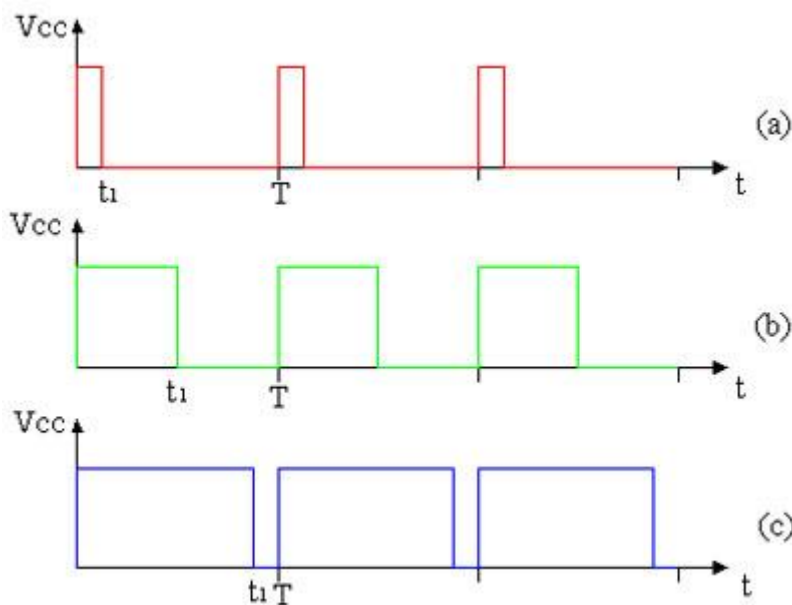
Brzina elektromotora se određuje postavljanjem određenog napona na elektromotorima. Mikrokontroleri mogu raditi samo u dva stanja visoko u kojem je napon jednak radnom naponu *pin*-ova (u slučaju *Arduino Due*-a to je 3,3 V) i nisko u kojem je napon približno jednak 0. U takvom načinu rada elektromotor bi se mogao gibati samo brzinom jednakom maksimalnom naponu kojeg šalje H-most upravljani od strane elektromotora.

Kako bi se mogla postići promjena brzina mikrokontroler koristi PWM (*Pulse Width Modulation*), pomoću kojeg je moguće na izlazu mikrokontrolera dobiti spektar brzina između visokog i niskog napona (u slučaju *Arduino Due*-a između 3,3 V i 0 V).

PWM je metoda kojom mikrokontroler pretvara digitalni signal u odgovarajući analogni signal (u ovom slučaju napon za upravljanje elektromotrima). To se ostvaruje tako da se na relativno velikoj frekvenciji ponavljaju ciklusi visoke (3,3 V) i niske razine signala (0 V). Ovisno o vremenu trajanja visokog stanja dobiva se ukupna snaga signala. To je bolje prikazano slikom [Slika 22] [17].

Kod *Arduino Due*-a PWM signal je 8-bitan, što znači da se period u kojem signal u visokoj razini može kretati između 0 (0%) i 255 (100%). To znači da, pomoću PWM signala, napon od 3,3 V možemo podijeliti na 255 dijelova.

U *Arduino IDE* PWM signal se generira preko funkcije *analogWrite()* koja na izabrani *pin* šalje željenu vrijednost PWM signala (0 do 255). Kod izrade programa za regulaciju brzine (PD regulatora u ovom radu), potrebno je da vrijednost upravljačke jedinice bude u rasponu od 0 do 255, a ne vrijednost napona koja se šalje na elektromotore.



**Slika 22. PWM signali**

Također valja napomenuti da PWM signal generiran frekvencijom standardnom za *Arduino Due* proizvodi vrlo neugodan zvuk. To se može popraviti tako da se promjeni frekvencija PWM signala na više frekvencije. No, pošto elektromotrima upravljamo preko H-mosta koji može podnijeti rad na ograničenom spektru frekvencija važno je biti oprezan kod mijenjanja frekvencije PWM signala. U sklopu ovog rada frekvencija PWM signala se nije mjenjala.

### 4.3 Komunikacija između mobilnog robota i računala

U sklopu rada potrebno je uspostaviti komunikaciju između mobilnog robota i računala. Na računalu korisnik postavlja željene vrijednosti parametara mobilnog robota i određuje što mobilni robot treba napraviti. Računalo također šalje naredbu za početak rada mobilnog robota.

Računalo prima podatke iz mobilnog robota, sortira ih i preko njih izrađuje grafičke prikaze pojedinih varijabli (npr brzine, upravljačke varijable, itd.).

Mobilni robot šalje računalu povratnu informaciju o namještenim parametrima, kao i neke, korisniku zanimljive, izračunate vrijednosti.

Robot iz računala prima vrijednosti pojedinih parametara na osnovu kojih se mijenjaju neke upravljačke veličine, a time i sam način rada mobilnog robota. Mobilni robot također od računala prima naredbu za početak rada programa.

Mobilni robot komunicira sa računalom preko *Xbee* uređaja opisanog u prethodnom poglavlju. Mobilni robot i računalo međusobno ostvaruju bežičnu serijsku komunikaciju.

Serijska komunikacija je vrsta komunikacije u kojoj uređaj preko serijskog kanala šalje informacije na drugi uređaj bit po bit.

Pošto je mobilni robot programiran u sučelju *Arduino IDE*, a program na računalu je isprogramiran u *Python*-u, načini izvođenja serijske komunikacije su drugačiji. Zbog toga serijsku komunikaciju možemo podijeliti na:

- Komunikacija od strane mobilnog robota
- Komunikacija od strane računala

#### **4.3.1 Serijska komunikacija - mobilni robot**

*Arduino IDE* za serijsku komunikaciju koristi biblioteku (*library*) *Serial*.

Za uspješnu serijsku komunikaciju potrebno je u inicijalizaciji programa dodati naredbu *Serial3.begin()* koja otvara željeni kanal (u radu kanal 3 što pokazuje brojka 3 ispred *Serial*) i postavlja brzinu prijenosa podataka. Za uspješan razmjenu podataka obavezno je da ta brzina bude ista i za mobilni robot i za računalo. U ovom radu brzina komunikacije je postavljena na 9600 bit/s. Maksimalna brzina prijenosa je određena mogućnostima *Xbee* uređaja, no valja napomenuti kako se povećanjem brzine prijenosa povećava i potrošnja energije.

Slanje podataka od strane mobilnog robota je vrlo jednostavno i odvija se preko naredbe *Serial3.println()* koja šalje željenu vrijednost ili string na računalo i postavlja se u novi red. Važno je za napomenuti da postoji i naredba *Serial3.print()* koja piše vrijednosti u istom redu, no takav način zapisa računalo pogrešno interpretira.

Primanje podataka je malo složenije pošto računalo šalje mobilnom robotu niz podataka koji mobilni robot prima znak po znak, pa se javlja problem razlučivanja podataka. Taj problem je riješen preko funkcije *getSerial()* [18]. Ta funkcija razvrstava vrijednosti pomoću znaka '/' kojeg računalo šalje kako bi označilo kraj vrijednosti. Ta funkcija bi niz [2,/,3,3,/,1,1,2,/] razvrstala na vrijednosti 2,33,112. Za funkciju treba napomenuti dvije činjenice:

1. Funkcija radi dok god se ne pojavi znak '/'. Nakon toga ona vraća dobivenu vrijednost koja će biti jedna od varijabli. Kako bi se očitala druga vrijednost niza potrebno je

ponovo pozvati funkciju. Zbog toga je potrebno biti poznata točna količina informacija koje mobilni robot prima od računala.

2. Funkcija pretvara dobivene znakove u prirodne brojeve (*abs(integer)*). Sve decimalne vrijednosti koje šaljemo sa računala je zato potrebno pretvoriti u *integer* (prirodan broj), pa ih u mikrokontroleru ponovo pretvoriti u decimalnu vrijednost. Tako bi, na primjer, broj 0,22 računalo pretvorilo u 22 i poslalo mobilnom robotu. Mobilni robot bi zatim tu vrijednost podijelio sa 100 i pretvorio u svoju stvarnu vrijednost. Naravno, vrijednost sa kojom računalo množi decimalni broj mora biti poznata.

#### 4.3.2 Serijska komunikacija - računalo

Serijska komunikacija u *Python*-u se pomoću biblioteke *pySerial* koji sadrži funkcije za postavljanje serijske veze kao i za čitanje i slanje podataka preko serijske veze.

Prije nego što je moguća komunikacija potrebno je prvo otvoriti kanal serijske veze. Kako bi mogli otvoriti kanal, potrebno je poznavati *port* na kojem se *Xbee* uređaj nalazi i upisati ga u programu (u ovom radu se koristio 'COM12').

Računalo na mobilni robot šalje nizove znakova izrađenih na način da zadovoljavaju kriterij za funkciju *getSerial()* na mobilnom robotu.

Primjer niza koji računalo šalje bi bilo 0/a/nizb/, gdje su:

- 0: najčešće prva vrijednost koju šalje računalo je nekakva vrijednost koja nema veze sa poslanim nizom, pa je na početku svakog novog niza poslana 0 koju mobilni robot ignorira.
- a: Skupina parametara koji čije će se vrijednosti upisivati. Ovaj dio će biti detaljnije opisan u sljedećem poglavlju.
- nizb: Niz koji sadrži predodređenu količinu parametara (npr. 1,1/0/,1,0,1,/)

Računalo sa mikrokontrolera prima komade podataka (koje mobilni robot odvaja sa *Serial3.println()*), razdjeljuje ih i, ako je potrebno, razvrstava u pojedine nizove.

Valja napomenuti kako je potrebno definirati *timeout* koji prekida izvršavanje pojedine funkcije kako ne bi došlo do zaglavlivanja programa zbog gubitka veze između mobilnog robota i računala.

#### 4.4 Komunikacija između dva mikrokontrolera

Mobilni robot ima dva mikrokontrolera, od kojih jedan služi za izvođenje glavnog programa, dok drugi program provodi rad ultrazvučnih senzora. Kako su za reaktivnu navigaciju potrebne vrijednosti dobivene iz ultrazvučnih senzora, mora se ostvariti komunikacija između mikrokontrolera.

U *Arduino IDE* to se izvodi preko biblioteke *Wire*. Ova biblioteka nam omogućava da se ostvari *I2C/TWI* komunikacija između dvaju mikrokontrolera. Za uspješno provođenje takve komunikacije ta dva mikrokontrolera je potrebno je žicama spojiti *SDA* i *SCL pin*-ove mikrokontrolera, kao i *GND pin*. Osnovna ideja *I2C/TWI* komunikacije je da jedan mikrokontroler radi kao "gospodar" (*master*) koji provodi komunikaciju, dok drugi mikrokontroler radi kao "sluga" (*slave*) iz kojeg se vade podatci ili upisuju ovisno o odluci "gospodara". U ovom radu ulogu "gospodara" vrši mikrokontroler koji upravlja samim mobilnim robotom, dok ulogu "sluge" poprima mikrokontroler koji prikuplja podatke iz ultrazvučnih senzora.

Način na koji komunikacija između ta dva mikrokontrolera funkcionira može se opisati u nekoliko koraka koji se ponavljaju dok god traje glavni program:

1. U glavnom programu dolazi do potrebe za informacijom iz mikrokontrolera "sluge".
2. Mikrokontroler "gospodar" otvara komunikaciju i šalje zahtjev za podacima na mikrokontroler "slugu".
3. Mikrokontroler "sluga" prekida izvođenje svog programa kako bi poslao zatražene podatke.
4. Nakon što je mikrokontroler "gospodar" dobio traženu informaciju, zatvara komunikaciju.
5. Sluga nastavlja sa radom svog programa.

Kod očitavanja ultrazvučnih senzora taj se postupak treba ponavljati za svaki senzor odvojeno. Također, radi jednostavnijeg izvođenja komunikacije decimalne vrijednosti dobivenih udaljenosti je potrebno pretvoriti u prirodan broj (*integer*) prije nego što se šalju u mikrokontroler "gospodar". Te će se vrijednosti zatim u mikrokontroleru "gospodaru" ponovo pretvoriti u decimalni oblik.

Sama biblioteka *Wire* je izrađena za 8-bitne mikrokontrolere (*integer* reprezentiran preko 16 *bit-a*), pa kada bi u glavnom programu pokušali poslanu vrijednost očitati kao prirodan broj (*integer*) dobili bi loše vrijednosti (*integer* u *Arduino Due*-u je reprezentiran sa 32 bita).

Kako bi se taj problem rješio u mikrokontroleru "slugi" je potrebno vrijednost senzora razdjeliti na 4 bajta. Program "gospodar" šalje zahtjev za ta 4 bajta koja zatim rekonstruira u stvarnu vrijednost. Metoda pretvaranja *integer*-a u bajtove i obrnuto se izvodi pomoću *bit shiftinga*.

#### 4.5 Rad ultrazvučnih senzora

Sam način upravljanja ultrazvučnim sensorima *HC-SR04* je vrlo jednostavan i može se opisati u nekoliko koraka:

1. Mikrokontroler pali *TRIG pin* i generira ultrazvučni signal frekvencije 40 kHz.
2. Mikrokontroler zatvara *TRIG pin*, otvara *ECHO pin* i počinje štopati vrijeme.
3. Nakon što se signal vrati zatvara se *ECHO pin* i zabilježava se vrijeme.
4. Određuje se udaljenost prepreke od senzora preko (29).

Ovaj postupak se ponavlja za svaki senzor jedan iza drugog.

Kako će u ovom radu mobilni robot koristiti podatke iz 5 ultrazvučnih senzora (3 prednja i 2 stražnja), napravljena su dva eksperimenta u kojem jedan mikrokontroler (sluga) mjeri vrijednosti senzora, dok drugi mikrokontroler (gospodar) u određenim intervalima čita senzore. Prvi eksperiment testira mogu li se u intervalu od 0.1 sekunde mjenjati vrijednosti na ultrazvučnim sensorima. Drugi eksperiment simulira rad senzora koji će se koristiti u reaktivnoj navigaciji.

##### 4.5.1 Eksperiment 1

Za provođenje eksperimenta izrađena su dva programa. Prvi program "Master" radi tako prikuplja vrijednosti senzora svakih 0,1 sekundi i ispisuje ih na ekran. Drugi program "Slave" radi tako da mjeri senzore i, nakon što je izvršio mjerenja, ispiše očitane vrijednosti na ekran. Pošto sučelje *Arduino IDE* može u jednom trenutku imati otvoren samo jedan serijski kanal, a za provođenje ovog eksperimenta je potrebno da su otvorena dva kanala serijske komunikacije (jedan za "Master" drugi za "Slave"), korišten je program *CoolTerm* koji može otvarati više serijskih kanala odjednom.

Rezultati eksperimenta su prikazani na slici [Slika 23]. Gornji ekran na slici prikazuje podatke koje ispisuje mikrokontroler "Slave", dok donji program prikazuje podatke koje ispisuje mikrokontroler "Master". Iz slike se vidi da "Slave" i "Master" ne ispisuju u istom trenutku iste vrijednosti. Također se vidi da mikrokontroler "Slave" puno rijeđe ispisuje podatke od



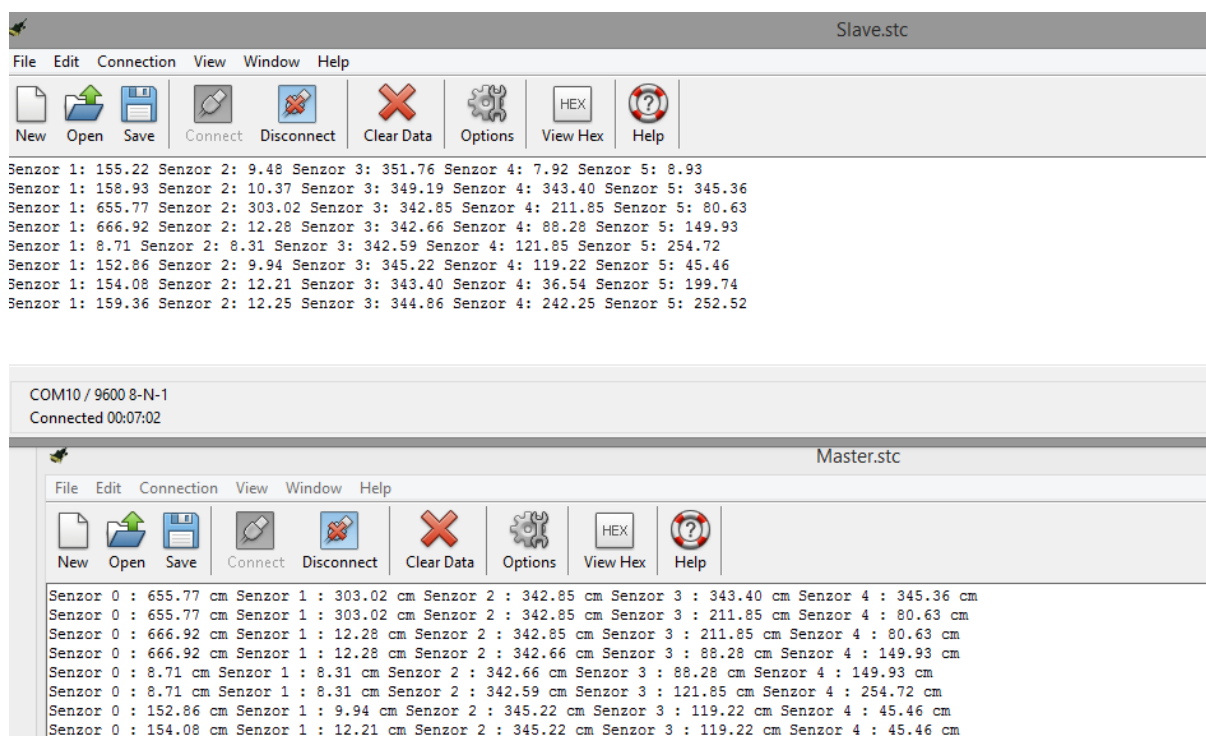
mikrokontrolera "Master". Sporije pisanje mikrokontrolera "Slave" se može objasniti iz dva razloga.

Prvi razlog je to što mikrokontroler "Slave" mora prije nego što ponovo ispiše rezultate ponoviti postupak mjerenja za svih 5 senzora. To znači da, u slučaju da se jedan od senzora nalazi daleko od prepreke, može doći do zastoja u mjerenju senzora zbog čekanja da se udaljeni val vrati. Pošto se mobilni robot konstantno giba, to bi moglo dovesti do sudara s preprekom.

Drugi razlog je taj što je funkcija za serijsku komunikaciju, potrebna za ispis očitavanja senzora na ekran, vremenski zahtjevnija što dodatno usporava mjerenje.

Iz provedenog eksperimenta može se izvući nekoliko zapažanja:

- Pošto ultrazvučni senzori mjere jedan iza drugog, može doći do prespore reakcije senzora na prepreku (pogotovo kod pokretnih prepreka).
- Mikrokontroler "Master" je nekada znao očitati vrijednosti senzora koje je mikrokontroler "Slave" tek naknadno ispisao, što dokazuje kako funkcija `Wire()` prekida rad senzora kako bi poslao dotad izmjerene vrijednosti (potencijalno problem, ako se mjerenje prekine tokom čekanja na povratak signala).
- Serijska komunikacija je vremenski zahtjevan proces koji se treba što rijetko koristiti u programu.



**Slika 23. Rezultati eksperimenta sa ultrazvučnim sensorima**

Kako bi se donekle sanirao problem lošeg iščitavanja zbog udaljenosti jednog od senzora od prepreke, moguće je malo promjeniti program. Pošto znamo da se mobilni robot, dok se giba naprijed, ne može sudariti sa preprekom koja se nalazi iza njega i obrnuto, kako bi se ubrzalo mjerenje senzora moguće je "isključiti" senzore koji mjere suprotno od smjera kojim se mobilni robot giba.

To se može napraviti preko funkcije pomoću 2 *external interrupt*-a koji bi radili kao prekidači. Jedan bi služio da, kada se mobilni robot giba naprijed, isključi stražnje senzore, dok bi drugi isključivao prednje senzore dok se robot giba unatrag.

Iako eksperiment nije potpuno precizan, koristan je za donošenje zaključaka pomoću kojih je izrađen konačan program.

#### 4.5.2 Eksperiment 2

U reaktivnoj navigaciji, radi pojednostavljenja je postavljeno da mobilni robot mjeri sa samo prednja tri ultrazvučna senzora. Stražnja dva ultrazvučna senzora rade samo dok se mobilni robot giba unatrag i služe da zaustave program ako, gibajući se unatrag, dođe preblizu prepreci (bliže od 4 cm).

