

# Kretanje mobilnih robota u zadanoj formaciji

---

**Zrinušić, Ivan**

**Undergraduate thesis / Završni rad**

**2022**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:235:280005>

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

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

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Ivan Zrinušić**

Zagreb, 2022. g

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

**ZAVRŠNI RAD**

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

Student:  
Ivan Zrinišić

Zagreb, 2022.g.

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

Zahvaljujem se mentoru prof. dr. sc. Mladenu Crnekoviću na ukazanoj stručnoj pomoći u izradi ovog završnog rada, kao i na iznimnoj ljubaznosti i strpljivosti u rješavanju problema, i upita.

Također, zahvaljujem se asistentu Marinu Lukasu na pomoći pri razjašnjavanju pojedinih funkcija u Pythonu.

Zahvaljujem se svojoj obitelji koji su mi bili podrška i oslonac tijekom cijelog preddiplomskog studija.

Ivan Zrinušić



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**

Središnje povjerenstvo za završne i diplomske ispite

Povjerenstvo za završne i 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: 602 – 04 / 22 – 6 / 1	
Ur.broj: 15 - 1703 - 22 -	

## ZAVRŠNI ZADATAK

Student: **Ivan Zrinišić** JMBAG: **0035220090**

Naslov rada na hrvatskom jeziku: **Kretanje mobilnih robota u zadanoj formaciji**

Naslov rada na engleskom jeziku: **Mobile Robots Movement in a Given Formation**

Opis zadatka:

Prvi korak u koordinaciji više mobilnih robota je vožnja u zadanom rasporedu. Održavanje formacije više robota preduvjet je za ostala složenija djelovanja (npr. zajedničko prenošenje istog predmeta).

Potrebno je pronaći komercijalne primjene mobilnih robota koji rade u zadanom rasporedu. Na eMiR mobilnim robotima ostvariti vožnju u formaciji za najmanje dva robota u rasporedu jedan iza drugog.

U radu je potrebno:

- definirati teoretske osnove vožnje mobilnih robota u formaciji,
- predložiti algoritam vođenja za formaciju najmanje dva eMiR mobilna robota,
- na poligonu eksperimentalno verificirati predloženi algoritam.

Potrebno je navesti korištenu literaturu, te eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2021.

Zadatak zadao:

Prof. dr. sc. Mladen Crneković

Datum predaje rada:

1. rok: 24. 2. 2022.  
 2. rok (izvanredni): 6. 7. 2022.  
 3. rok: 22. 9. 2022.

Predvideni datumi obrane:

1. rok: 28. 2. – 4. 3. 2022.  
 2. rok (izvanredni): 8. 7. 2022.  
 3. rok: 26. 9. – 30. 9. 2022.

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

**SADRŽAJ**

SAŽETAK .....	1
<b>1. UVOD</b> .....	<b>3</b>
<b>2. TEORIJSKA OSNOVA FORMACIJE</b> .....	<b>5</b>
2.1 Metoda temeljena na ponašanju .....	7
2.2 Metoda voditelj-pratilač .....	8
2.3 Metoda virtualne strukture .....	8
2.4 Ostale metode .....	9
2.4.1 Metoda razmještaja .....	9
2.4.2 Metoda udaljenosti .....	10
<b>3. IMPLEMENTACIJA I PRIMJENA</b> .....	<b>11</b>
3.1 CAV metoda upravljanja .....	11
3.1.1 FC metoda upravljanja .....	12
3.1.2 ACC sustavi .....	12
3.2 Poljoprivreda .....	14
3.3 Logistika .....	15
<b>4. eMiR ROBOT</b> .....	<b>16</b>
4.1 Energetski sustav robota .....	16
4.2 Sustav za mjerenje .....	16
4.3 Upravljački sustav robota .....	16
4.4 Infracrveni senzori .....	19
<b>5. ALGORITAM UPRAVLJANJA</b> .....	<b>20</b>
5.1 Oprema .....	23
5.1.1 Arduino Mega 2560 .....	23
5.1.2 Bluetooth modul WRL 00582 .....	25
<b>6. ZAKLJUČAK</b> .....	<b>26</b>
LITERATURA .....	27
PRILOZI .....	28