U prethodnom eksperimentu je objašnjeno da će se stražnji senzori paliti i gasiti preko *external interrupt*-a, ali također valja napomenuti da će se koristiti još jedan *external interrupt* koji će javiti da je prepreka preblizu (u reaktivnoj navigaciji on će zaustaviti program i javiti da je došlo do sudara).

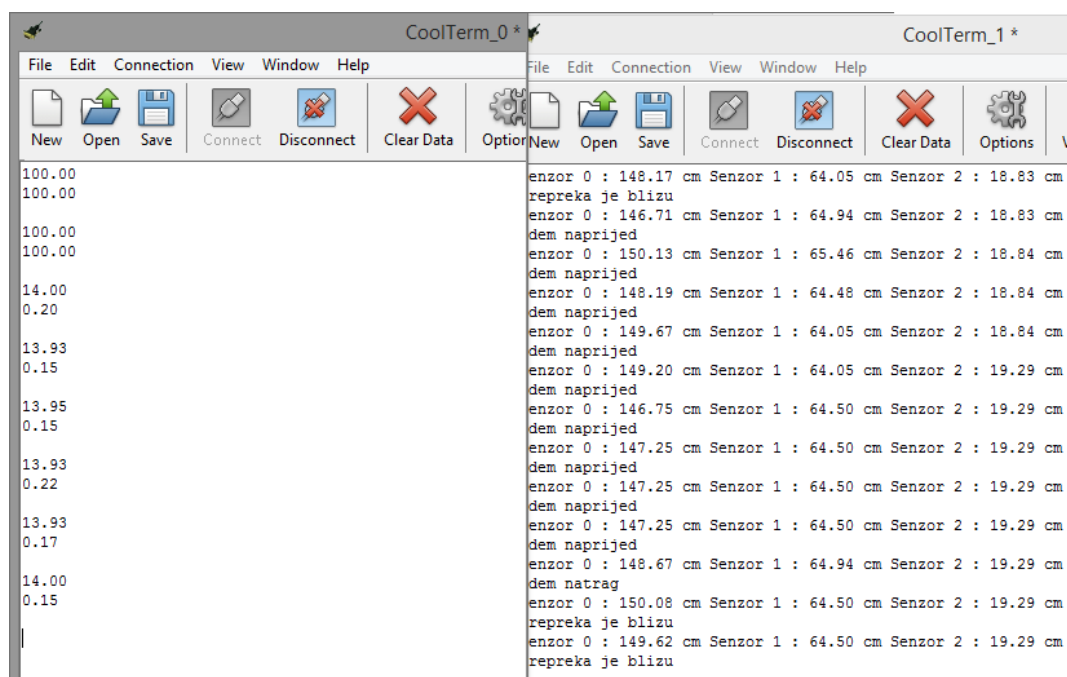
Kao i u prethodnom eksperimentu testiranje je provedeno uz pomoć dva mikrokontrolera "Master" i "Slave".

Mikrokontroler "Master" u intervalima od 5 sekundi mijenja način gibanja naprijed i natrag i ovisno o gibanju pali i gasi stražnje senzore. Tokom tog intervala svakih 0.5 sekundi čita prednja tri ultrazvučna senzora i ispisuje njihove rezultate. Također ovisno o ostvarenom uvjetu ispisuje jednu od tri rečenica:

- Ako se giba naprijed ispisuje se "Idem naprijed"
- Ako se giba unatrag, a stražnji senzori ne očitavaju ispod 4 cm ispisuje se "Idem natrag"
- U slučaju da bilo koji od stražnjih ultrazvučnih senzora izmjerio ispod 4 cm ispisuje se "Prepreka je blizu"

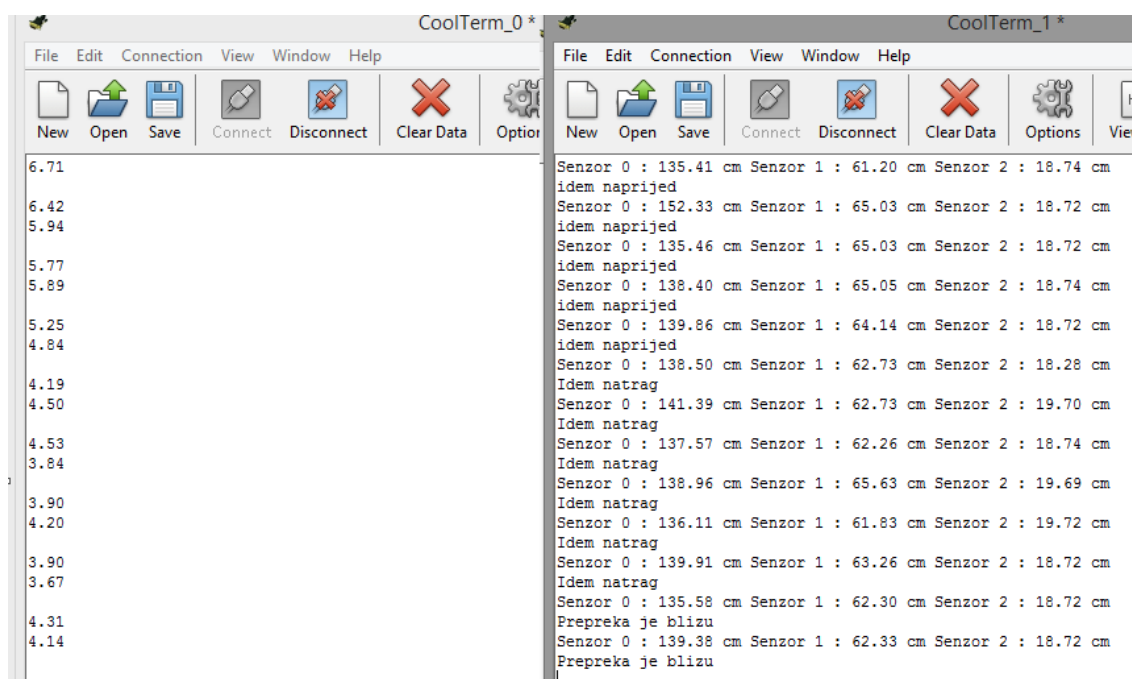
Mikrokontroler "Slave" prvo mjeri prva tri senzora, a ako se giba natrag očitava stražnja dva senzora. Nakon što je očitao stražnja dva senzora vrši se provjera je li jedan od njih očitao ispod 4 cm. Ako je dobio potvrđan odgovor, šalje napon na "Master", koji kao rezultat mijenja ispis na "Prepreka je blizu" u slučaju da se giba unatrag. Ako dobije negativan odgovor stavlja napon na nisko stanje. Također valja napomenuti da u trenucima kada "Master" ide naprijed vrijednosti stražnjih senzora postavljaju se na 100 (kako bi bilo pregledno da stražnji senzori ne rade).

Pred stražnje senzore je približavana bilježnica s tvrdim koricama koja je simulirala prepreku. Prvi rezultati [Slika 24] su ukazali na jedan problem. Iako je bilježnica bila jednako udaljena od oba senzora, jedan senzor je pokazivao dobar rezultat, dok je drugi senzor mjerio jako male vrijednosti (0,15 i slično). Pošto su lijevi i desni stražnji ultrazvučni senzori iste orijentacije i postavljeni relativno blizu jedan uz drugog [Slika 6], dogodilo se je da je drugi senzor počeo sa mjerenjem dok je ultrazvučni val napravljen od prethodnog senzora još uvijek prolazio blizu senzora. To je rezultiralo time da je ultrazvučni registrirao prethodni ultrazvučni val kao svoj i time izračunao pogrešnu udaljenost (što je razlog zašto se ultrazvučni senzori pokreću jedan iza drugog).



**Slika 24. Problem očitavanja stražnjih ultrazvučnih senzora**

Rješenje tog problema je uvođenje kašnjenja između svakog novog mjerenja od 5 ms. [Slika 25] prikazuje rad stražnjih senzora nakon što se uvelo navedeno kašnjenje.



**Slika 25. Ispravan rad stražnjih ultrazvučnih senzora**

Sa uvođenjem kašnjenja počeka su se javljati preciznija mjerenja, no također su se ukazala dva problema.

Prvi problem je taj da, iako je prepreka stajala na istoj udaljenosti, javljale su se razlike u očitavanju senzora u pojedinim ciklusima. Ta razlika je relativno velika na malim udaljenostima, što može uvesti nestabilnost u izračunu referentne trajektorije. Pošto je ultrazvuk val, on se neće uvijek isto odbiti od prepreke i zato senzor neće uvijek dobiti iste rezultate. To može biti jako velik problem kada prepreka nije ravno u odnosu na senzor, jer se može dogoditi da se val odbije pod kutem koji se ne vrati uz senzor, što može dovesti do krivog mjerenja. Sličan bi se problem mogao dogoditi u uskim prostorima kada bi se valovi počeli odbijati od nasuprotnih zidova.

Drugi problem je u tome što se provjera udaljenosti stražnjih senzora odvija tek nakon što su obavljena sva mjerenja, pa bi se u nekim slučajevima moglo dogoditi da senzor ne uspije na vrijeme informirati mobilni robot da se treba zaustaviti.

Ta dva problema nam ukazuju da ultrazvučni senzor nije idealan za precizno mjerenje udaljenosti mobilnog robota od prepreke, pa je zato potrebno postaviti relativno veliki radius na kojem dijeluje repulzivna sila (28).

## 5. REAKTIVNA NAVIGACIJA

Prvi korak rada se sastojao od izrade mobilnog robota na kojem bi se provela praktična izvedba reaktivne navigacije metodom potencijalnih polja.

Cijeli tijek izrade programa za reaktivnu navigaciju se može se podijeliti na tri koraka:

1. Izrada PID regulatora za pokretanje kotača mobilnog robota,
2. Izrada analitičkog neizrazitog regulatora,
3. Reaktivna navigacija preko metode potencijalnih polja.

U prvom i drugom koraku se može raditi sa samo jednim mikrokontrolerom, dok se u zadnjem koraku koriste ultrazvučni senzori, pa je potrebno koristiti 2 mikrokontrolera.

U sklopu rada su izrađena su 3 programa koji se koncentriraju na svaki od ta tri koraka. Iako svaki novi program sadrži elemente prethodnog programa, svaki od ta tri programa predstavljaju probleme koji se moraju detaljnije proučavati sami za sebe.

Prvi program se bavi namještanjem parametara PID regulatora koji su potrebni za uspješno upravljanje mobilnim robotom (preduvjet za analitički neizraziti regulator).

Drugi program se bavi namještanjem parametara analitičkog neizrazitog regulatora za uspješno praćenje referentne trajektorije (preduvjet za reaktivnu navigaciju).

Treći program se bavi namještanjem parametara u metodi potencijalnih polja za uspješno izbjegavanje prepreke (cilj rada).

U ovom poglavlju će biti opisani način rada svakog od pojedinih programa, kao i eksperimenti provedeni na mobilnom robotu. Neka predložena rješenja iz prethodnih programa nisu iskorištena u sljedećim programima zbog vremenskog ograničenja izrade rada i lošeg *hardware*-a. Koraci nisu potpuno optimizirani, no eksperimenti su izvedeni tako da provjere ispravnost samih algoritama.

Prije opisa samih programa potrebno je, radi bolje predodžbe, navesti popis spojenih pin-ova na upravljačkom mikrokontroleru [Tablica 7] i mikrokontroleru sa ultrazvučnim sensorima [Tablica 8].

Tablica 7. Popis *pin*-ova na upravljačkom mikrokontroleru

Uređaj	<i>Pin</i> na uređaju	<i>Pin</i> na mikrokontroleru
H-most	<i>ENA</i>	2
	<i>IN1</i>	3
	<i>IN2</i>	4
	<i>IN3</i>	7
	<i>IN4</i>	8
	<i>ENB</i>	6
Lijevi enkoder	<i>OUT</i>	50
Desni enkoder	<i>OUT</i>	52
Xbee	<i>RST</i>	<i>RESET</i>
	<i>RX</i>	15 ( <i>RX3</i> )
	<i>TX</i>	14 ( <i>TX3</i> )
<i>Wire</i> protokol	20 ( <i>SDA</i> )	20 ( <i>SDA</i> )
	21 ( <i>SCL</i> )	21 ( <i>SCL</i> )
<i>External interrupt-i</i>	31	31
	33	33
	35	35
	37	37

Tablica 8. Popis *pin*-ova na mikrokontroleru sa ultrazvučnim senzrima

Uređaj	<i>Pin</i> na uređaju	<i>Pin</i> na mikrokontroleru
Lijevi prednji senzor	<i>ECHO</i>	2
	<i>TRIG</i>	3
Srednji prednji senzor	<i>ECHO</i>	4
	<i>TRIG</i>	5
Desni prednji senzor	<i>ECHO</i>	6
	<i>TRIG</i>	7

Lijevi stražnji senzor	<i>ECHO</i>	8
	<i>TRIG</i>	9
Desni stražnji senzor	<i>ECHO</i>	10
	<i>TRIG</i>	11
<i>Wire</i> protokol	20 ( <i>SDA</i> )	20 ( <i>SDA</i> )
	21 ( <i>SCL</i> )	21 ( <i>SCL</i> )
<i>External interrupt-i</i>	31	31
	33	33
	35	35
	37	37

## 5.1 Testiranje parametara PID regulatora

Kako bi mobilni robot uspješno postizao željene brzine potrebno je regulirati brzinu okretaja kotača. U tu svrhu izrađen je program za testiranje parametara PID regulatora za dva načina pobude ulazne (tražene) brzine:

1. Promjeniva *step* pobuda u kojoj korisnik određuje prvi interval, dok su ostali predodređeni u mikrokontroleru.
2. Sinusoidna promjena ulazne brzine u kojem korisnik bira amplitudu sinusoide.

Za testiranje parametara PID regulatora izrađena su dva programa. Prvi program je izrađen u Arduino IDE (mikrokontroler), dok je drugi izrađen u *Python*-u.

### 5.1.1 Mikrokontroler

Na mikorkontroleru se nalazi program koji regulira brzinu vrtnje kotača mobilnog robota preko PID regulatora čije parametre određuje korisnik.

Sam program nakon početne inicijalizacije čeka podatke iz računala i, ovinso o tome, provodi odgovarajuće akcije.

Nakon što od računala dobije naredbu za početak rada PID regulatora, on počinje sa radom. PID regulator se izvodi 12 sekundi, te u periodima od 0,1 sekunde izvodi regulaciju kotača (funkcija *Regulator()*).

Funkcija *Regulator()* se može opisati u nekoliko koraka:

- Prvo se u ovisnosti o predznaku ulazne brzine za lijevi i desni kotač određuje smjer vrtnje kotača (H-most).
- Upravljački signali se zatim provode kroz filter prvog reda (biti će objašnjeno kasnije)
- Nakon toga dolazi do iščitavanja lijevog i desnog enkodera
- Iz dobivenih vrijednosti se određuje brzina lijevog i desnog kotača. Ovdje valja napomenuti, pošto je enkoder korišten u radu jednofazan, on može samo odrediti brzinu vrtnje, no ne i njen smjer. Smjer brzine pretpostavljamo da je jednak smjeru upravljačke varijable.
- Nakon što su izračunate brzine vrtnje resetira se brojač lijevog i desnog enkodera.
- Određuje se pogreška između upravljačke veličine i izmjerene veličine
- Iz izračunate pogreške se određuju  $P$ ,  $D$  i  $I$  članovi za lijevi i desni kotač
- $I$  član se uspoređuje sa svojom maksimalnom vrijednošću  $I_{max}$ . U slučaju da je  $I$  član veći od  $I_{max}$  on poprima vrijednost  $I_{max}$ .
- Na osnovu vrijednosti  $P$ ,  $D$  i  $I$  članova mijenja se upravljački signal lijevog i desnog kotača
- Prije nego što se upravljački signal šalje na H-most, provodi se limitacija PWM signala (kako ne bi se poslala vrijednost iznad 255, što bi dovelo do pogreške).
- Dobiveni upravljački PWM signali se šalju na H-most, koji preko njih pokreće elektromotore.

Ovisno o načinu generiranja ulaznih brzina, postoje dvije verzije promjene ulaznih brzina.

Za promjenjivu *step* pobudu ulazne brzine lijevog i desnog kotača se mijenjaju svakih 3 sekunde.

Za sinusoidni odziv ulazne brzine, nakon izvršenog regulacijskog kruga dolazi do promjene ulazne brzine preko sinusne funkcije.

Nakon što je prošlo 12 sekundi, mikrokontroler šalje na računalo izračunate vrijednosti u pojedinom periodu za:

- Brzinu vrtnje lijevog kotača  $v_L$ ,
- Brzinu vrtnje desnog kotača  $v_R$ ,
- Linearnu brzinu mobilnog robota  $v$ ,
- Lijeve upravljački signal  $p_L$ ,
- Desne upravljački signal  $p_R$ ,
- Lijevu ulaznu brzinu  $u_L$ ,



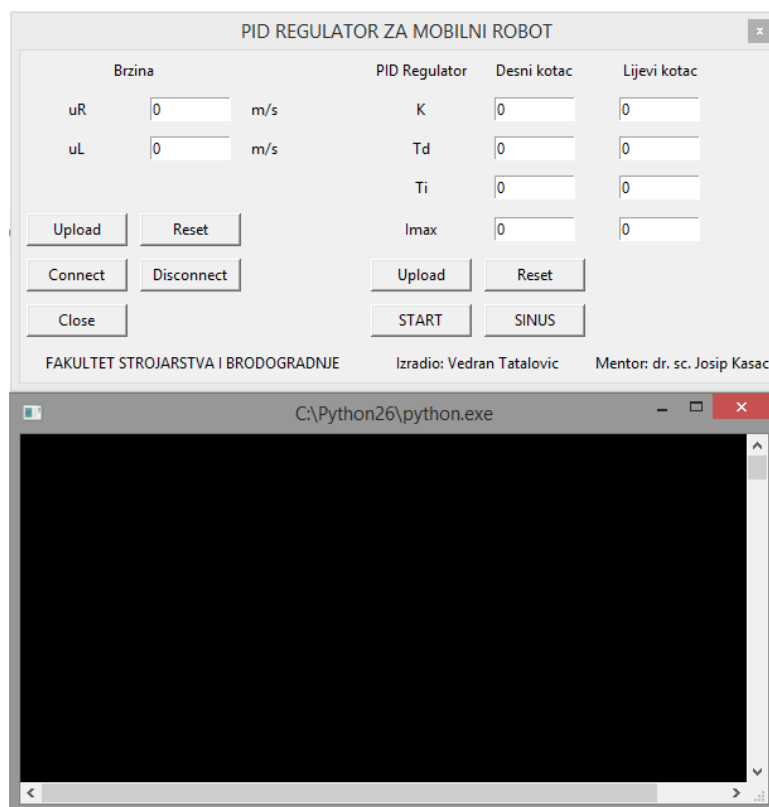
- Desnu ulaznu brzinu  $u_R$ .

Kao što je navedeno u prošlom poglavlju serijska komunikacija je vremenski zahtjevan proces, pa bi se slanje podataka moralo događati nakon izvršenja rada programa. No pošto se izmjerene vrijednosti moraju spremati u niz podataka, to bi kod dugog rada programa moglo dovesti do preljevanja stoga usred iskorištenja radne memorije. Zato se u ostalim programima serijska komunikacija vodi tokom trajanja programa.

Također valja napomenuti da je kod izračuna brzina vrtnje lijevog i desnog kotača uveden uvjet koji govori da su te brzine u početnom trenutku uvijek 0 (znalo je doći do pojave velikih brzina potencijalno uzrokovanih zbog  $\varepsilon$ ). Taj je uvjet postavljen u svim uvjetima gdje se pojavljuje  $\varepsilon$ .

### 5.1.2 Računalo

U sklopu rada za namještnje parametara i ispis rezultata u *Python*-u je izrađeno grafičko sučelje (GUI). Rezultati se ispisuju u grafičkom prikazu radi bolje razlučivosti. Sam izgled sučelja prikazuje [Slika 26].



Slika 26. Grafičko sučelje za testiranje PID regulatora

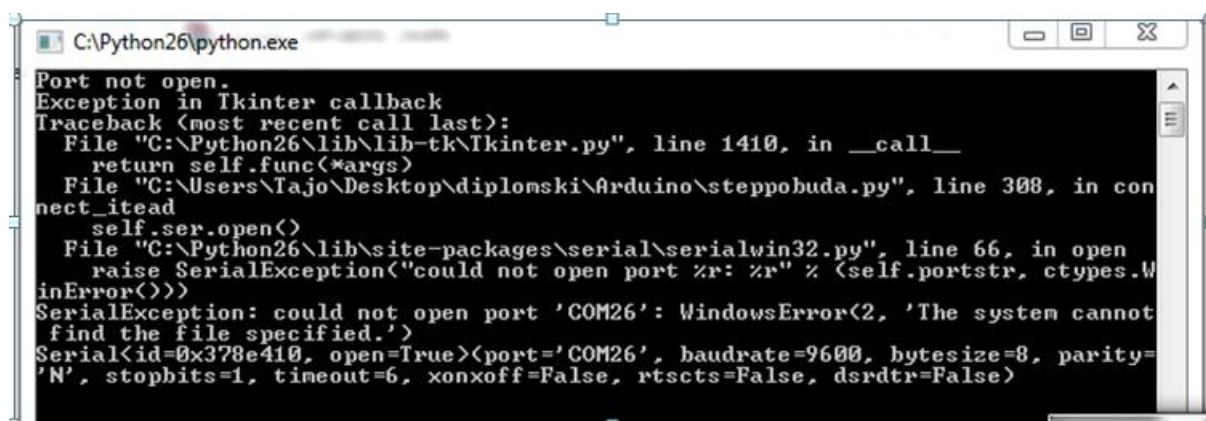
Grafičko sučelje se sastoji od radnog dijela u koji služi za rad sa mikrokontrolerom i prozora u kojem se ispisuju podaci koje prima sa mobilnog robota, kao i još par dodatnih informacija važih za rad samog programa.

### 5.1.2.1 Uputstva za korištenje

Pritiskom na gumb „Connect“ otvara se komunikacija između mikrokontrolera i računala. Ako korisnik pokuša pritisnuti bilo koji drugi gumb (osim "Close") prije uspostavljanja komunikacije, na zaslonu će se ispisati "Port not open". U slučaju da nije moguće naći željeni uređaj, program će javiti grešku. Ako je komunikacija uspješna, program će obavijestiti korisnika. [Slika 27] prikazuje sva tri slučaja s time da najdonja dva reda pokazuju uspješnu komunikaciju. Važno je za napomenuti ako se nakon pritiska tipke "Connect" brzo stisne neka druga tipka, može se dogoditi da mikrokontroler ne registrira odabranu naredbu.

Tipka "Disconnect" zatvara komunikaciju između računala i mikrokontrolera.

Tipka "Close" izlazi iz programa. To je ujedno i jedini način da se izađe iz programa (osim, naravno, kod prisilnog zatvaranja).



```

C:\Python26\python.exe
Port not open.
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python26\lib\lib-tk\Tkinter.py", line 1410, in __call__
    return self.func(*args)
  File "C:\Users\Tajo\Desktop\diplomski\Arduino\steppobuda.py", line 308, in connect_itead
    self.ser.open()
  File "C:\Python26\lib\site-packages\serialwin32.py", line 66, in open
    raise SerialException("could not open port %r: %r" % (self.portstr, ctypes.W
inError()))
SerialException: could not open port 'COM26': WindowsError(2, 'The system cannot
find the file specified.')
Serial<id=0x378e410, open=True>(port='COM26', baudrate=9600, bytesize=8, parity=
'N', stopbits=1, timeout=6, xonxoff=False, rtscts=False, dsrdtr=False)

```

Slika 27. Uspostava komunikacije

Na lijevoj strani sučelja u kućice korisnik upisuje željene početne ulazne brzine lijevog i desnog kotača. Pritiskom na tipku "Upload" koja se nalazi ispod "Brzina" te se brzine šalju na mikrokontroler. Sučelje će ispisati niz koji je poslalo, kao i povratnu informaciju iz mikrokontrolera. Valja napomenuti kako je program izrađen da se ulazne brzine mogu kretati između -1 i 1 m/s pa vrijednosti veće od njih tretira kao jednu od tih graničnih vrijednosti.

Pritiskom na tipku "Reset" vrijednosti u kućicama sučelja se postavljaju u 0 (no ne i na mikrokontroleru).

Sa desne strane se nalazi prostor za upisivanje parametara regulatora i pripadajući gumbi (parametri u jednadžbama (9) do (14)).

Pritiskom na tipku "START" pokreće se rad PID regulatora za promjenjivu *Step* pobudu ulaznih brzina.

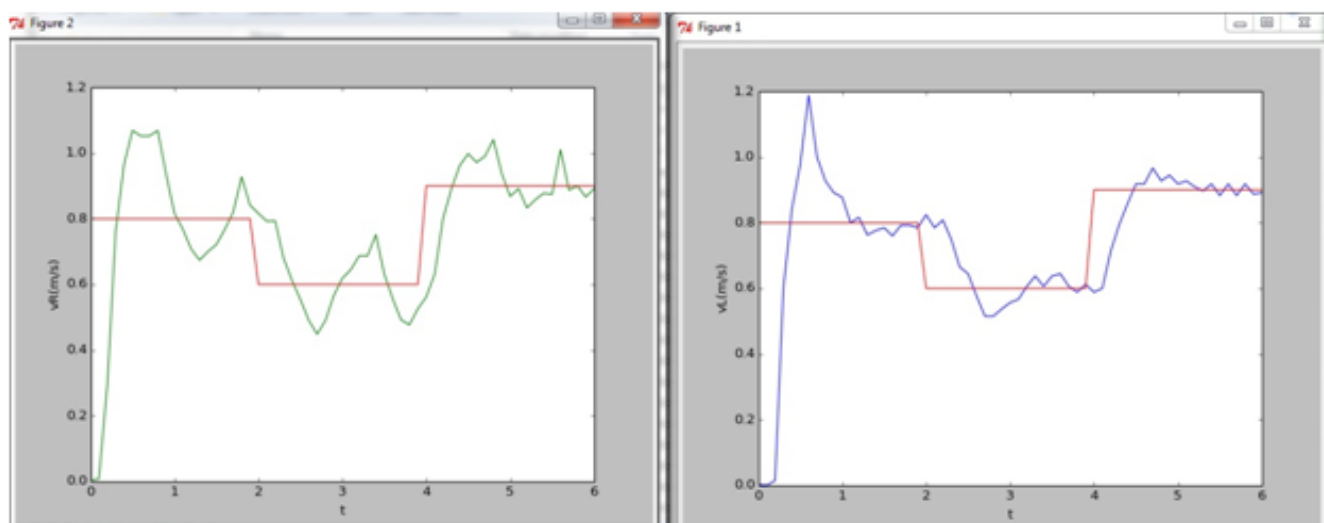
Pritiskom na tipku "SINUS" pokreće se rad PID regulatora za sinusoidalnu promjenu ulaznih brzina.

Nakon što se odabrala jedna od prethodne dvije tipke i nakon što je mikrokontroler završio sa radom, računalo ispisuje grafičke prikaze brzina i upravljačkih PWM signala.

### 5.1.3 Izvedena testiranja

Svrha ovog programa je određivanje parametara regulatora pomoću kojih bi se uspješno provelo upravljanje lijevim i desnim kotačem elektromotora. Uspješno upravljanje kotačima je preduvjet za uspješno sljeđenje referentne trajektorije preko analitičkog neizrazitog regulatora.

Kao najbolji vrsta regulatora se pokazao PD regulator čiji je odziv prikazan na slici [Slika 28] (više o samom odabiru tipa regulatora može se proučiti u [19]).



**Slika 28. Odziv PD regulatora**

Iz danog odziva mogu se izvaditi neki vrlo važni zaključci:

- Kod klasičnog PD regulatora će kod porasta brzine najčešće doći do preskakanja tražene vrijednosti.
- Jednaki parametri PD regulatora na različitim brzinama djeluju drugačije na sustav i mogu dovesti do oscilacija.

- Enkoderi, zbog niske rezolucije i zbog neizbježne pojave šuma u signalu, znaju uvesti grešku, te za isti upravljački PWM signal mogu izmjeriti različite brzine (test je izveden u praznom hodu kotača kako bi se eliminirale razlike u brzini zbog proklizavanja kotača)
- Lijevi i desni elektromotor imaju drugačiji odziv za iste parametre PD regulatora (nezgodno za regulaciju).

Ti problemi također ukazuju da elektromotori rade daleko od idealnog teoretskog rada (opisanog u potpoglavlju 2.2), pa bi mobilni robot reguliran klasičnim PD regulatorom mogao imati velikih poteškoća u ostvarivanju uspješnog gibanja.

Jedan način za rješavanje tog problema bi bio izrada adaptivnog regulatora koji bi mjenjao parametre PD regulatora ovisno o promjeni ulazne brzine.

No, zbog vremenskog ograničenja u izradi rada, odlučeno je pokušati poboljšati odziv PD regulatora sa jednostavnijim metodama.

U nastavku potpoglavlja biti će opisan proces poboljšanja odziva klasičnog PD regulatora.

Također valja napomenuti da enkoder u početku testiranja, radi povećanja rezolucije, enkoder mjeri u režimu rada *CHANGE* (opisan u potpoglavlju 4.1).

Poboljšanja koja su predložena su:

- Prefiltracija ulaznih brzina lijevog i desnog kotača.
- Uvođenje minimalne vrijednosti upravljačkog PWM signala kada ulazna brzina nije 0.
- Mjenjanje načina očitavanja enkodera na režim rada *RISING* (potpoglavlje 4.1).

Parametri PD regulatora korišteni tokom izrade poboljšanja (kao i u nastavku rada) su:

$$K_{pL} = K_{pR} = 45, T_{dL} = T_{dR} = 1,5 \quad (53)$$

Napomena: Iako su parametri PD regulatora, čiji je odziv prikazan na slici [Slika 28], bili  $K_{pL} = K_{pR} = 60, T_{dL} = T_{dR} = 0,2$ , konačno odabrani parametri (53) su dali bolje odzive.

### 5.1.3.1 Prefiltracija ulaznog signala

Kod velikih promjena ulaznih brzina, u klasičnom PD regulatoru najčešće dolazi do prekoračenja tražene brzine [Slika 28]. Za stabilno gibanje mobilnog robota je od velike važnosti da se to prekoračenje eliminiira ili barem minimizira.

Prekoračenje je rezultat jako velike regulacijske pogreške (8), koja dovodi do velike promjene upravljačkog PWM signala (najizraženije kod pokretanja mobilnog robota).

Kako bi se izbjegle velike regulacijske pogreške, moguće je ulazne brzine lijevog i desnog kotača provesti kroz predfilter prvog reda. Predfiltracijom se *step* odziv ulazne brzine lijevog i desnog kotača mijenja u odziv P1 člana čija formula u diskretnom sustavu glasi:

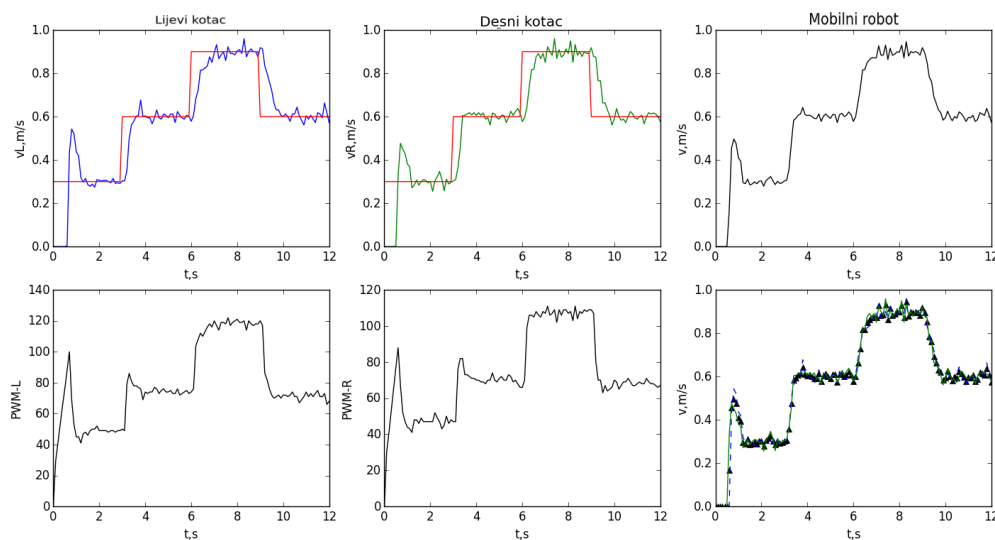
$$\begin{aligned} u_{fL}(NT) &= u_{fL}((N-1)T) + (u_L(NT) - u_{fL}((N-1)T))e^{-k_f N_{fL} T} \\ u_{fR}(NT) &= u_{fR}((N-1)T) + (u_R(NT) - u_{fR}((N-1)T))e^{-k_f N_{fR} T} \end{aligned} \quad (54)$$

gdje su  $N_{fL}$  i  $N_{fR}$  koraci filtracije koji se resetiraju svaki put kada dođe do promjene ulazne brzine. Period diskretnog sustava  $T$  je jednak 0,1, koeficijent nagiba eksponencijalne funkcije  $k_f$  je jednak 10.

Nakon predfiltracije ulaznih brzina pogreške lijevog i desnog kotača u diskretnom sustavu postaju:

$$\begin{aligned} e_L(NT) &= u_{fL}(NT) - v_L(NT) \\ e_R(NT) &= u_{fR}(NT) - v_R(NT) \end{aligned} \quad (55)$$

Primjer odziva PD regulatora nakon uvođenja predfiltracije ulaznih brzina dan je na slici [Slika 29].



**Slika 29. Odziv regulatora sa predfiltracijom ulaznih brzina (Step)**

Iz dobivenih odziva se vidi kako predfiltracijom ulaznih brzina dolazi do smanjenja preskoka.

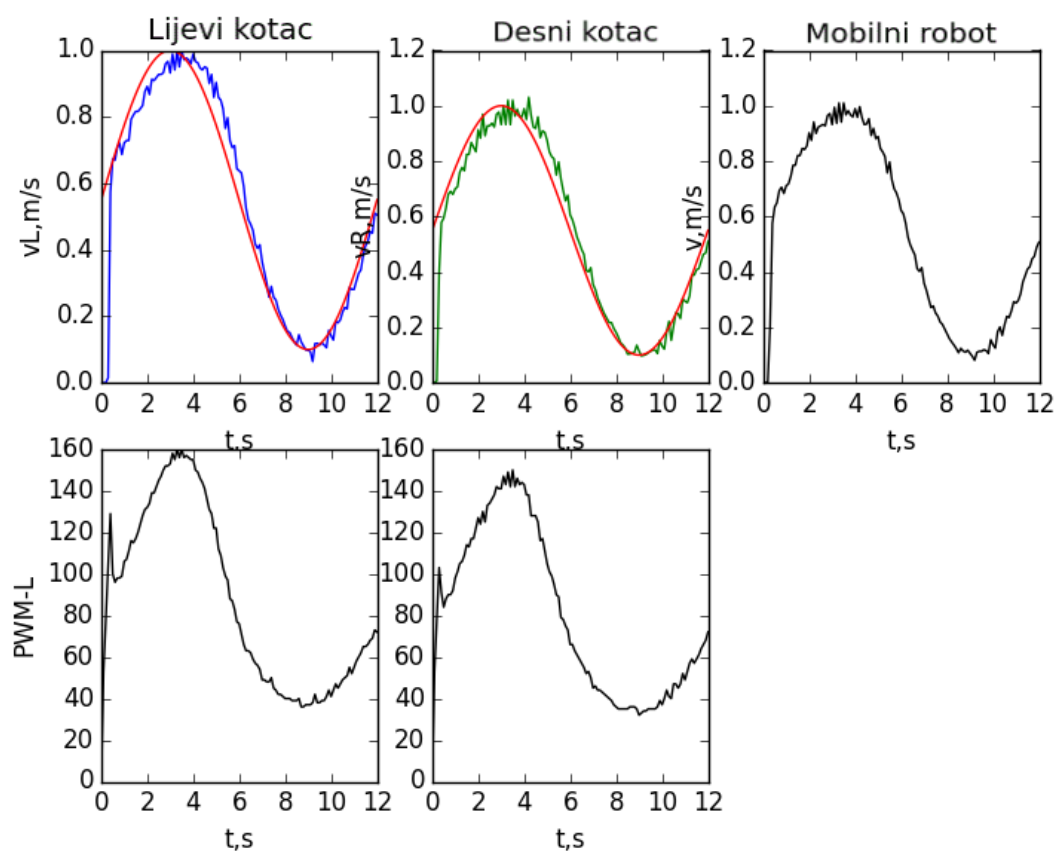
Valja napomenuti kako je pri pokretanju motora preskok neizbježan kod brzina čiji je upravljački PWM signal (napon na elektromotrima) manji od upravljačkog PWM signala (napona na elektromotrima) potrebnog za savladavanje pokretnog momenta.

Napomena: Na slici [Slika 29] se vidi kako su za postizanje istih ulaznih brzina lijevog i desnog kotača potrebni drugačiji upravljački PWM signali. To je za regulaciju kotača velik problem.

Predfiltraciju također valja isprobati za sinusoidalnu promjenu ulaznih brzina lijevog i desnog kotača (u idealnom slučaju analitički neizraziti regulator bi trebao davati postepene promjene ulazne brzine, a ne skokovite).

Odziv predfiltriranih ulaznih brzina je prikazan na slici [Slika 30]. Kao što se vidi na slici, kod sinusoidalne promjene je došlo do blagog kašnjenja u odzivu. Razlog tome je što kod promjene ulazne brzine dolazi do resetiranja koraka filtracije  $N_{fL}$  i  $N_{fR}$ , što dovodi do kašnjenja u radu PD regulatora.

Napomena: Na nekim slikama poput [Slika 30] je naziv Y-osi je došao na prethodni graf. Taj se problem u programu rješio.



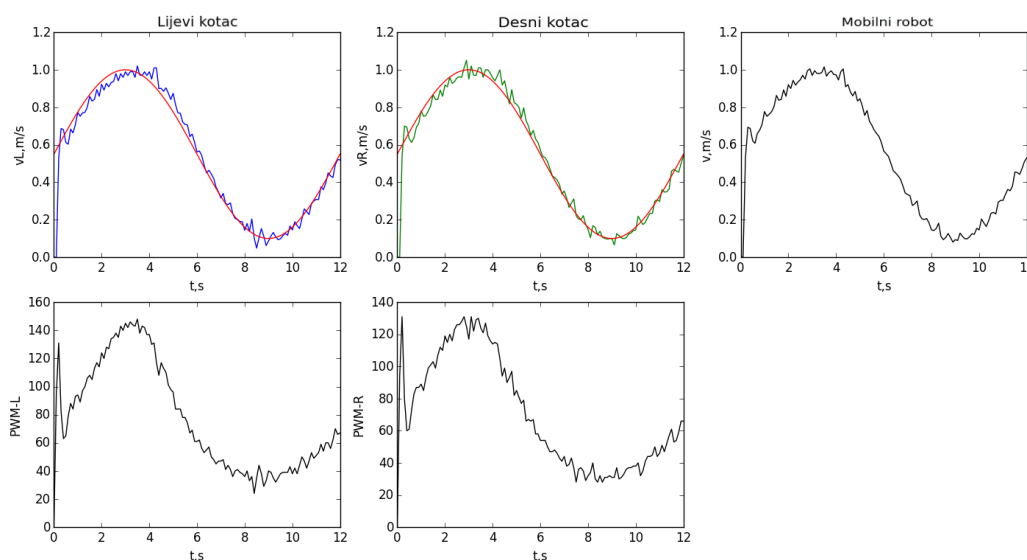
**Slika 30. Odziv regulatora sa predfiltracijom brzina (Sinusoida)**

Kako bi se riješio problem resetiranja koraka filtracije, uvedena je minimalna promjena ulazne brzine lijevog i desnog kotača od 0,1 m/s za resetiranje koraka filtracije  $N_{fL}$  i  $N_{fR}$ .

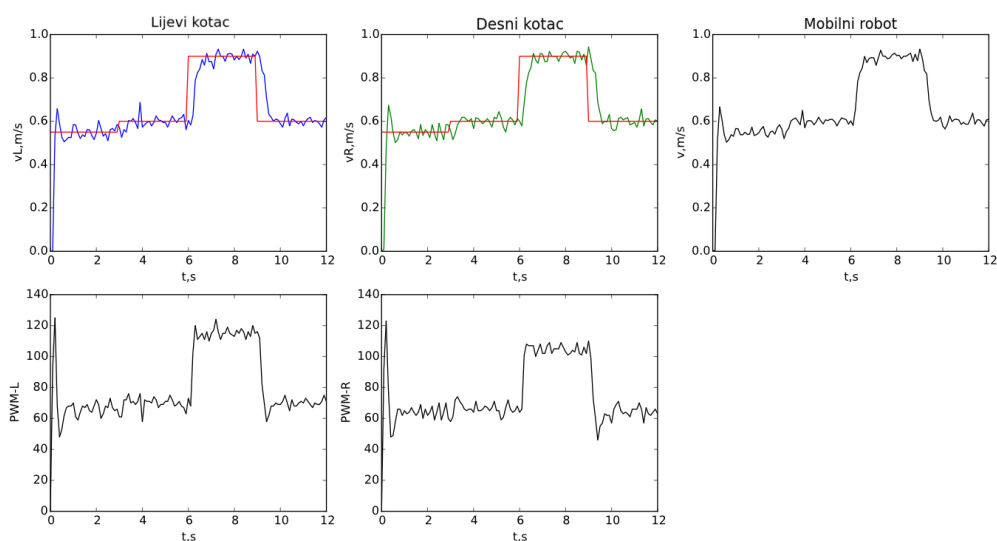
Nakon uvođenja minimalne promjene ulazne brzine došlo je do poboljšanja u odzivu PD regulatora ([Slika 31] za sinusoidalnu, [Slika 32] za *Step* promjenu ulaznih brzina).