**POPIS SLIKA**

Slika 1 Primjer mobilnog robota za detekciju mina.....	4
Slika 2 Mobilni roboti u formaciji .....	6
Slika 3 Metoda temeljena na ponašanju.....	7
Slika 4 Metoda voditelj-pratilac.....	8
Slika 5 Metoda virtualne strukture s dva robota .....	9
Slika 6 Prikaz FC metode .....	12
Slika 7 Primjena formacijskih gibanja u poljoprivredi .....	15
Slika 8 Mobilni roboti u skladištima.....	15
Slika 9 Prikaz upravljačkog sustava robota .....	16
Slika 10 Oblik naredbi robotu.....	17
Slika 11 Roboti eMiR .....	18
Slika 12 Karakteristika senzora .....	19
Slika 13 Prikaz gibanja mobilnih platformi .....	20
Slika 14 Tijek upravljanja formacijom robota.....	21
Slika 15 Prikaz mjernog sustava robota.....	22
Slika 16 Gibanje robota pri referentnoj brzini od 20 % .....	22
Slika 17 Gibanje robota pri referentnoj brzini od 80 % .....	23
Slika 18 Arduino ATmega 2560 pločica.....	25

## POPIS TABLICA

Tablica 1 eMiR naredbe.....	18
Tablica 2. Specifikacije ATmega 2560 pločice .....	21



## SAŽETAK

U ovom radu izvesti ćemo algoritam za gibanje u formaciji dvaju edukativnih mobilnih robota konstruiranih na Katedri za strojarску automatiku. Potencijalno rješenje problema formacijskog gibanja mobilnih robota je osnova za efikasnije i učinkovitije obavljanje mnogih poslova. Kretanje vozila u formacijama je već dulje vrijeme zanimljivo područje proučavanja jer pronalazi potencijalnu primjenu u vojsci, prometu i poljoprivredi. O razvijanju područja gibanja vozila u formacijama najbolje govori project SARTRE ( Safe Road Trains for the Environment) čiji je cilj omogućiti gibanje osobnih automobile u formaciji na autocestama. Projekt je pokrenut 2009. godine uz pomoć Europske unije. Vrijednost samog projekta iznosi 6.4 milijuna eura. Princip tehnologije slijeđenja u današnjem svijetu vidimo sve više. U radu su pokrivenе također i teorijske osnove ostvarivanja gibanja u formaciji te su isti primjeri prikazani u stvarnom svijetu.

Ključne riječi: eMiR, programiranje, mobilni roboti, formacija, konvoj

## **SUMMARY**

In this paper, we will perform an algorithm for the motion in the formation of two educational mobile robots constructed at the Department of Mechanical Automation. A potential solution to the formation motion problem of mobile robots is the basis for more efficient and effective performance of many jobs. The movement of vehicles in formations has been an interesting area of study for a long time because it finds potential applications in the military, transport and agriculture. The SARTRE (Safe Road Trains for the Environment) project, whose goal is to enable the movement of passenger cars in formation on highways, best describes the development of the area of vehicle movement in formations. The project was launched in 2009 with the help of the European Union. The value of the project itself is 6.4 million euros. We are seeing more and more of the principle of tracking technology in today's world. The paper also covers the theoretical foundations of movement in formation, and the same examples are presented in the real world.