Iz provedenih testiranja može se zaključiti da uvođenje filtracije ulaznih brzina lijevog i desnog kotača pozitivno utječe na minimizaciju preskoka PD regulatora. No također valja napomenuti da filtracija ulaznog signala usporava odziv PD regulatora, što nije dobro pošto bi u idealnom slučaju kotači morali u istom trenutku (ili u jako kratkom roku) dostići zadanu ulaznu brzinu.

Unatoč toj manji predfiltracija ulazne brzine lijevog i desnog kotača je korisna u upravljanju mobilnim robotom, pošto je kašnjenje u odzivu manje štetno nego preskakanje ulazne brzine.



**Slika 31. Odziv PD regulatora uz uvođenje minimalne promjene ulaznih brzina za resetiranje koraka filtracije (Sinusoida)**



**Slika 32. Odziv PD regulatora uz uvođenje minimalne promjene ulaznih brzina za restiranje koraka filtracije (*Step*)**

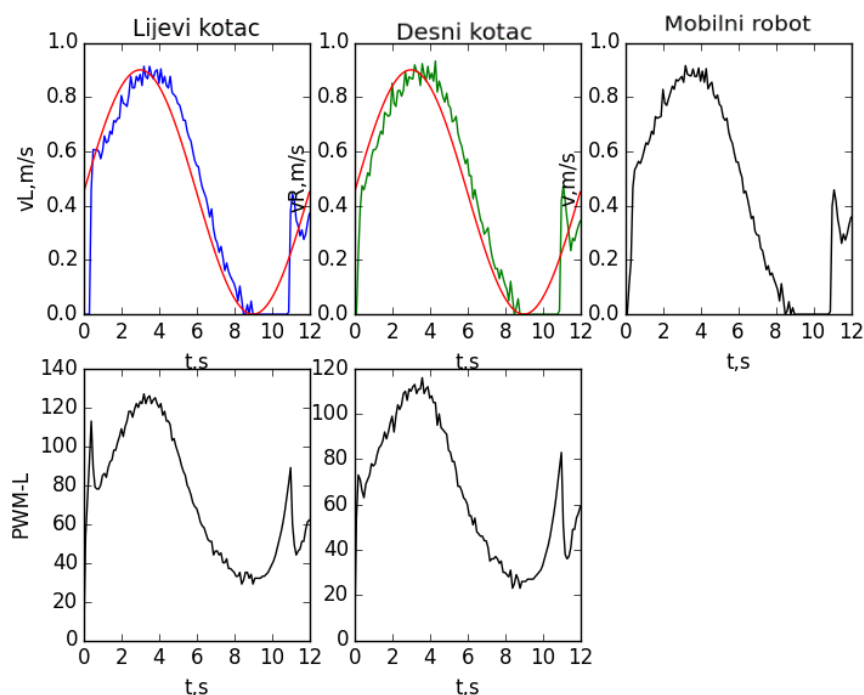
### 5.1.3.2 Problem pokretanja elektromotora

Kod testiranja rada PD regulatora zamijećeno je da se na malim brzinama vrtnje događa jedan veliki problem [Slika 33]. Na malim brzinama došlo je do zaustavljanja vrtnje kotača (upravljački PWM signal je postao manji od minimalnog potrebnog PWM signala za vrtnju elektromotora).

Zaustavljanje kotača mobilnog robota je loše zbog dva razloga:

1. Da bi se motor ponovno pokrenuo upravljački PWM signal mora postati veći od minimalno potrebnog upravljačkog PWM signala za pokretanje motora, što kod malih brzina dovodi do preskoka brzine, a to negativno utječe na stabilnost upravljanja.
2. Kada se kotači ne gibaju, a na elektromotore se šalje napon, struja koja prolazi kroz elektromotore je maksimalna, što može dovesti do oštećenja elektromotora zbog pregrijavanja.



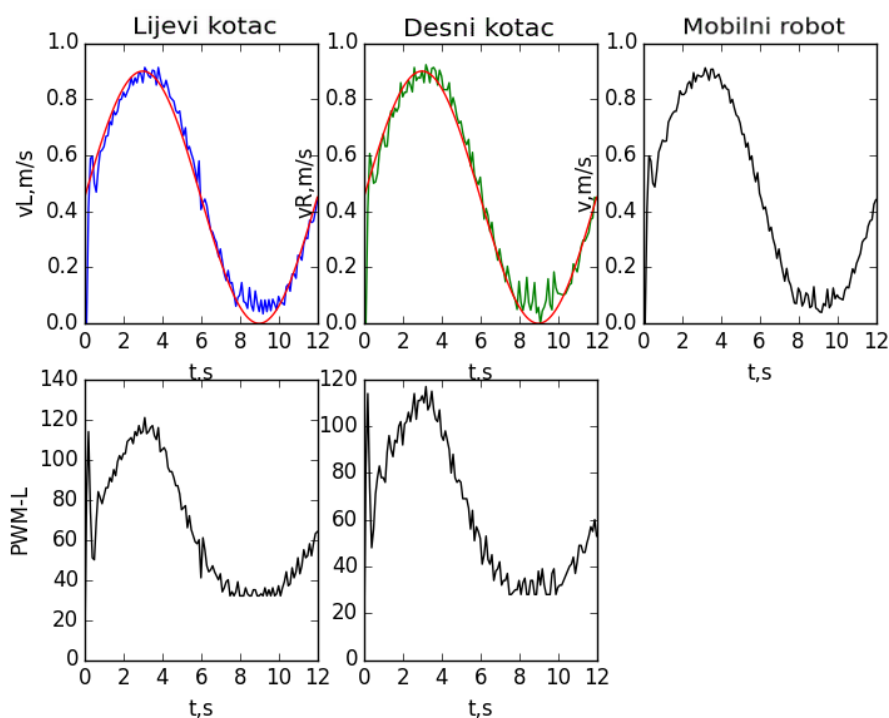


**Slika 33. Problem regulacije malih ulaznih brzina**

Kako bi se izbjegao taj problem potrebno je eksperimentalno odrediti minimalni upravljački PWM signal za koji se kotači još mogu vrtiti. Nakon što se taj signal odredio, on se povećava za otprilike 10% i postavlja kao minimalni upravljački PWM signal za ulazne brzine [Slika 34].

Teoretska ideja: Mobilni robot se reaktivnom navigacijom giba prema cilju. U trenutku kada dođe do željenog cilja mobilni robot bi trebao završiti sa radom. Analitički neizraziti regulator bi mogao odrediti ulazne brzine koje bi mogle dovesti do zaustavljanja kotača. Kod ponovnog pokretanja kotača nakratko dolazi do velikih brzina rotacije kotača koje su rezultat savladavanja početnog momenta elektromotora. Ta velika brzina, iako kratka, bi mogla uvesti veću pogrešku nego uvođenje minimalne brzine.

Problem: Glavni problem ovog rješenja je u tome što se minimalni PWM signal mora eksperimentalno odrediti kako bi se mogao uvesti. Kod gibanja mobilnog robota po podlozi bez poznavanja dinamike robota, upravljački PWM signal bi trebao biti dovoljno velik kako bi mogao spriječiti zaustavljanje mobilnog robota na različitim podlogama i usponima. Također, zaustavljanje kotača usred proklizavanja je neizbježno, pa uvođenje minimalnog upravljačkog PWM signala ne može potpuno izvršavati svoju funkciju (pošto je najčešća pojava zaustavljanja kotača rezultat proklizavanja).



**Slika 34. Uvođenje minimalnog upravljačkog PWM signala**

Iako bi uvođenje minimalnog PWM signala teoretski sprječilo problem ponovnog pokretanja mobilnog robota, zbog navedenih problema nije korišten u daljnim programima.

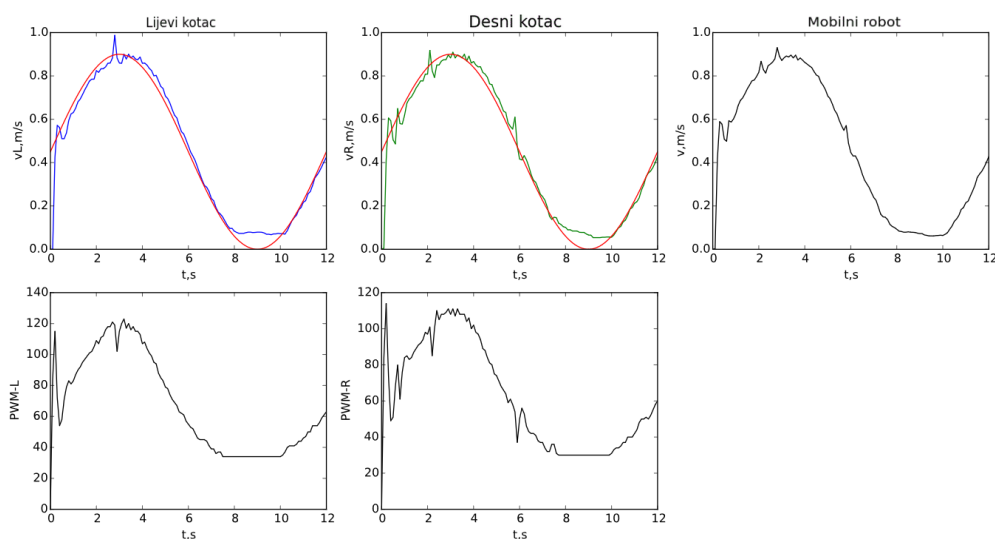
### 5.1.3.3 Promjena načina očitavanja enkodera u *RISING*

Na prethodnim slikama ([Slika 28] do [Slika 34]) enkoderi su povećavali svoju vrijednost svaki puta kada bi došlo do promjene stanja na osjetilu svjetlosti (režim rada *CHANGE*). Takav režim rada je korišten jer nam daje dvostruko veću rezoluciju, čime se poboljšava preciznost mjerenja brzina. No, iz prethodnih odziva se vidjelo da su odzivi imali puno malih šiljaka koji su dovodili do blagih oscilacija.

Jedan od glavnih razloga toga jest što su na korištenom enkoderu [Slika 13] prorezi i puni djelovi različitih veličina (dobri enkoderi bi trebali imati identičan razmak).

Kako bi se te blage oscilacije izbjegle, režim rada enkodera promijenjen je na *RISING*, koji mjeri samo promjene stanja iz niskog u visoko.

Odziv PD regulatora za takav režim rada enkodera je prikazan na slici [Slika 35].



**Slika 35. Odziv PD regulatora režim rada enkodera RISING**

Izrađeni odziv je puno glađi, no valja zamjetiti da su šiljci koji se pojavljuju mnogo izraženiji nego što su bili u prethodnim odzivima. Razlog tome je što se smanjenjem rezolucije povećala vremenska razlika između dva nova povećanja vrijednosti koju broji enkoder.

Također valja napomenuti da neke pogreške u mjerenju nastaju i zbog šuma koji nastaje radom enkodera, koje rezultiraju dosta velikim pogreškama u mjerenju brzine.

Ovim režimom rada enkodera se izgadio odziv, no potencijalne pogreške bi mogle dovesti do jače destabilizacije mobilnog robota (pošto bi regulator puno jače reagirao na krivo izmjerenu brzinu).

Iako su pogreške usred krivog očitavanja enkodera u ovom režimu rada veće nego u prethodnom, one se pojavljuju puno rjeđe. Zbog tog razloga u ostatku rada će se koristiti režim rada enkodera *RISING*.

U oba slučaja pogreške očitavanja enkodera dovode do krivog mjerenja položaja robota i kao takve su vrlo štetne za reaktivnu navigaciju u prostoru (pošto se pogreška mjerenja akumulira). Za bolje upravljanje mobilnim robotom potrebno je eliminirati te pogreške mjerenja enkodera.

U teoriji, to bi se moglo sanirati tako da se spremne svi vremenski trenutki kada je enkoder povećao svoju vrijednost. Zatim bi, prije nego što se odredi brzina trebalo bi sva ta određena vremena pretvoriti u frekvencije, sortirati ih i eliminirati sve frekvencije koje su veće od neke srednje vrijednosti, zajedno sa tim povećanjem.

Taj proces bi bio vremenski zahtjevan i kao takav nije prigodan za regulaciju, gdje je poželjno da je period ponavljanja regulacije što kraći, i zato u ovom radu nije implementiran.

Nažalost, najbolji način eliminacije pogreške u određivanju pomaka je u uvođenju dodatnih senzora za ustvrđivanje položaja mobilnog robota u prostoru. Za potrebe ovog rada najbolja bi bila vanjska kamera koja bi služila samo za provjeru položaja mobilnog robota (pošto je tema rada reaktivna navigacija mobilnog robota u nepoznatoj okolini, uvođenjem vanjske kamere, okolina više ne bi bila nepznata).

#### **5.1.4 Zaključak testiranja**

U ovim testiranjima je pokušano na jednostavan način optimizirati rad odabranog PD regulatora. Tako je ukazano na neka potencijalna rješenja i na probleme koje bi bilo potrebno riješiti za uspješno upravljanje mobilnim robotom. U daljnjim programima većina predloženih rješenja nisu korištena, iako su ostavljena u kodu programa. Razlozi toga će biti objašnjeni kasnije, no glavni razlog je manjak vremena provednog u optimizaciji pojedinog koraka u programu.

Kako je optimizacija PID regulatora vrlo složen, ali i zanimljiv proces, ovaj program je odvojen od konačnog programa kako bi bio otvoren za dodatna proširenja i testiranja.

### **5.2 Testiranje parametara analitičkog neizrazitog regulatora**

Nakon što u prethodnom koraku uspješno optimizirani parametri regulatora za regulaciju brzine kotača, potrebno je dobivena rješenja dodati u ovaj program kako bi se krenulo u testiranje rada analitičkog neizrazitog regulatora (tj. optimiziralo praćenje referentne trajektorije).

Kao i u prethodnom programu, za tu svrhu je izrađeno grafičko sučelje (GUI) u kojem korisnik određuje referentnu trajektoriju (u obliku nekoliko točaka cilja) i parametre analitičkog neizrazitog regulatora.

Za potrebe ovih testiranja izrađena su dva programa, čiji način rada će ukratko biti opisan.

#### **5.2.1 Mikrokontroler**

Sam način rada je sličan kao programa za testiranje parametara PID regulatora, pa će biti opisane samo razlike.

Za razliku od prethodnog programa, trajanje rada nije vremenski ograničeno. Kraj programa se događa tada kada je mobilni robot došao do svih određenih točaka, s time da je zbog pogreške u čitanju enkodera uvedeno dopušteno odsutpanje od točke cilja (0,005 m i za  $x$  i  $y$  koordinate).

Pošto program više nije vremenski ograničen, više nije moguće odrediti potrebnu veličinu niza varijabli iz kojih će računalo izrađivati grafove, pa je potrebno slati podatke na računalo tokom rada analitičkog regulatora.

Nakon što su parametri postavljeni i definirani, te postavljeni početni uvjeti mobilnog robota i referentne trajektorije, dok mobilni robot nije prošao kroz sve zadane točke, u intervalima od 0,1 sekunde izvršava se funkcija *fuzzy()*, čiji se rad može opisati u koracima:

- Izračunaj normu vektora (19).
- Izračunaj normirani vektorski produkt (21).
- Izračunaj skalarni produkt (25).
- Izračunaj kutnu brzinu (25) i linearnu brzinu (23) ili (24).
- Filtracija linearne i kutne brzine (isti razlog i način kao u prethodnom programu).
- Određivanje ulazne brzine lijevog (6) i desnog (7) kotača.
- PD regulator. Način rada je isti kao opisan u prethodnom programu, s time da nakon očitavanja brzine lijevog (47) i desnog (48) kotača slijedi određivanje položaja mobilnog robota integracijom jednadžbi (1), (2) i (3). Nakon što je položaj mobilnog robota određen, slijedi provjera je li mobilni robot došao do željenog cilja.
- Slanje podataka na računalo redosljedom:
  1. ulazna brzina lijevog kotača  $u_L$ ,
  2. ulazna brzina desnog kotača  $u_R$ ,
  3. brzina lijevog kotača  $v_L$ ,
  4. brzina desnog kotača  $v_R$ ,
  5. brzina mobilnog robota  $v$ ,
  6. upravljački PWM signal za lijevi kotač  $p_L$ ,
  7. upravljački PWM signal za desni kotač  $p_R$ ,
  8. linearna brzina  $u_1$ ,
  9. kutna brzina  $u_2$ ,
  10. X položaj mobilnog robot  $x$ ,
  11. Y položaj mobilnog robota  $y$ ,

## 12. Kut zakreta mobilnog robota $\theta$ .

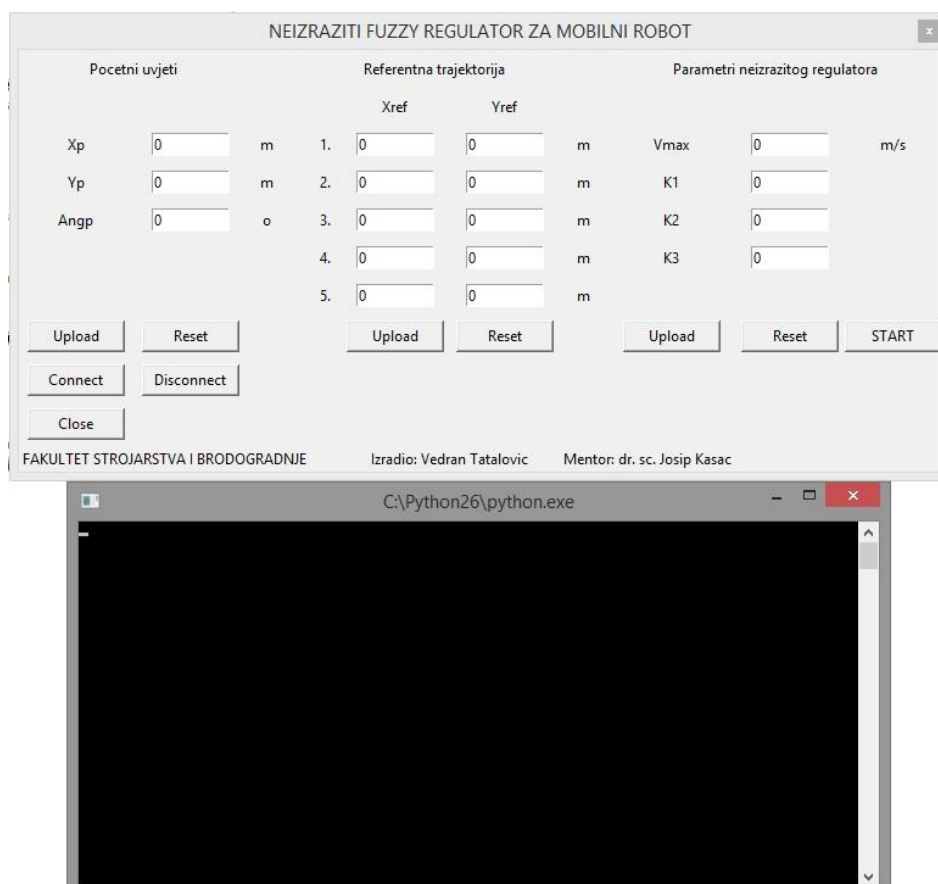
Mikrokontroler šalje jako puno podataka na računalo u jako kratkom vremenu (12 podataka svake 0,1 sekunde). Serijsko slanje podataka na računalo je vremenski zahtjevno i velika količina podataka može opteretiti program na računalo (znalo se dogoditi da se program jako uspori). Također, serijska komunikacija troši relativno puno energije, pa je poželjno da se minimizira količina podataka koji se šalju na računalo.

To se za ovaj slučaj može izvesti tako da se na računalo šalju samo podatci koji se ne mogu izračunati preko neke od jednadžbi (znači samo  $v_L$ ,  $v_R$ ).

No, pošto bi to zahtjevalo da računalo također sadrži ukomponiran set svih jednadžbi za izračun analitičkog neizrazitog regulatora, to u sklopu ovog rada nije izrađeno.

### 5.2.2 Računalo

Kao i u prethodnom programu, izrađeno je grafičko sučelje za namještanje parametara koji izgleda kao na slici [Slika 36].



Slika 36. Grafičko sučelje za testiranje analitičkog neizrazitog regulatora

Način rada programa je isti kao i onaj prethodnog programa (potpoglavlje 5.1.2), s time da sada korisnik može namještati sljedeće parametre:

- Početni položaj mobilnog robota u prostoru.
- Do 5 točaka cilja (s time da ako su  $x$  i  $y$  cilja jednaki 0, program ih zanemaruje).
- Parametre analitičkog neizrazitog regulatora  $v_{max}$ ,  $k_1$ ,  $k_2$  i  $k_3$  korištene u jednadžbama (22), (23) i (24).  $\omega_{max}$  (22) je izražen preko  $v_{max}$  preko jednadžbe.

$$\omega_{max} = 0,35(2v_{max})/L \quad (56)$$

Također, računalo iscertava grafički prikaz poslane vrijednosti u tri različita prozora:

1. Vremenske odzive parametara  $v_L$ ,  $v_R$ ,  $v$ ,  $p_L$  i  $p_R$ .
2. Vremenske odzive parametara  $u_1$ ,  $u_2$ ,  $x$ ,  $y$  i  $\theta$ .
3. Putanju mobilnog robota u prostoru i referentna trajektorija.

### 5.2.3 Izvedena testiranja

Kako bi se mogla izvesti reaktivna navigacija mobilnog robota u nepoznatom prostoru potrebno je osigurati da mobilni robot može pratiti trajektoriju koja će biti generirana. U tu svrhu je izrađen program za testiranje rada analitičkog neizrazitog regulatora. Preduvjet za uspješno izvršavanje ovog programa je zadovoljavajuće riješena regulacija kotača mobilnog robota.

Analitički neizraziti regulator nije detaljno testiran, no rezultati su pokazali da je algoritam ispravan. Iako rezultati testiranja nisu zadovoljavajući, ukazuju na neke probleme i korisni su za daljnju razradu reaktivne navigacije.

U sklopu ovog rada izrađena su sljedeća testiranja:

1. Testiranje rada analitičkog regulatora na praznom hodu za izračun translacijske brzine preko (23) i za  $\omega_{max}$ . Cilj je definiran kao jedna točka.

$$\omega_{max} = 2v_{max}/L \quad (57)$$

2. Isti kao prethodni test, samo za trajektoriju izrađenu od 3 točke.
3. Neuspjeli pokušaj dolaska mobilnog robota do točke cilja.
4. Uspjeli dolazak mobilnog robota do cilja.
5. Pokušaj praćenja trajektorije izrađene od 3 točke.
6. Dolazak mobilnog robota do cilja, uz filtraciju linearne i kutne brzine.

7. Isto kao prethodni test, ali uz smanjenje  $k_3 = 1$ .
8. Pokušaj praćenja trajektorije izrađene od 3 točke.

Maksimalna linearna brzina za sve testove je  $v_{max} = 0,5$ . Parametri  $k_1$ ,  $k_2$  su jednaki 1 dok je  $k_3 = 3$  za sve testove osim u 7. testu.

U testovima gdje se pokušava doći do samo jednog cilja, cilj postavljen u koordinati (1 1).

U testovima gdje je trajektorija izrađena od 3 točke čije koordinate glase (0,5 0,5), (0,5 1) i (1 1).

Radi bolje predodžbe, na mjestu konačnog cilja u realnom testiranju mobilnog robota je postavljena baterija.

U testovima od 3. do 8. je linearna brzina izražena jednačbom (24), a  $\omega_{max}$  jednačbom (56).

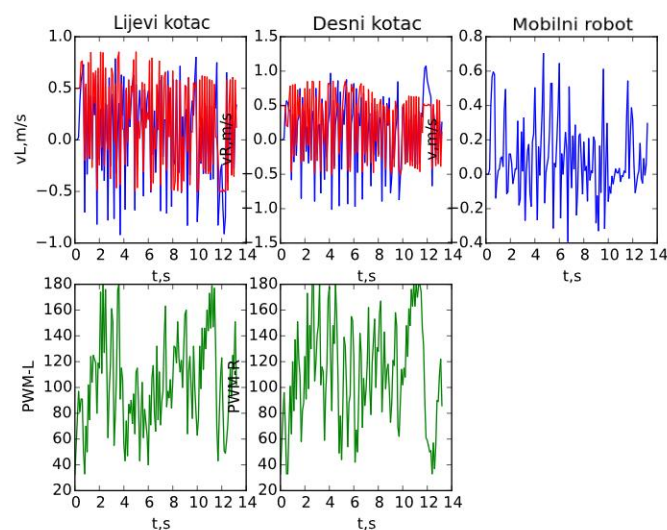
Također, pošto su testovi rađeni sa mobilnim robotom na podu, uz testove 3 do 8 je snimljen popratni video rada mobilnog robota.

Testiranja je napravljeno i više, no opisana su samo ona koja su zanimljiva za prikaz rada analitičkog neizrazitog regulatora, a da nije došlo do pogreške zbog lošeg očitavanja enkodera ili neuspjelog iscertavanja grafova zbog gubitka informacije tokom greške u serijskoj komunikaciji između mobilnog robota i računala.

### 5.2.3.1 1. Test

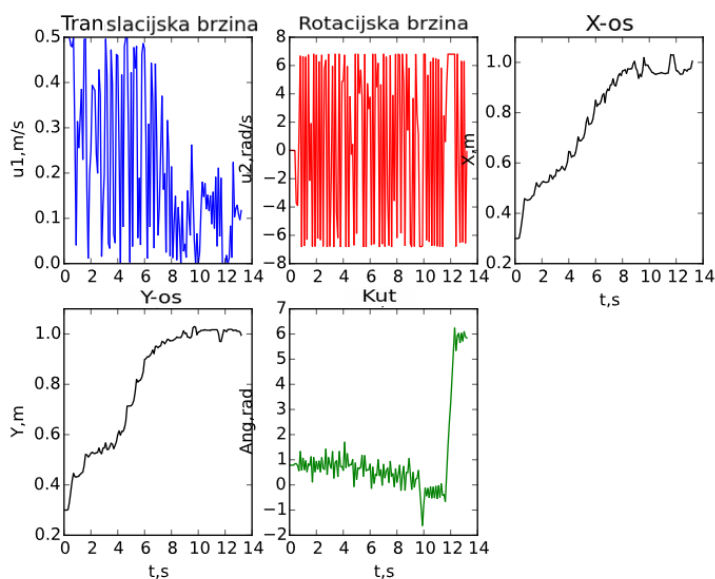
Prvi test je izrađen radi testiranja rada algoritma za gibanje u kojem linearna brzina može biti pozitivna (23).

Dobiveni odzivi su prikazani na slici [Slika 37].

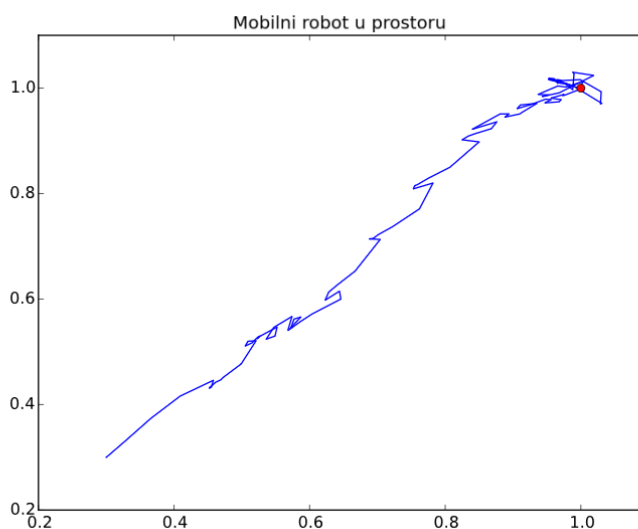


a) Odzivi PD regulatora





b) Odzivi analitičkog neizravnog regulatora



c) Mobilni robot u prostoru

**Slika 37. Rezultati 1. testa**

Prema dobivenim rezultatima vidi se da se mobilni robot počinje gibati prema željenom cilju. Kutna i linearna brzina se jako mijenjaju mijenjaju, što negativno rezultira na odzive PD regulatora koji jako odskaču od traženih ulaznih brzina (što je začaran krug, pošto to negativno utječe na novo generiranu kutnu i linearnu brzinu).

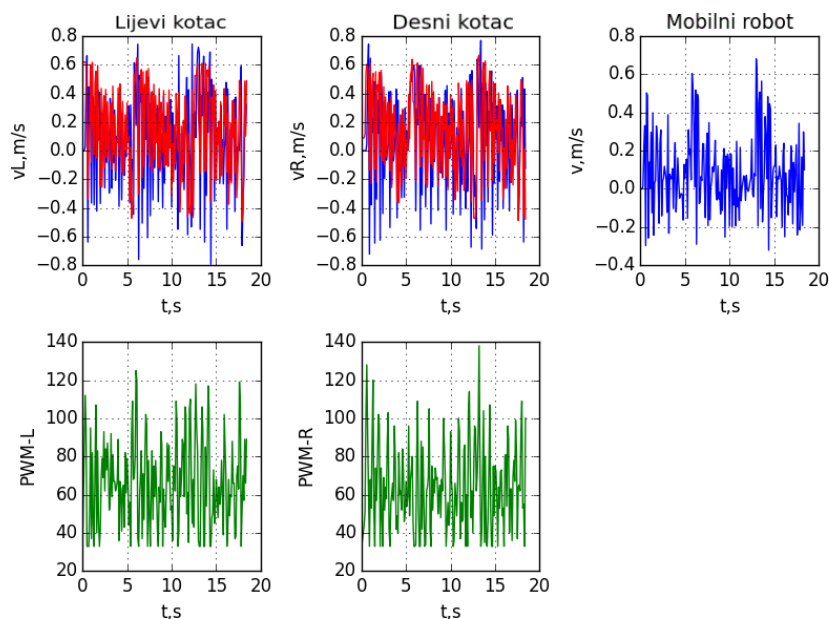
Također se vidi problem upravljanja samo preko pozitivnih linearnih brzina (23). Svaki put kad mobilni robot prođe cilj, on se mora zaokrenuti i probati doći do cilja (što jako dugo traje).

No, pošto je test izrađen na praznom hodu kotača, jedino što nam on stvarno može dokazati jest da algoritam radi ispravno.

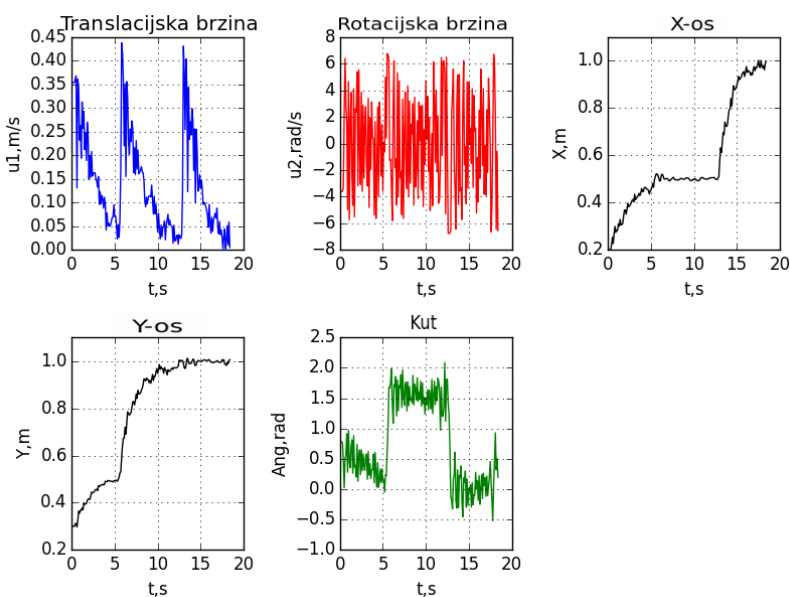
### 5.2.3.2 2. Test

U drugom testu je testiran rad ostatka programa. Početni položak mobilnog robota je postavljen na koordinate (0,2 0,3) a kut zakreta  $\theta$  na 0,5.

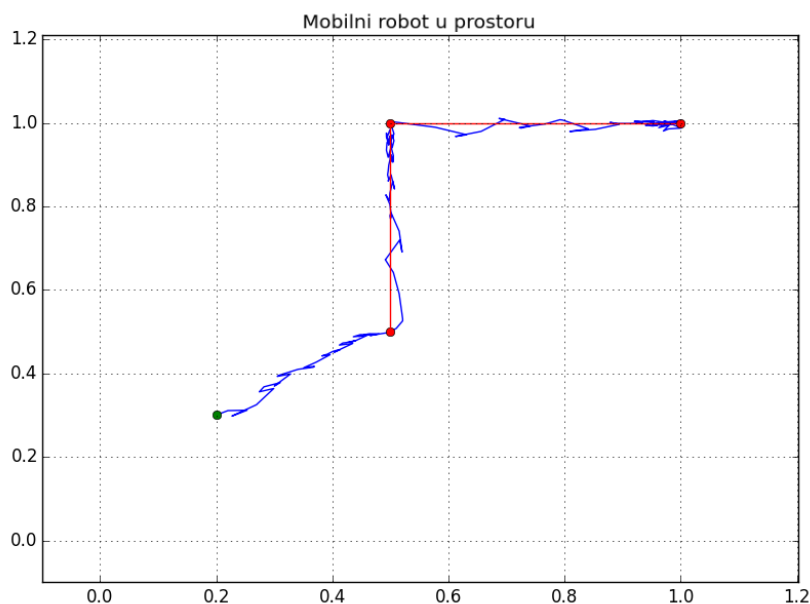
Dobiveni izlazi su sljedeći [Slika 38].



a) Odzivi PD regulatora



b) Odzivi analitičkog neizravnog regulatora



c) Mobilni robot u prostoru

**Slika 38. Rezultati 2. testa**

Iz ovog testa jedina zanimljivost koja se nije mogla vidjeti iz prethodnog testa je da linearna brzina radi dobro jer što se više približava cilju postaje sve manja.

Također ovim testom je dokazano da cijeli algoritam teoretski radi ono što se od njega očekuje.

### 5.2.3.3 3. Test

U prethodna dva testa dokazano je da izrađeni algoritam doista slijedi referentnu trajektoriju preko analitičkog neizrazitog regulatora. No također se pokazalo da u slučaju u kojem linearna brzina može biti samo pozitivnih vrijednosti može doći do velikih problema pri cilju. Pošto mobilni robot ima i stražnje ultrazvučne senzore moguće je i uvesti negativne vrijednosti linearne brzine (24). Iako u realnosti stražnji senzori nisu osposobljeni da odguruju mobilni robot od prepreke, za potrebe rada se pretpostavlja da jesu.

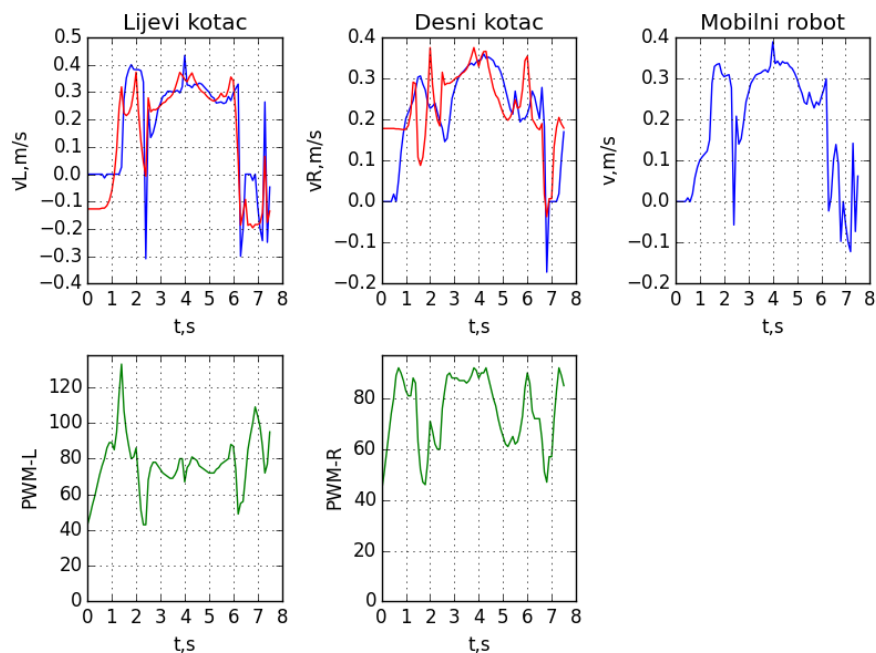
Kako bi se stvarno mogao testirati rad analitičkog neizrazitog regulatora, potrebno je mobilni robot staviti na pod i postići da u realnom slučaju uspije dostići trajektoriju.

Rezultati prikazani na slici [Slika 39] prikazuju rezultate gibanja kako ih je mobilni robot izgenerirao "u svojoj glavi", dok video [20] prikazuje kako se stvarno mobilni robot gibao.

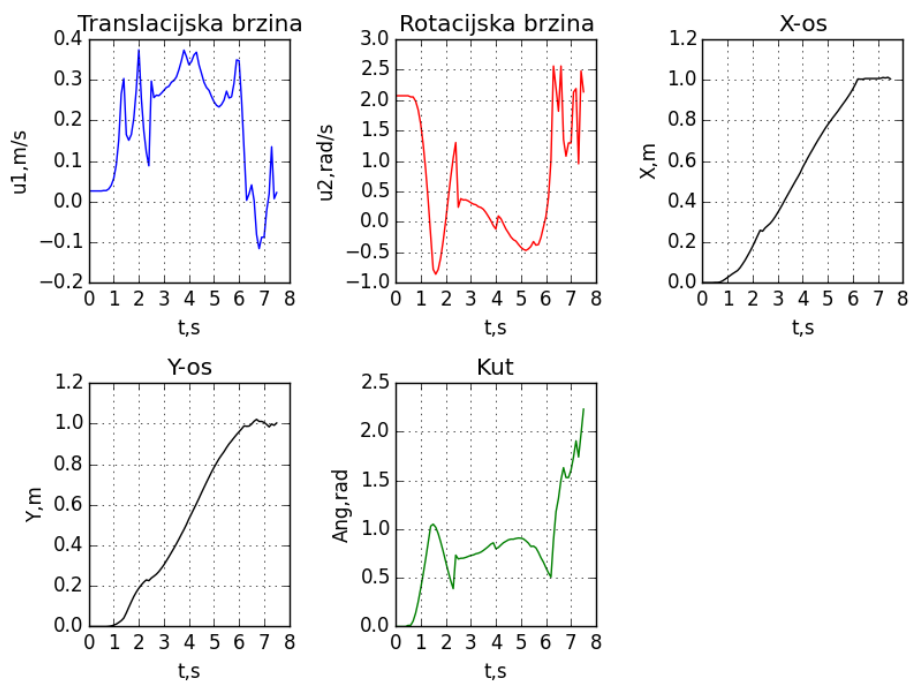
Dobiveni odzivi prikazuju relativno stabilno gibanje mobilnog robota do cilja u relativno kratkom vremenu.

Međutim, priloženi video prikazuje da je mobilni robot uistinu počeo sa takvim gibanjem, no zbog greške u čitanju enkodera dogodilo se da je mobilni robot krivo prepoznao gdje se nalazi

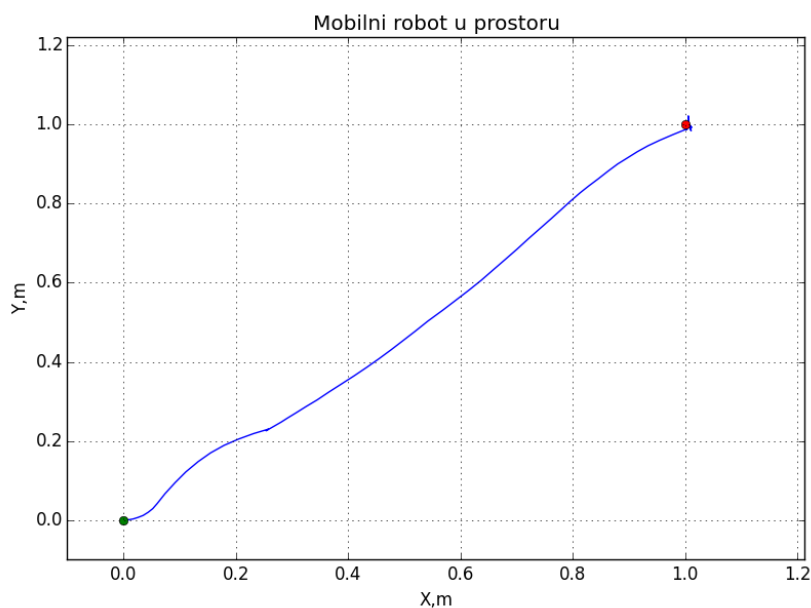
u prostoru i kao rezultat toga nastavio krivom putanjom. Ta potencijalna pogreška rada može biti rezultat velikih "šiljaka" u odzivu brzine lijevog i desnog kotača.



a) Odzivi PD regulatora



b) Odzivi analitičkog neizrazitog regulatora



c) Mobilni robot u prostoru

**Slika 39. Rezultati 3. testa**

Iako su odzivi koje je mobilni robot izračunao bili relativno dobri, mobilni robot u stvarnosti nije uspio doći do željenog cilja. Zbog tog razloga ovaj test nije dobar kao dokaz rada programa.