Key words: eMiR, programming, mobile robots, formation, convoy

## 1. UVOD

S obzirom na razvoj robotike kao i područja senzorike, pojavljuju se nove izvedbe robotskih sustava te dolazi do otvaranja novih neistraženih područja. Mnogi proizvođači robota u periodu industrijalizacije ulažu više u razvoj robotike. Za razliku od fiksiranih industrijskih robota, mobilnom robotu pokretljivost nije ograničena. Također, mogu raditi u velikom radnom prostoru i istraživati nepoznatu okolinu te izvoditi zadatke kada je to potrebno. Koriste se u poslovima koje obavljaju ljudi ili humanoidni roboti kao što su nadzor, istraživanje, kućna sigurnost, pomoć u domaćinstvu itd. U sklopu ovog rada razvijen je algoritam za pokretanje formacije dvaju eMiR robota, uz potreban kod za komunikaciju s robotima koji je napisan u programskom jeziku Python. U radu će biti prikazana i razrađena teorija upravljanja formacijom grupe robota, među kojima je i strategija slijeđenja robota, prikazano je na koji način ostvarujemo strategije, uz razradu prednosti i nedostataka svake strategije. Mnogi inženjeri su pokušali riješiti praćenje i određivanje putanje robota, no idealno rješenje nije pronađeno. Suradnja između više autonomnih mobilnih robota bitan je preduvjet za uspješno obavljanje mnogih zadataka koji zahtijevaju više resursa nego što jedan robot može pružiti. Postoje mnoge koordinirane misije kao što su kontrola formacije, koordinirano istraživanje itd. Problem kontrole formacije sastoji se od dizajniranja upravljačkih ulaza za vozila tako da formiraju i održavaju unaprijed definirani geometrijski oblik u prostoru čak i uz prisutnost vanjskih smetnji. Kontrola je koncipirana na način da vozila korigiraju svoj položaj gledajući položaj svojih susjeda. Primjer takvog jednog zadatka je uklanjanje mina iz minskog polja. Razvoj gibanja robota u formaciji u tom slučaju bi spasio mnoge živote. Također, gibanje robota u formaciji bi olakšalo mnoge poslove poput transporta, poljoprivrede, skladištenja. Naravno u sve to se mora uzeti ekonomski aspekt, no uz dobro implementirane algoritme gdje ne bi nailazili na probleme u upravljanju sama izvedba postaje ekonomski isplativa.



**Slika 1** Primjer mobilnog robota za detekciju mina

## 2. TEORIJSKA OSNOVA FORMACIJE

Formacija robota može biti okarakterizirana kao grana robotike koja proučava koordinaciju grupe mobilnih robota kako bi formaciji dala određeni oblik. Takva formacija se koristi za izvedbu zadataka. Grupni poslovi više robot uključuju transport, istraživanje i nadzor. Svi ovi zadatci imaju svoju namjenu u današnjem svijetu kao što su spasilačke misije u sudarima ili poljoprivreda. U prethodnim godinama mobilni roboti su bili korišteni u polju mobilne senzorske mreže i medicinskim operacijama. Svi ovi zadatci su vrlo kompleksni i teški za izvođenje. Naravno primjenu možemo naći u granama industrije kod kojih njihova upotreba olakšava rad kao npr. u transportu objekata, u skladištima, medicini, kontroli sigurnosti i drugim bitnim aspektima današnjeg života. Najveći izazov je pronalazak najboljeg algoritma koji sa sobom donosi još nekoliko problema. Prvi problem koji je potrebno riješiti je početna pozicija robota, nakon toga putanju od početne do konačne pozicije u formaciji. U potonjem procesu treba uzeti u obzir i moguću koliziju s objektima na putanji. Za svakog robota možemo definirati dvije udaljenosti u odnosu na ostale u formaciji:

- Longitudinalna udaljenost- udaljenost po osi gibanja
- Lateralna udaljenost-udaljenost po osi okomitoj na os gibanja

Po međusobnom odnosu u formaciji između robota razlikujemo:

- Linijsku formaciju- formacije kod kojih je longitudinalna udaljenost jednaka nuli dok su roboti na određenoj bočnoj udaljenosti
- Konvoj formacija- kod ove formacije su mobilni roboti poredani jedan iza drugoga, odnosno bočna udaljenost je jednaka nuli
- Proizvoljna udaljenost

Za potrebe pozicioniranja u lokalnom okruženju razvijeni su različiti sustavi lokalizacije, a za tu primjenu koriste se različiti senzorski sustavi poput infracrvenih daljinomjera, ultrazvučnih senzora, sonara i laserskih daljinomjera. Procesi planiranja putanje su važan aspekt gibanja robota u formaciji. Ovi procesi se dijele na globalne i lokalne u odnosu na informaciju sa senzora.

Globalna metoda garantira nam da će robot stići do svoga cilja.

Formacija robota može biti podijeljena u dva dijela:

1. Grupiranje
2. Upravljanje formacijom

Upravljanje formacijom znači stabiliziranje i zadržavanje željenog oblika poput formacije prikazane na Slici 2.