#### 5.2.3.4 4. Test

U prethodnom testu, zbog krivog očitavanja enkodera, mobilni robot nije uspio doći do traženog cilja. U ovom testu je uz iste parametre neizrazitog regulatora mobilni robot dobio izlaze prikazane na sljedećoj slici [Slika 40].

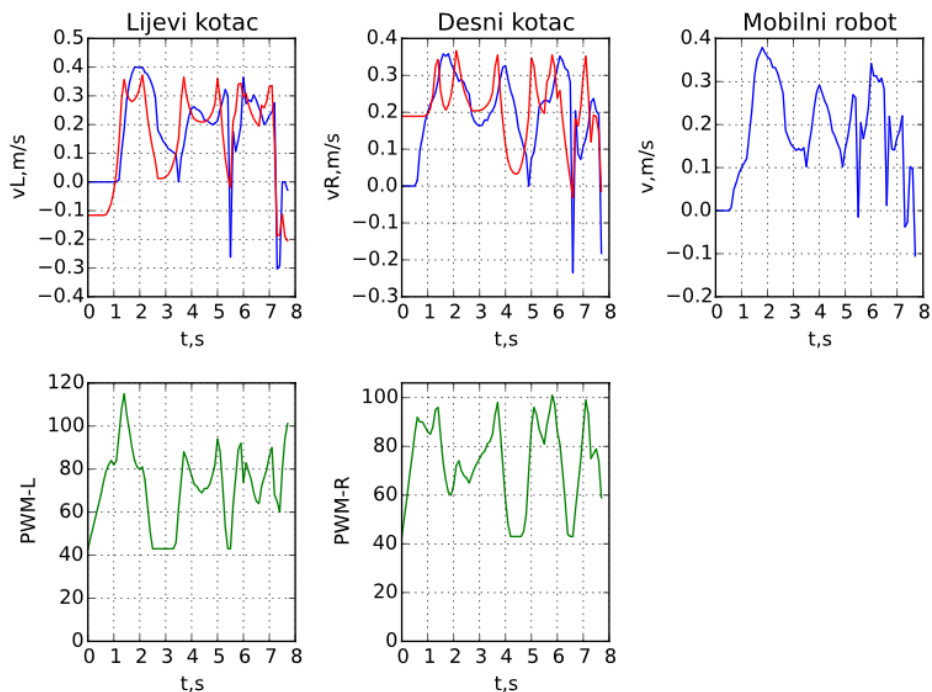
Kao što se vidi iz priloženog videa [21], mobilni robot je uspio doći do zadanog cilja radeći pritom gibanje skoro identično onome kojeg je mobilni robot izračunao.

Ovim testom je dokazano da algoritam radi ispravno i da su potencijalne pogreške rezultat lošeg rada enkodera.

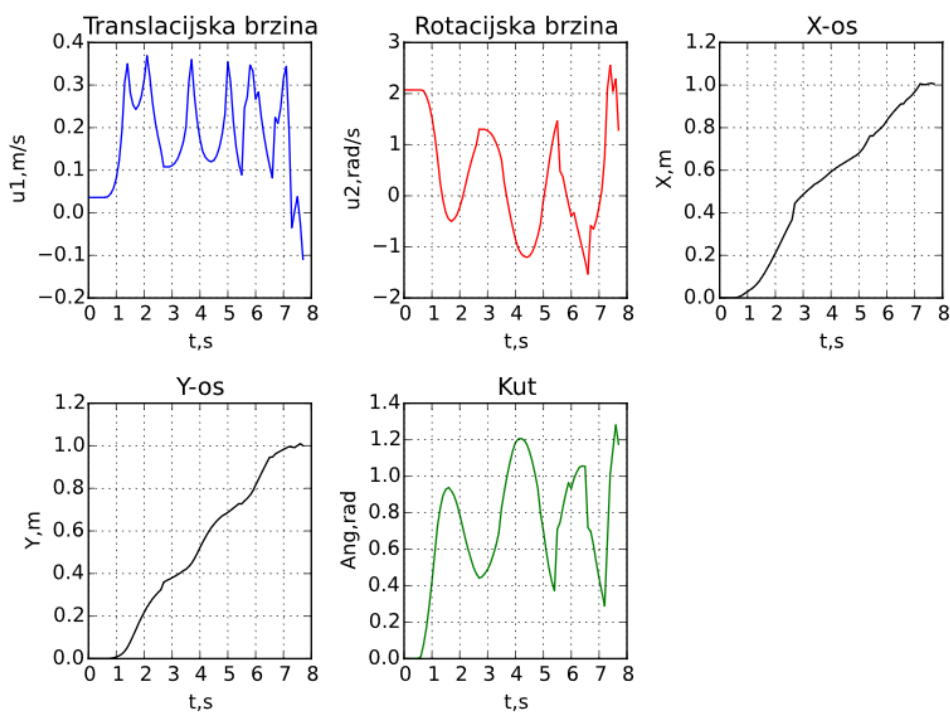
Radi bolje sigurnosti test je isproban 10 puta za iste parametre regulatora i robot je uspio doći dovoljno blizu cilju da se može smatrati uspješnim (unutar razumljivih raspona pogreške) u 5 od 10 slučajeva.

Prethodna dva testa ukazuju na jedan problem koji se događa u praktičnoj izvedbi gibanja mobilnog robota. Mobilni će robot za iste parametre analitičkog neizrazitog regulatora i PD regulatora dobiti drugačije odzive u različitim pokušajima.

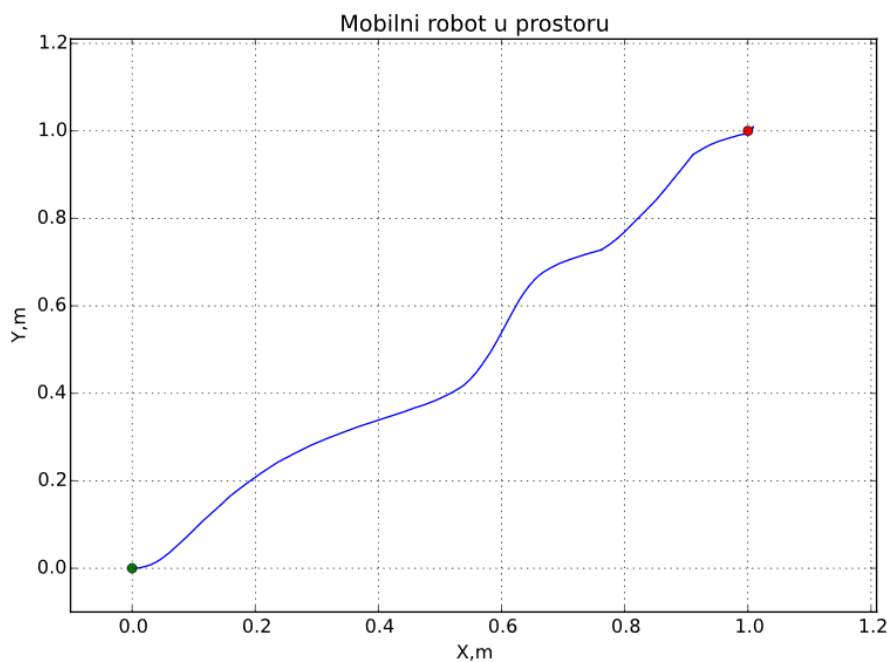
Osiguravanje skoro istih odziva za iste parametre je možda najveća, ali ujedno i najzanimljivija, problematika u realnoj izvedbi mobilnog robota.



a) Odzivi PD regulatora



b) Odzivi analitičkog neizrazitog regulatora



c) Mobilni robot u prostoru

**Slika 40. Rezultati 4. testa**

### 5.2.3.5 5. Test

Prethodni testovi su dokazali ispravnost programa, no ujedno su ukazali na problem određivanja putanje samo preko informacija iz enkodera.

Iako zbog nemogućnosti osiguranja velike ponovljivosti (barem 9 od 10 uspješnih pokušaja) navigacija mobilnog robota samo pomoću enkodera nije dobra, daljnja testiranja su izrađena u pokušaju nalaženja problema koji najviše onemogućuje dobru putanju samo preko informacija prikupljenih iz enkodera.

Ovaj test je prikaz pokušaja mobilnog robota da dođe do tri uzastopne točke, s time da je mobilni robot počeo gibanje na točki (0,1 0,2). Na koordinate (0 0) je postavljena baterija kako bi se bolje prikazao početni pomak mobilnog robota.

Rezultirajući odzivi su bili dobri (i brzi i precizni). No, kao što se vidi na videu [22], mobilni robot nije ostvario takvo gibanje (uzevši u obziru da je ovo najbolje gibanje od 10 pokušaja, možemo sa sigurnošću reći da je navigacija pomoću informacija iz samo enkodera neizvediva za bilo kakve složenije putanje).

No, snimljeni video ukazuje na to da se pogreške javljaju kada mobilni robot mora puno promijeniti kut zakreta  $\theta$ .

Teoretska jednadžba promjene kuta zakreta je (3). No, pošto mobilni robot ne može odmah ostvariti traženu kutnu brzinu, realna jednadžba za promjenu kuta zakreta je:

$$\dot{\theta} = \omega \quad (58)$$

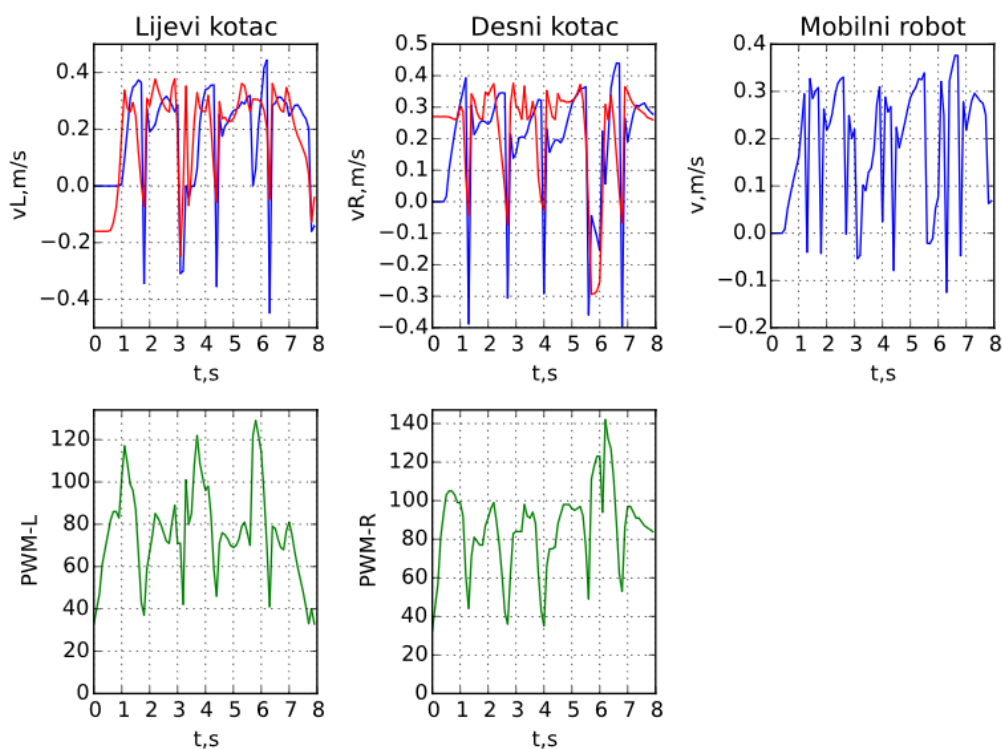
gdje je  $\omega$  jednak.

$$\omega = \frac{v_R - v_L}{L} \quad (59)$$

Pošto je kut zakreta izražen u radijanima (rad), za značajne promjene brzine koje nastaju pogreškom mjerenja enkodera nastaje velika pogreška u izračunatom kutu.

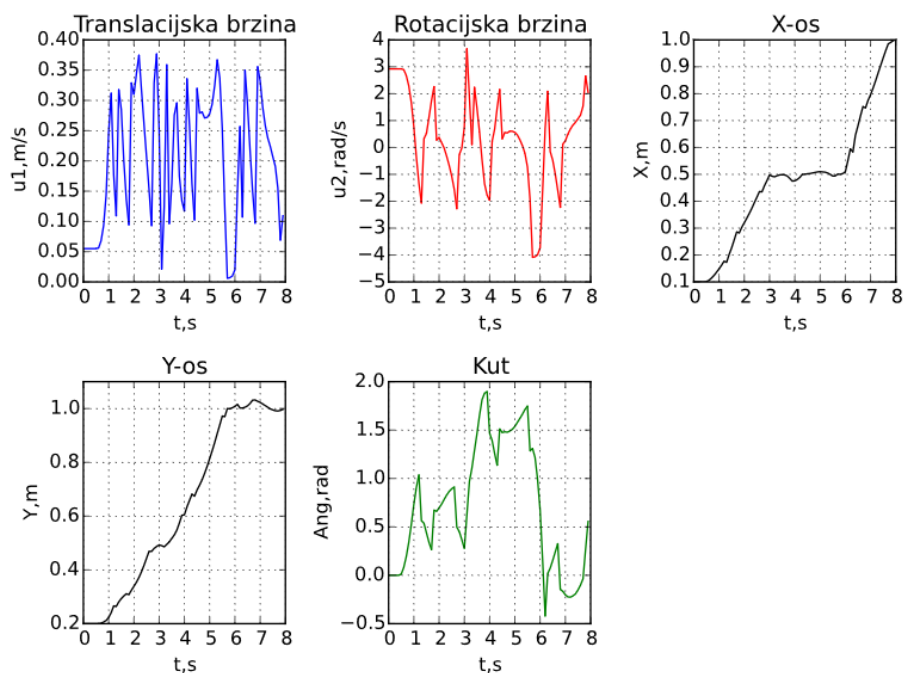
Dokaz da je najveći problem u loše izračunatom kutnom zakretu prikazuje video [23].

Kao što se vidi u videu u trenutku velike promjene kuta zakreta, enkoderi su krivo očitali vrijednosti, što je dovelo do toga da mobilni robot misli da je pravilno orjentiran, iako je zapravo pogriješio za skoro 180°.

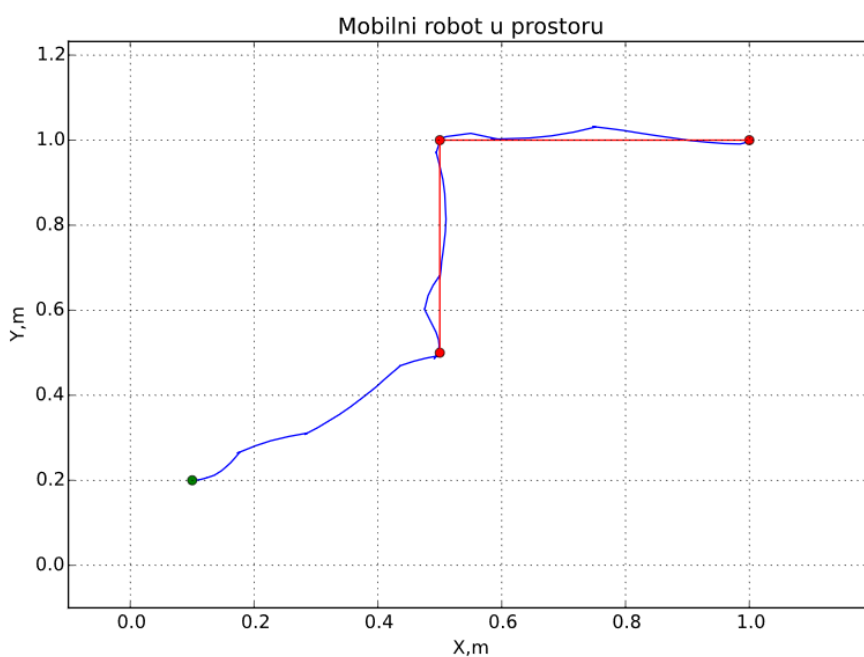


a) Odzivi PD regulatora





b) Odzivi analitičkog neizrazitog regulatora



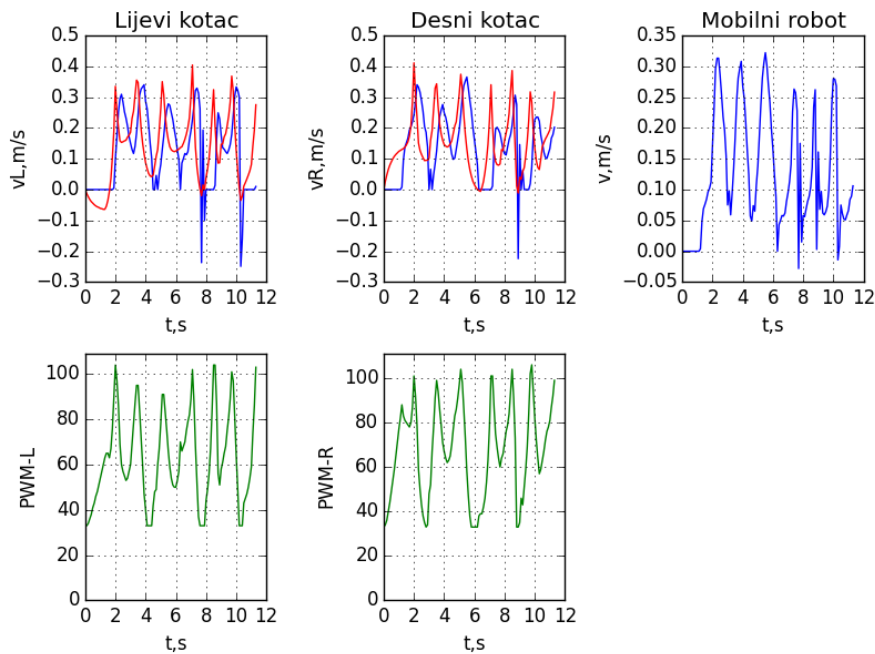
c) Mobilni robot u prostoru

**Slika 41. Rezultati 5. testa**

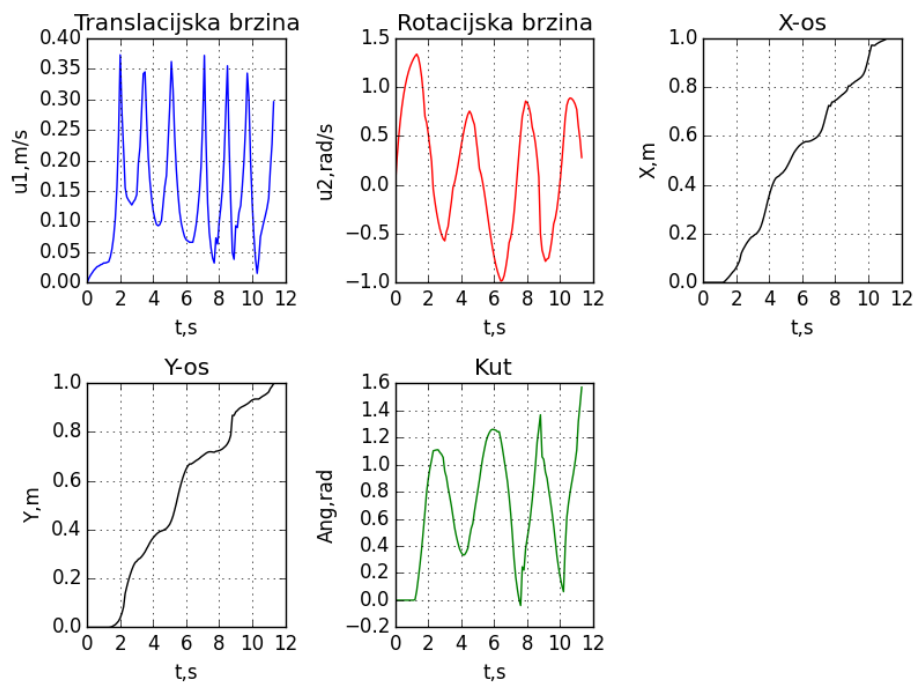
### 5.2.3.6 6. Test

Kako je u prethodnom programu odziv sustava bio bolji sa uvedenom filtracijom ulaznih brzina, u narednim testovima to je isto isprobano na ulaznoj linearnoj i kutnoj brzini.