**Slika 2 Mobilni roboti u formaciji**

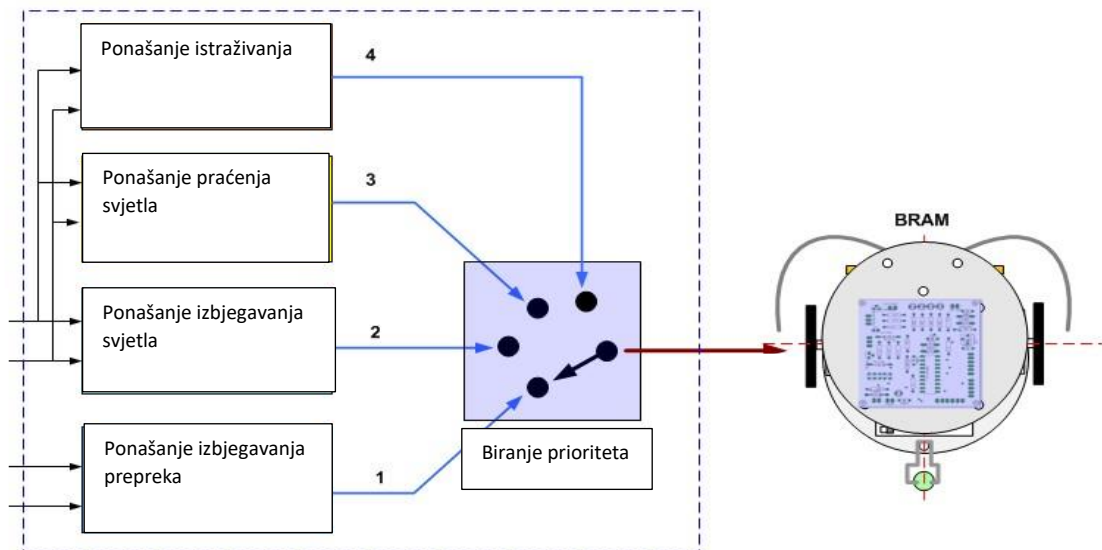
Mnoga istraživanja u robotici su se svodila na kontrolu formacije, koja se definira kao kontrola grupe robota te izbjegavanje kolizija uz zadržavanje željenog oblika. Uzevši sve u obzir, postoje tri različita pristupa kontroli formacije:

1. Metoda temeljena na ponašanju
2. Metoda voditelj-pratilac
3. Metoda virtualne strukture

## 2.1 Metoda temeljena na ponašanju

Metoda temeljena na ponašanju dijeli kompleksni navigacijski zadatak na više podzadataka. Svaki podzadatak je samostalni kontrolni modul koji se bavi određenim problemom navigacije.

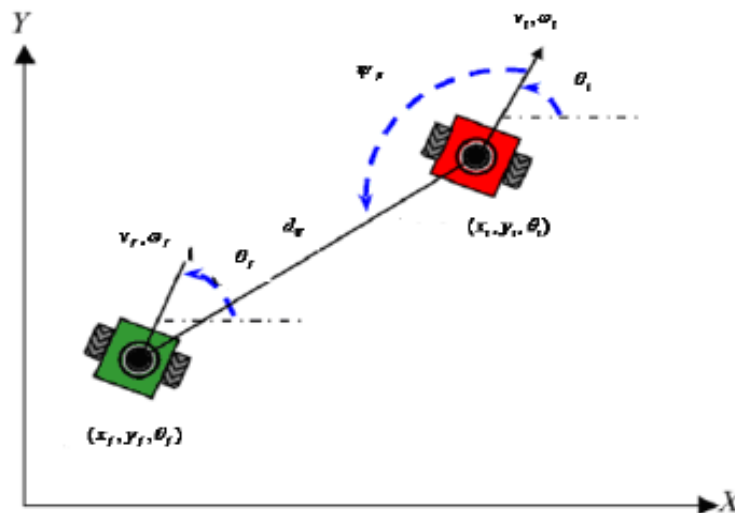
Svaki robot se upravlja sam, tako da pazi na faktore koji utječu na idući korak u izvođenju formacije. Određivanje putanjeza svakog robota dobije se izračunavanjem srednje vrijednosti faktora za izbjegavanje prepreka, istraživanje formacije te prilaz cilju. Metoda temeljena na ponašanju je sada generalno prihvaćena kao efektivna za mobilne robote visokih performansi. Učenje u domeni mobilnih robota je izuzetno teško, pogotovo u nestacionarnim uvjetima. Metoda temeljena na ponašanju se pokazala korisnom za robote koji rade u stvarnim uvjetima. Kontroler robota se sastoji od grupe ponašanja (behaviours), svaki sa svojim različitim zadatkom. Naprimjer, ponašanje „izbjegavanje objekta“ ima zadatak sprečavanja kolizije s objektima. Ponašanja se aktiviraju odgovarajući na vanjske ili unutarnje uvjete, stanje senzora i vanjski sustav. Ponašanje robota čini skupovi pod-ponašanja, a skup pod-ponašanja se sastoji od vektora iznosa i smjera. Težina vektora se mijenja ovisno o težinskim parametrima. S obzirom na dobivene vanjske informacije robot bira odgovarajuće ponašanje, te se u skladu s njim giba.