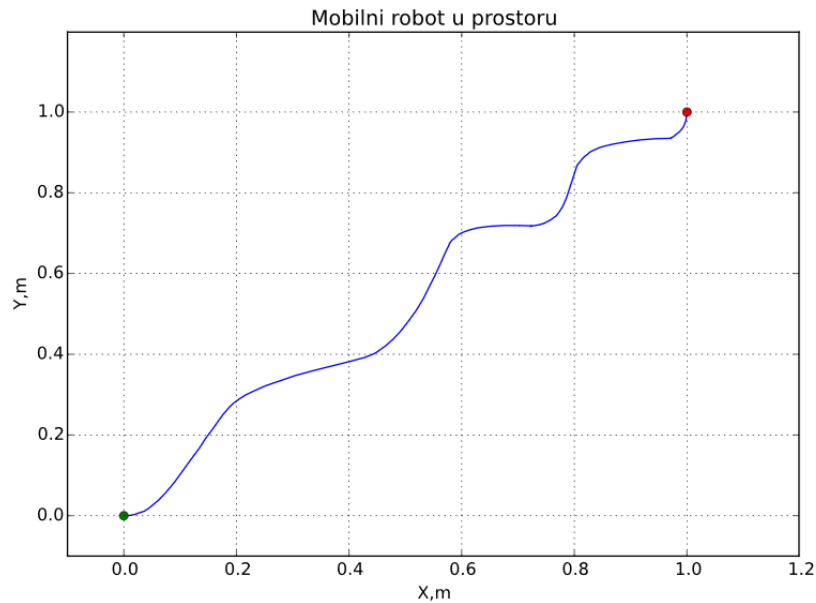
Iz rezultirajućih odziva [Slika 42] i snimljenog videa [24], može se zaključiti da filtracijom se mobilni robot počinje gibati nešto oscilatornije prema prepreci. Također se vrijeme potrebno da mobilni robot dođe do cilja povećalo.



a) Odzivi PD regulatora



b) Odzivi analitičkog neizrazitog regulatora



c) Mobilni robot u prostoru

**Slika 42. Rezultati 6. testa**

Iako bi se iz danih odziva i snimljenog filma moglo reći da uvođenje filtracije nije dobra ideja, valja napomenuti kako od napravljenih 10 testiranja, vozilo je došlo relativno uspješno do cilja 8 puta. No, pošto velika greška u enkoderima može dovesti do velike greške u programu (a samim time i udaljavanju od cilja), prolaznost programa možda jedini način određivanja kvalitete programa.

Valja napomenuti kako je filtracija ulaznih brzina lijevog i desnog kotača otežavala ostvarenje gibanja, pa je bila isključena (iako je ostavljena u programu za daljnje poručavanje).

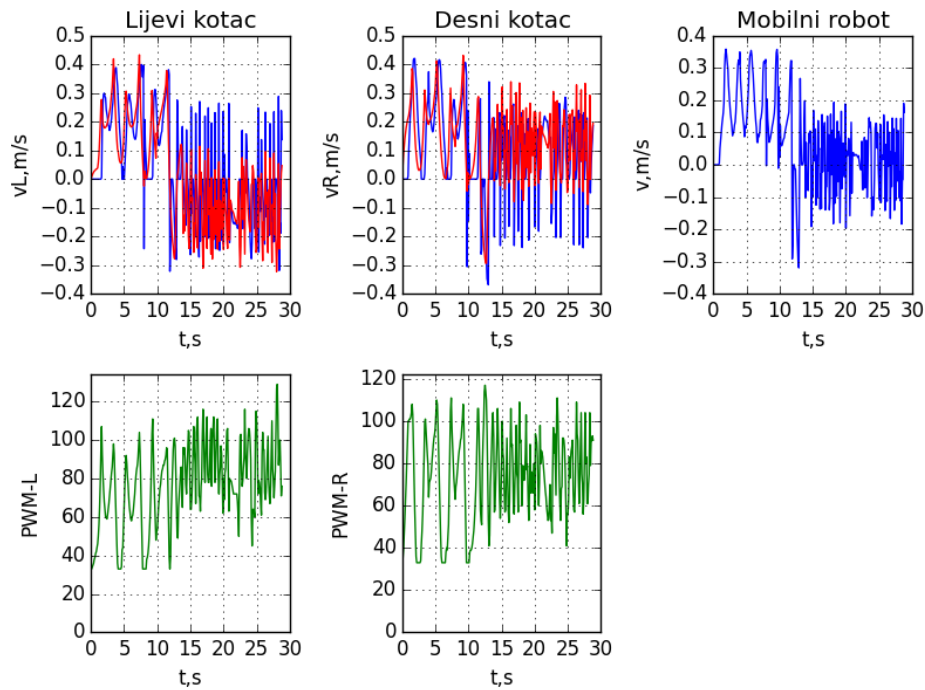
*5.2.3.7 7.Test*

Zbog manjka vremena (i kratkog životnog vijeka baterija) nije provedeno puno testiranja u kojima se mijenjaju parametri  $k_1$ ,  $k_2$  i  $k_3$ .

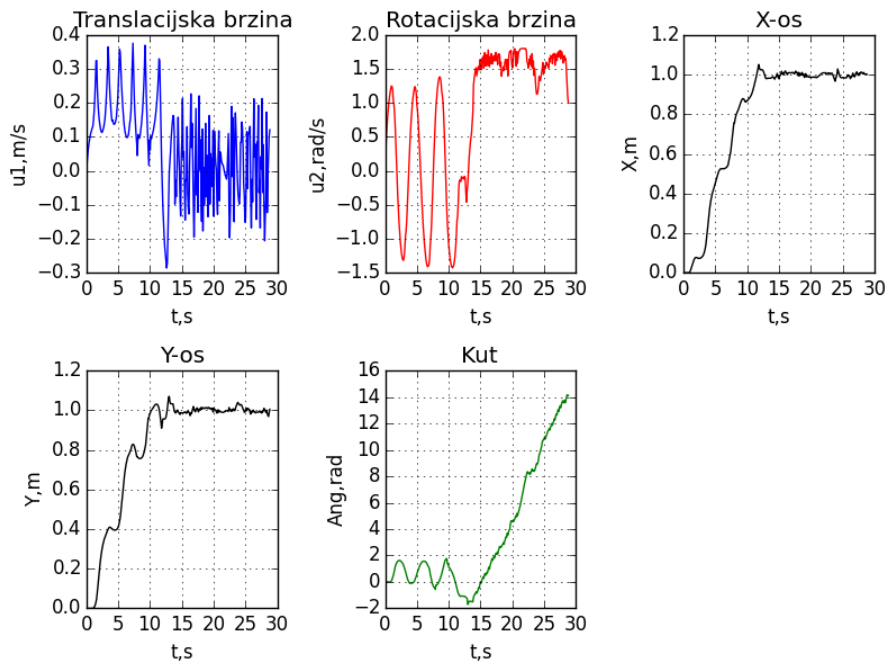
U ovom je testu je promjenjena vrijednost  $k_3$  na 1 ( $k_3$  utječe na generiranu linearnu brzinu (23)).

Iz dobivenih odziva [Slika 43] i snimljenog videa [25] se vidi da je došlo do jače dominacije linearne brzine. No, ovdje se dogodio jedan drugi problem. Zbog dominantne linearne brzine, mobilni robot nije uspio pogoditi cilj pa je dugo (oko 16 sekundi) proveo pokušavajući pronaći cilj (to također govori da bi bilo poželjno povećati prihvatljivo odstupanje od postizanja cilja).

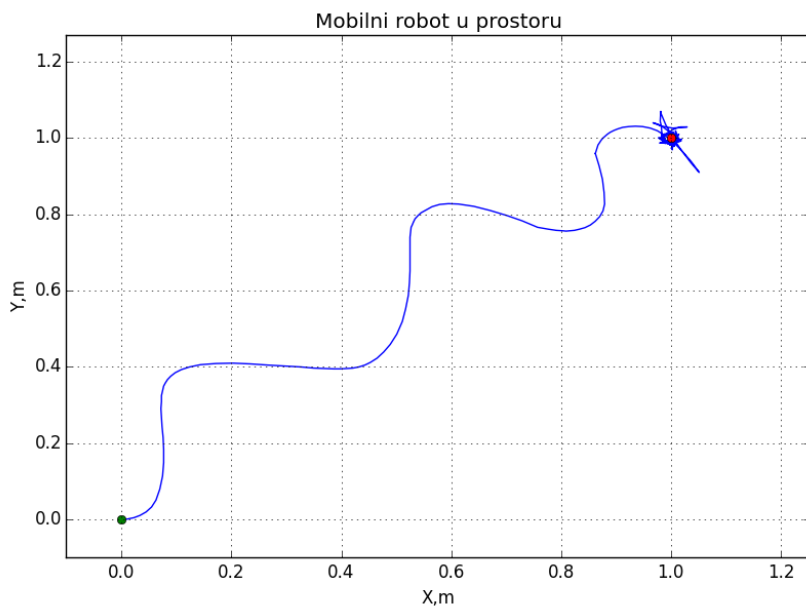
Također je isprobano povećanje  $k_3$  člana, što je dovelo do smanjenja oscilacija (pošto je dominantnija bila kutna brzina koja je održavala putanju u obliku pravca).



a) Odzivi PD regulatora



b) Odzivi analitičkog neizrazitog regulatora

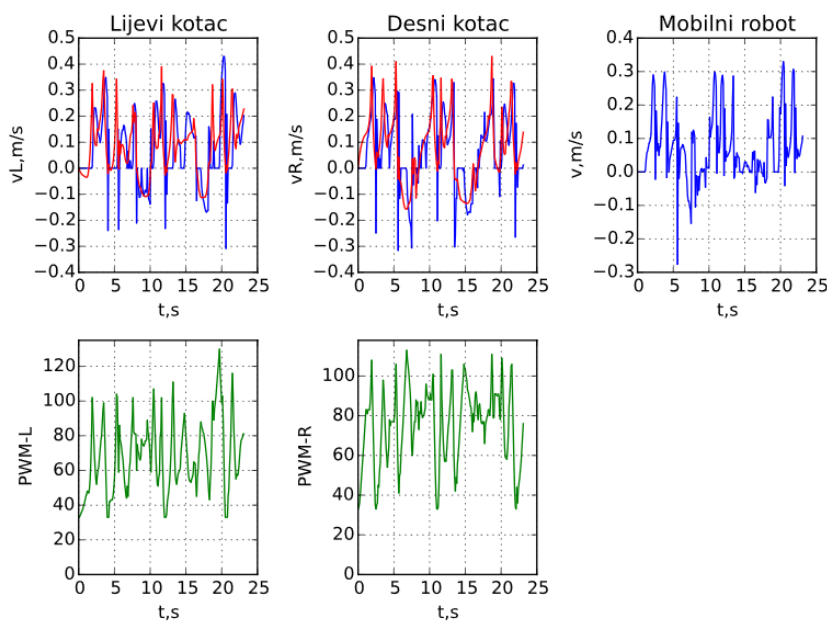


c) Mobilni robot u prostoru

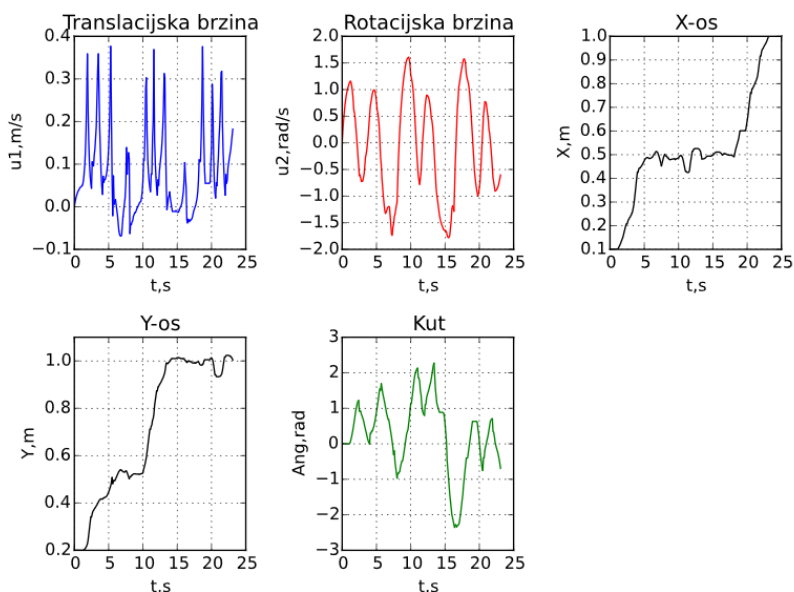
**Slika 43. Rezultati 7. testa**

5.2.3.8 8. Test

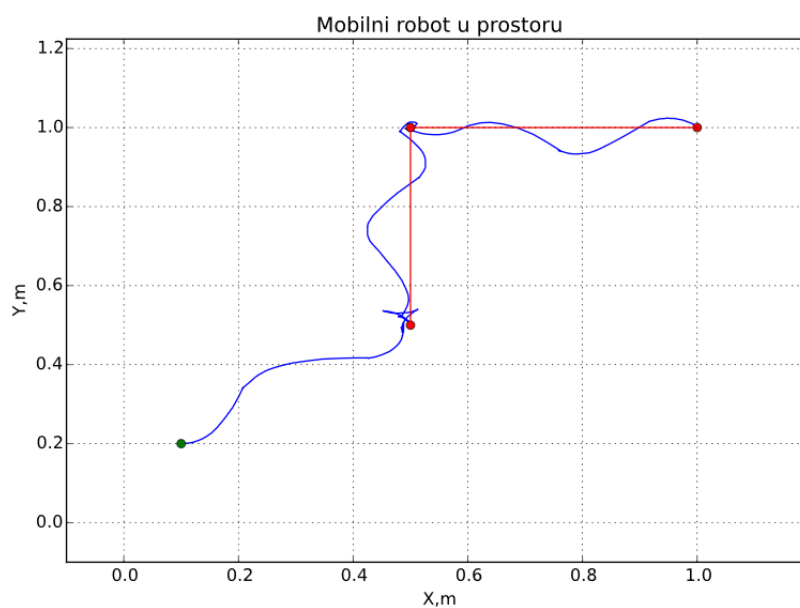
U zadnjem testu je pokušano izvesti gibanje identično onome u 5. testa uz filtraciju generirane linearne i kutne brzine. Iz dobivenih odziva [Slika 44] i snimljenog videa [26] se može zamjetiti da, osim povećanja potrebnog vremena puta, nije došlo do značajne promjene.



a) Odzivi PD regulatora



b) Odzivi analitičkog neizrazitog regulatora



c) Mobilni robot u prostoru

Slika 44. Rezultati 8. testa

### 5.2.4 Zaključci testiranja

Iz provedenih testova je dokazano da izrađeni program radi i za realno gibanje mobilnog robota na podlozi. No, također je ukazano na dva problema kod realizacije tog gibanja.

Prvi i najveći problem su pogreške koje su nastale kao posljedica loše izračunatih vrijednosti, koje su bila posljedica loših očitavanja na enkoderima. Taj problem je posebice izražen kod

velikih promjena u kutu zakreta mobilnog robota (tj. u situacijama u kojima se mobilni robot mora rotirati na mjestu). Jedan od načina sanacije tog problema je, u slučaju da želimo izbjeći korištenje povratne informacije o položaju robota (npr. vanjska kamera), uvođenje troosnog akcelerometra. Troosni akcelerometar, iako i sam pati od akumulacije pogreške poput enkodera, je precizniji u određivanju kuta zakreta mobilnog robota nego što to može ostvariti enkoder.

Drugi problem je u tome što nastaju veliki preskoci brzine zbog neoptimiziranosti PD regulatora.

Jedan od načina poboljšanja odziva PD regulatora se zasniva u činjenici kako se DC elektromotorima brzina mijenja linearno u odnosu na povećanje napona na jednom spektru brzina.

U ovom radu je istestirano da se elektromotori ponašaju linearno u odnosu na napon u rasponu brzina od 0,1 m/s do 0,8 m/s.

Ideja poboljšanja odziva PD regulatora je ta da bi se u situacijama velikih promjena ulaznih brzina koristio aproksimirani upravljački signal dobiven iz linearne ovisnosti promjene brzine o naponu određene eksperimentalno (u naprednijoj izvedbi mobilni robot bi ju analizom podataka određivao sam). Preduvjet za to bi bilo ograničenje linearne brzine ispod brzina na kojima još vrijedi linearna ovisnost brzine vrtnje.

Takva metoda bi potencijalno mogla služiti i za eliminaciju velikih razlika između izmjerene brzine na enkoderima (nastalih usred greške mjerenja).

No, pošto bi za pravilnu implementaciju toga bilo potrebno da je maksimalni napon na elektromotorima konstantan (što u ovom radu nije slučaj), ova metoda se nije implementirala. Zbog manjka vremena i nedostataka u *hardware*-a, detaljna ispitivanja promjene parametara analitičkog neizravnog regulatora i njihova optimizacija nisu izvršeni u sklopu ovog rada.

### 5.3 Testiranje parametara metode potencijalnih polja

Tema ovog rada je reaktivna navigacija mobilnog robota metodom potencijalnih polja. Za te potrebe izrađena su tri programa pomoću kojih bi se (nakon optimizacije prethodnih programa i dodavanja u program) moglo testirati generiranje referentne trajektorije primjenom metode potencijalnih polja.

1. Mikrokontroler "*Slave*" (opisan u potpoglavlju 4.5).
2. Mikrokontroler koji upravlja mobilnim robotom.

3. Računalo koje prima informacije.

### 5.3.1 Upravljački mikrokontroler

Mikrokontroler sadrži program koji je napravljen od prethodna dva programa i predstavlja posljednji korak u izradi reaktivne navigacije.

Glavna razlika u odnosu na prethodni program (opisan u potpoglavlju 5.2.1) je u tome što mobilni robot, umjesto da krene ravno prema zadanom cilju, postepeno generira trajektoriju koja se mijenja ovisno o podacima prikupljenim sa senzora.

Rad programa je izveden tako da se izvodi sve dok mobilni robot ne dođe do cilja ili dok ne prođe 21 sekunda.

Mikrokontroler u intervalima od 0,15 s poziva funkciju *ReactiveNav()*, čiji se način rada može opisati u koracima:

- Mikrokontroler prikuplja podatke sa senzora (opisano u potpoglavlju 4.5).
- Ovisno o iščitanim vrijednostima iz senzora, odlučuje se za određeni model po kojem se računa promjena referentne trajektorije (zbog lakšeg testiranja, taj korak je zakomentiran i neće se koristiti).
- Iz podataka dobivenih iz senzora određuju se koordinate prepreka (50), (51) i (52).
- Mikrokontroler izračunava atraktivni potencijal (27).
- Mikrokontroler izračunava repulzivni potencijal (28).
- Određuje se promjena referentne trajektorije ovisno o odabranom modelu. Za potrebe testiranja, model rada je preko rotacijskog potencijala (37) i (38).
- Integracijom dobivenih promjena referente trajektorije izračunava se nova referentna trajektorija.
- Ostatak programa je identičan prethodnom programu (potpoglavlje 5.2.1).

Mikrokontroler šalje na računalo (uz varijable opisane u prethodnom programu) još  $x_r$ ,  $y_r$  i  $V$ .

Zbog povećanja količine podataka koje se u svakom novom koraku šalju na računalo, period ponavljanja funkcije je povećano sa 0,1 s na 0,15 s. Za optimalnije izvođenje potrebno je minimizirati količinu podataka koji se šalju na računalo.

Kako je osnovna ideja rada programa sljedeća:

- Prikupi podatke iz senzora i na osnovu njih izgeneriraj novu trajektoriju.