Slika 3 Metoda temeljena na ponašanju (engl. Behaviour Based Approach)

## 2.2. Metoda voditelj-pratilac

U ovom pristupu podrazumijevamo da jedan robot (voditelj), ima informacije o referentnoj trajektoriji gibanja. Nadalje, svaki robot pratilac dobiva informacije od voditelja. Sa tehnološkog gledišta ovo ima direktni utjecaj na jednostavnost izvedbe jer je potreban manji broj senzora. Prednost ove metode je što je također jednostavna za izvedbu i razumijevanje. Dakako, ona ima i svoje nedostatke jer zahtijeva centralizirani pristup koji postaje nepogodniji za više robota. Ako se robot voditelj giba prebrzo ili ako je pratilac doživio koliziju s nekim objektom formacija se raspada. Kontroler zahtijeva samo vizualnu informaciju s kamere montirane na bilo kojoj poziciji na robotu voditelju. Također, ovaj pristup ne zavisi o mjerenju relativnog položaja i komunikaciji između voditelja i pratioca. Naravno, algoritam mora biti dobro napisan kako se ne bi izgubio kontakt između dva robota.



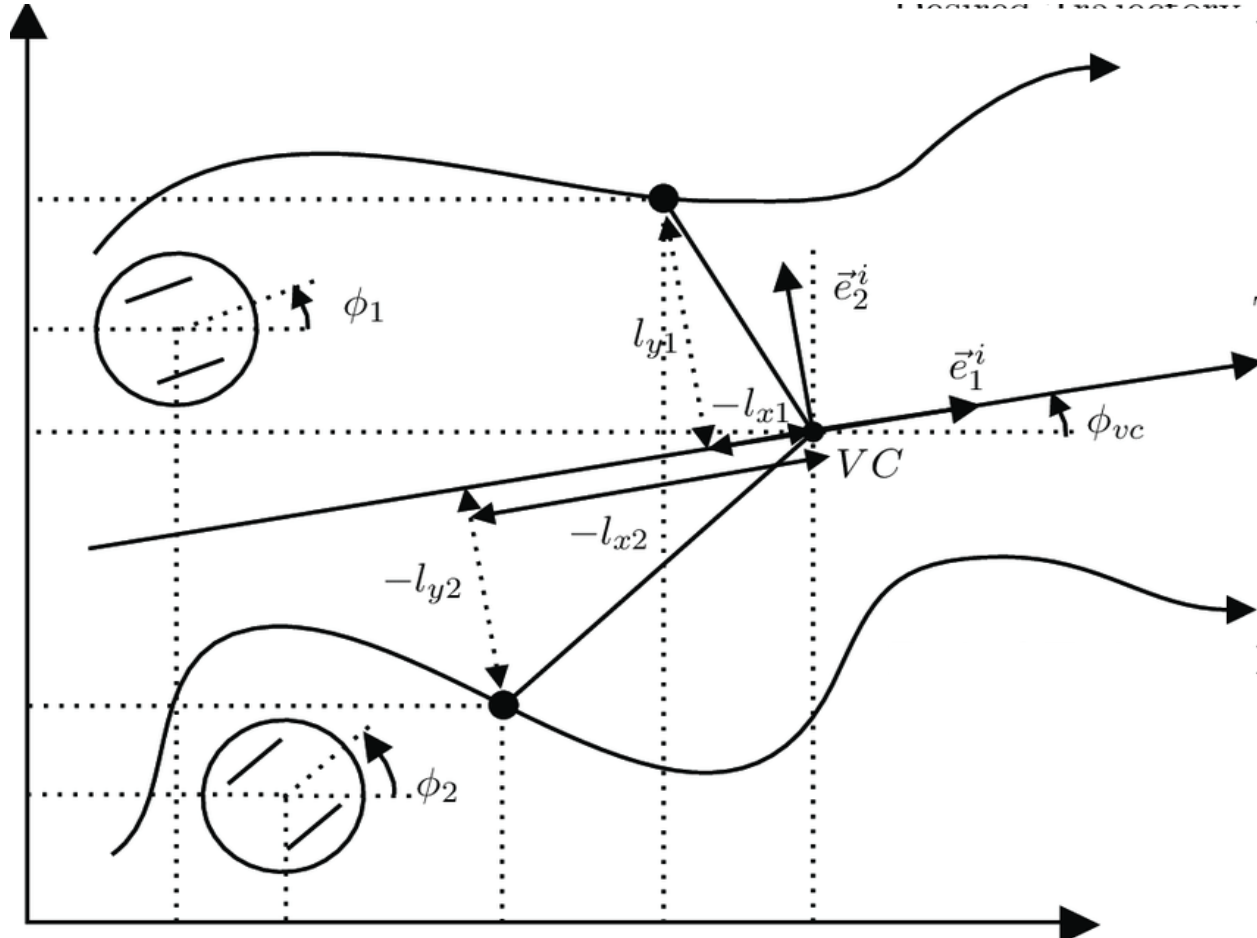
Slika 4 Metoda voditelj-pratilac (engl. Leader-Follower Approach)