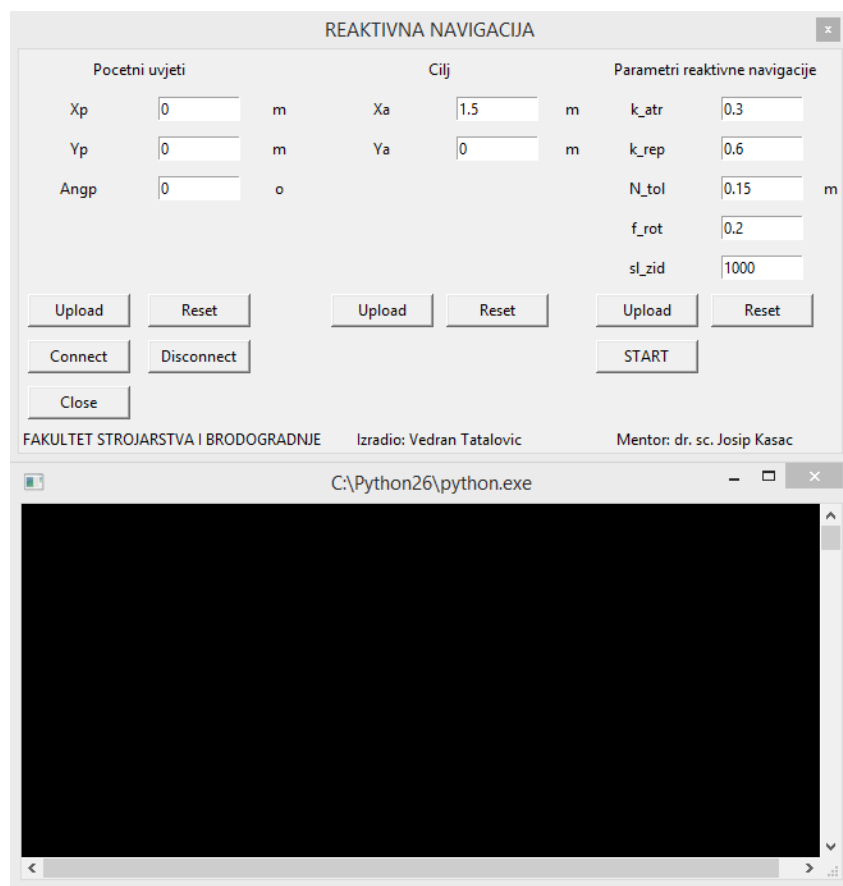


- Preko rada analitičkog neizrazitog regulatora ostvari linearnu i kutnu brzinu potrebnu za ostvarenje referentne trajektorije.
- Preko PD regulatora ostvari brzinu lijevog i desnog kotača koje ostvaruju željenu kutnu i linearnu brzinu.

Ako prethodna dva programa nisu dobro optimizirana za pretpostaviti je da će program imati poteškoća sa radom.

### 5.3.2 Računalo

Kao i u prethodnim programima izrađeno je grafičko sučelje za namještanje traženog cilja, početnih uvjeta mobilnog robota, kao i parametara neizrazitog regulatora [Slika 45].



Slika 45. Grafičko sučelje za testiranje metode potencijalnih polja

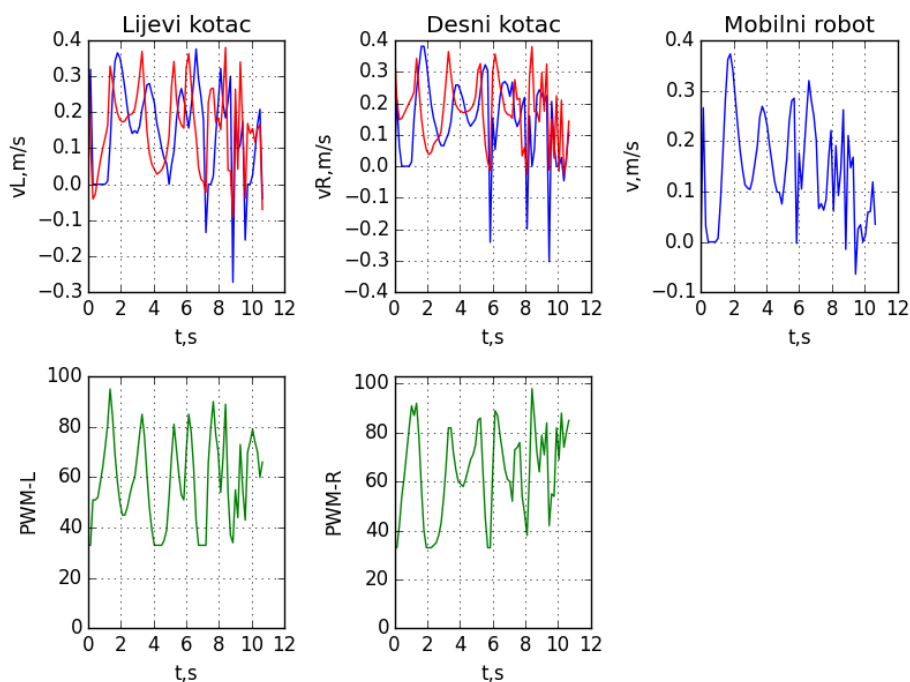
Parametri pomoću kojih korisnik namješta rad metode potencijalnih polja su, redom:  $k_a$ ,  $k_r$ ,  $k_a$ ,  $N_{tol}$ ,  $\mu_{rot}$  i  $sl$ .

Računalo također ispisiuje vremenski odziv referentne trajektorije  $x_r$  i  $y_r$ , no ne ispisiuje njihov odziv u prostoru.

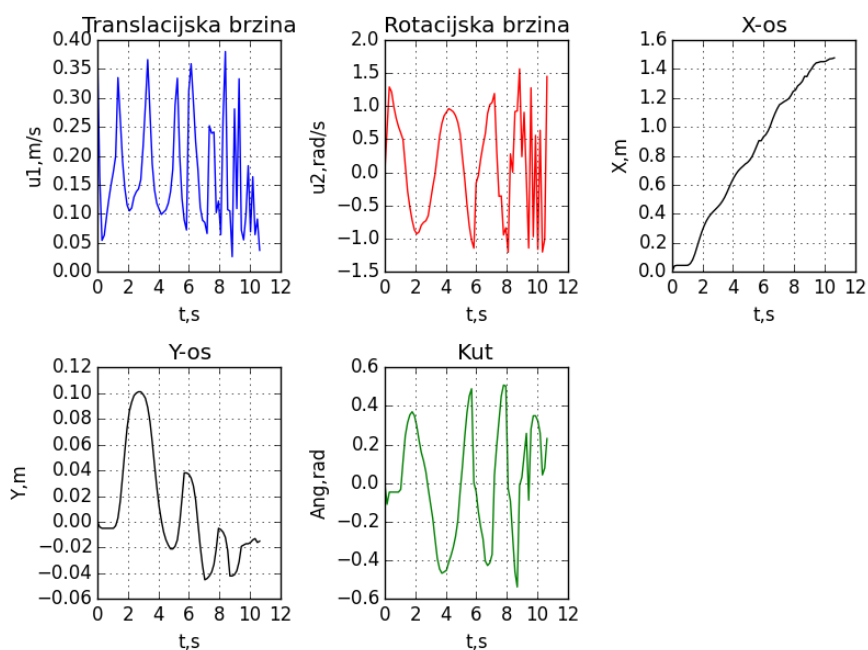
### 5.3.3 Izvedeno testiranje

Za provjeru rada programa testiran je rad programa za parametre namještene na slici [Slika 45]. Parametri analitičkog neizrazitog regulatora su namješteni isto kao u 1.Testu (5.2.3.1) dok su parametri PD regulatora isti kao u (53).

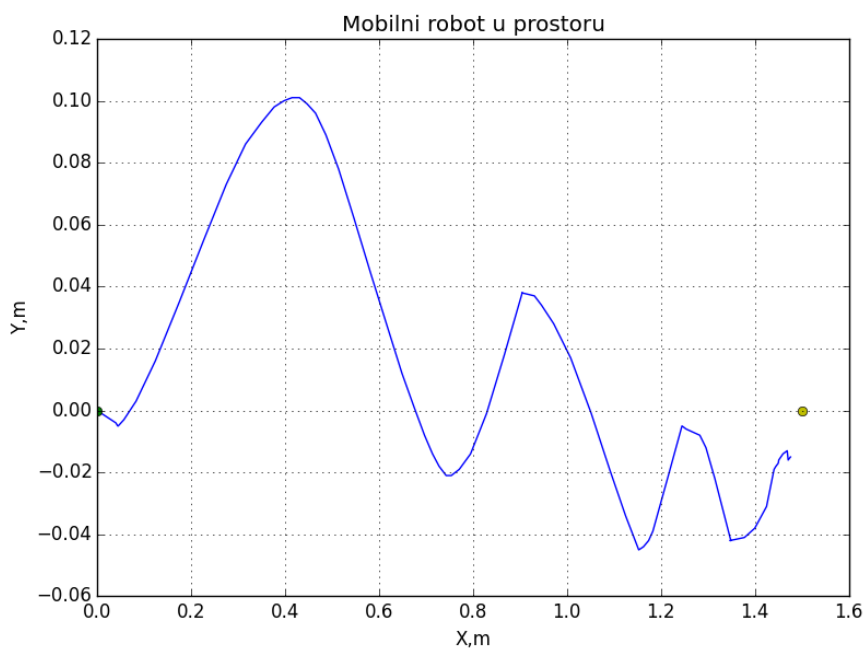
[Slika 46] prikazuje dobivene rezultate dok video [27] prikazuje što se stvarno dogodilo.



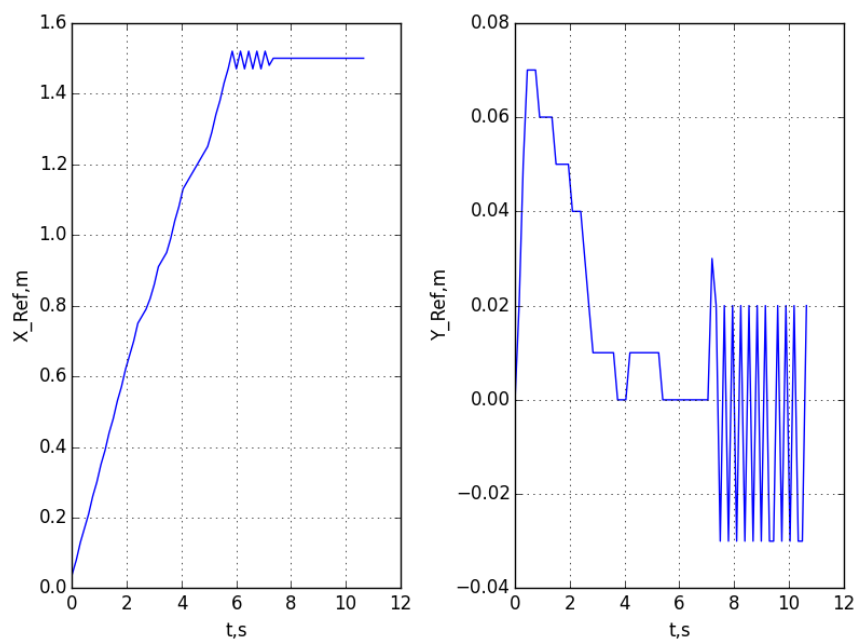
a) Odzivi PD regulatora



b) Odzivi analitičkog neizrazitog regulatora



c) Mobilni robot u prostoru



d) Odziv referentne trajektorije

**Slika 46. Rezultat testiranja metode potencijalnih polja**

Iz danih rezultata i snimljenog videa se vidi da mobilni robot nije registrirao prepreku i kao posljedica toga se sudario sa njom.

### 5.3.4 Zaključci testiranja

Iz prethodnog testiranja vidi da mobilni robot nije uspio zaobići prepreku (provjereno na još nekoliko pokušaja).

Potencijalni razlog tog neuspjeha je sljedeći. Metoda potencijalnih polja generira novu referentnu trajektoriju svakih 0,15 s koju bi mobilni robot preko analitičkog neizrazitog regulatora trebao pratiti tu trajektoriju namiještanjem vrijednosti linearne kutne brzine. U idealnom slučaju mobilni robot bi trebao na kraju upravljačkog kruga dostići zadanu referentnu trajektoriju. U potpoglavlju 5.1.3 ukazano je na kašnjenje mobilnog robota kod pokretanja, što onemogućava mobilnom robotu da prati generiranu trajektoriju. Pošto se referentna trajektorija povećava svakih 0,15 sekundi neovisno o položaju mobilnog robota, može se dogoditi da se referentna trajektorija postavi iza prepreke, što dovodi do sudaranja mobilnog robota sa preprekom.

Pošto je program testiran nekoliko puta sa različitim vrijednostima parametara, moguće je zaključiti kako bez dodatne optimizacije PD regulatora i analitičkog neizrazitog regulatora (što zahtjeva dodatan *hardware*), nije moguće uspješno realizirati klasičnu metodu potencijalnih polja.

Potencijani način rješenja tog problema je u generiranju repulzivnog potencijala u odnosu na položaj mobilnog robota. Za dovoljno velike vrijednosti  $k_r$  mobilni robot bi mogao zaobići prepreku, jer iako bi atraktivni potencijal već izgenerirao trajektoriju na cilj zbog jake repulzivne sile, referentna trajektorija bi se postavila ispred prepreke, što bi omogućilo mobilnom robotu da ju zaobiđe.

Primjer takvog rada je prikazan u videu [28]. Kao što se iz videa vidi mobilni robot je uspješno zaobišao prepreku i došao do cilja (iako zbog greške u komunikaciji između mobilnog robota i računala odzivi nisu snimljeni).

Zbog vremenskog ograničenja i zakaza *hardware*-a idejno rješenje nije detaljnije opisano.

## 6. ZAKLJUČAK

U ovom radu je analizirana reaktivna navigacija mobilnog robota. U tu svrhu su izrađena tri programa za odvojeno proučavanje parametara PID regulatora, analitičkog neizrazitog regulatora i metode potencijalnih polja. Također je izrađen mobilni robot sa diferencijalnim pogonom sa pet ultrazvučnih senzora za određivanje udaljenosti od prepreke, na kojemu je testiran rad programa.

Teoretske matematičke formule su morale biti prilagođene na rad u diskretnom vremenskom sustavu i prilagođene potrebama realnog sustava. Tako se, na primjer, parametri položaja mobilnog robota dobivaju preko podataka iščitanih sa enkodera, a za izračun repulzivne sile se koristila udaljenost mobilnog robota od prepreke. Izrađeni mobilni robot se sastoji od komponenata koje se mogu podijeliti na slijedeće podsustave: trup mobilnog robota, podsustav za kretanje, senzore, procesiranje, komunikaciju i napajanje. Zbog financijskog ograničenja, robot je napravljen od jednostavnih i jeftinih komponenti koje su dodatno otežale rad mobilnog robota i time onemogućili precizna mjerenja i uveli velike greške u sustav.

Kontrola brzine i položaja robota odvijala se preko enkodera. Upravljanje enkoderima je izvedeno preko *external interrupta* u režimu rada *RISING*. Taj režim rada je odabran zbog nejednake veličine proreza i punih dijelova na kotaču enkodera, zbog čega je u režimu rada *CHANGE* došlo do velikih oscilacija u mjerenjima. Bežična serijska komunikacija izvedena je preko dva *Xbee* uređaja. Zbog uštede energije serijska komunikacija je držana na najjednostavnijoj mogućoj razini. Kako bi se ubrzao rad programa, korišten je drugi mikrokontroler za prikupljanje podataka sa senzora. Za komunikaciju između dvaju mikrokontrolera koristio se *I2C/TWI* protokol, koji je omogućavao mobilnom robotu da u realnom vremenu prikuplja informacije sa mobilnog robota. Ultrazvučni senzori, kako ne bi došlo do krivih očitavanja, moraju raditi jedan iza drugoga sa malim zakašnjenjem.

Za regulaciju brzine kotača odabran je PD regulator. Uvođenjem predfiltracije kod velikih promjena ulaznih brzina došlo je do smanjenja preskoka referentnog stanja, čime je poboljšana stabilnost brzine vrtnje kotača. Pokazalo se da pokretni moment motora ima značajan utjecaj na performanse regulacije zbog pojave vremenskog pomaka i preskoka referentnog stanja. Moguće poboljšanje je određivanje minimalne brzine robota, kako bi se izbjeglo zaustavljanje pri malim vrijednostima ulaznih brzina. Mobilni robot je uspijevaao doći do referentne trajektorije uz pomoć analitičkog neizrazitog regulatora u sustavima koji nisu

zahtjevali prečesto zakretanje robota. Zbog lošeg čitanja enkodera, često je dolazilo do lošeg rada cijelog sustava (mobilnog robota) uslijed pogrešnog izračunavanja kuta zakreta.

Robot je uspio slijediti zadanu trajektoriju i zaobići prepreke koje su mu se našle na putu.

Zbog nepostojanja povratne informacije o položaju mobilnog robota, osim one iz enkodera, i relativno loših komponenti, rezultati testiranja nisu uvijek bili pouzdani. Pri dobrom radu komponenti, program je davao dobra rješenja, što je potvrdilo ispravnost programa i teoretske podloge korištene u radu.

Izrađeni programi su dobra osnova za daljnji razvoj reaktivne navigacije mobilnog robota. Kako bi se dobili bolji rezultati, od velike važnosti bi bilo uvesti dodatnu povratnu informaciju o položaju mobilnog robota, kao npr. troosni akcelerometar. Konstrukcija trupa mobilnog robota bi se trebala promjeniti kako bi se centar mase mobilnog robota nalazio na samom centru mobilnog robota na kojem bi se nalazili kotači. Za precizniju reaktivnu navigaciju bilo bi potrebno uvesti tri (umjesto dosadašnja dva) stražnja senzora koja bi bila pozicionirana isto kao i prednji.

## LITERATURA

- [1] Petrović, I.: Mobilna robotika-predavanja, Zagreb, 2004.
- [2] Velagić, J.: Mehatronika, predavanja, Sarajevo, 2012.
- [3] Tatalović, V.: Završni rad, Zagreb, 2013.
- [4] Kasać, J.: Neizravna logika, predavanja, Zagreb
- [5] Čužić, I.: Diplomski rad, Zagreb, 2013.
- [6] [http://free-zg.htnet.hr/Borko\\_Boranic/Izlazno\\_sucelje.htm](http://free-zg.htnet.hr/Borko_Boranic/Izlazno_sucelje.htm) (3.01.2015.)
- [7] <http://www.funnyrobotics.com/2011/03/arduino-with-l298n-based-dual-motor.html> (3.11.2015.)
- [8] Petrović, I.: Senzori za mobilne robote, predavanja, Zagreb, 2004
- [9] [http://abrobotics.tripod.com/Ebot/using\\_encoder.htm](http://abrobotics.tripod.com/Ebot/using_encoder.htm) (4.01.2015.)
- [10] [http://wiki.iteadstudio.com/Ultrasonic\\_Ranging\\_Module\\_HC-SR04](http://wiki.iteadstudio.com/Ultrasonic_Ranging_Module_HC-SR04) (4.01.2015.)
- [11] [http://www.dfrobot.com/wiki/index.php/Romeo\\_V2-All\\_in\\_one\\_Controller\\_\(R3\)\\_\(SKU:DFR0225\)#Specification](http://www.dfrobot.com/wiki/index.php/Romeo_V2-All_in_one_Controller_(R3)_(SKU:DFR0225)#Specification) (4.01.2015.)
- [12] <http://arduino.cc/en/Main/arduinoBoardLeonardo> (4.01.2015.)
- [13] <http://arduino.cc/en/Main/arduinoBoardDue> (4.01.2015.)
- [14] <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide> (4.01.2015.)
- [15] <https://learn.sparkfun.com/tutorials/using-the-logic-level-converter> (4.01.2015.)
- [16] <http://arduino.cc/en/Reference/attachInterrupt> (4.01.2015.)
- [17] <http://www.otpornik.com/elektronika/motori/osnovno-o-impulsno-sirinskoj-modulaciji.html> (4.01.2015.)
- [18] <http://forums.trossenrobotics.com/tutorials/how-to-diy-128/complete-control-of-an-arduino-via-serial-3300/> (4.01.2015.)
- [19] Tatalović, V.: Seminarski rad iz kolegija Neizravna i Digitalno Upravljanje, Zagreb, 2014.
- [20] <http://youtu.be/U8CG3KMZK4Y> (12.01.2015)
- [21] <http://youtu.be/DnLB-6SVn-k> (12.01.2015)
- [22] <http://youtu.be/GV6udGMsScA> (12.01.2015)
- [23] <http://youtu.be/18iPRB4wXQs> (12.01.2015)
- [24] <http://youtu.be/tf4gsJCw2Iw> (12.01.2015)

- [25] <http://youtu.be/v-LNdyx76Dk> (12.01.2015)
- [26] <http://youtu.be/wG2ayzkHyAc> (12.01.2015)
- [27] <http://youtu.be/-YGt2XKDXNo> (12.01.2015)
- [28] [http://youtu.be/0\\_4wFr4K5N0](http://youtu.be/0_4wFr4K5N0) (12.01.2015)



## **PRILOZI**

### I. DVD-R disc