## 2.3 Metoda virtualne strukture

Princip metode virtualne strukture se svodi na to da svaki robot dobije set kontrola kojima prati željenu putanju kao kruto tijelo. Najveća prednost ove metode je, što za razliku od prethodnih metoda, ima unaprijed definiranu formaciju. Također, metodom virtualne strukture možemo dobro



održavati formaciju za više robota te precizno pratiti već predefimirane putanje.



Slika 5 Metoda virtualne strukture s dva robota

## 2.4 Ostale metode

### 2.4.1 Metoda razmještaja

Metoda kod koje roboti kontroliraju pomicanje susjednih robota kako bi dobili željenu formaciju. Formacija je određena željenim pomacima u odnosu na globalni koordinatni sustav pod pretpostavkom da svaki robot može „osjetiti“ relativni položaj susjednog robota u odnosu na globalni koordinatni sustav. To znači da roboti moraju znati orijentaciju globalnog koordinatnog sustava te poziciju u odnosu na njega. ([2] A Survey of Multi-mobile Robot Formation Control , Abdulmuttalib Turkey Rashid)

### **2.4.2 Metoda udaljenosti**

Radi na principu upravljanja unutarnjim udaljenostima robota kako bi se dobila željena formacija. Pretpostavljamo da su roboti u mogućnosti mjeriti udaljenost susjednog robota u odnosu na svoj lokalni koordinatni sustav. Roboti ne moraju biti pravilno raspoređeni jedan u odnosu na druge. ([2] A Survey of Multi-mobile Robot Formation Control , Abdulmuttalib Turkey Rashid)

### 3. IMPLEMENTACIJA I PRIMJENA

Iz prethodnog poglavlja vidljivo je koliko je sama primjena mobilnih robota kompleksna. Unatoč tome u stvarnom svijetu postoji nekoliko primjena, posebno u transportu i automobilizmu. Postojeće istraživanje pokazalo je da gibanje u formaciji može pridonijeti efikasnosti i sigurnosti prometa. Postojeće metode kontrole formacije unaprijed definiraju proces promjene formacije i nisu detaljno razmotrile niti objasnile ponašanje vozila bez sudara. U daljnjem radu pobliže će biti obrađena strategija upravljanja formacijom s više trakova za povezana i automatizirana vozila (CAV).

#### 3.1. CAV metoda upravljanja

Postojeća metoda upravljanja formacijom (CAV) je pokazala da može unaprijediti sigurnost, efikasost i smanjiti potrošnju goriva u odnosu na automatsko upravljanje samo jednim vozilom. Većina metoda koordinacije s više vozila fokusira se samo na scenarij s jednim trakom gdje promatramo samo uzdužno ponašanje vozila i raskrižja. Kontrola konvoja vozila jedna je od najčešćih metoda u scenarijima s jednim trakom. U području koordinacije s više traka tri su uobičajena problema koja se moraju riješiti:

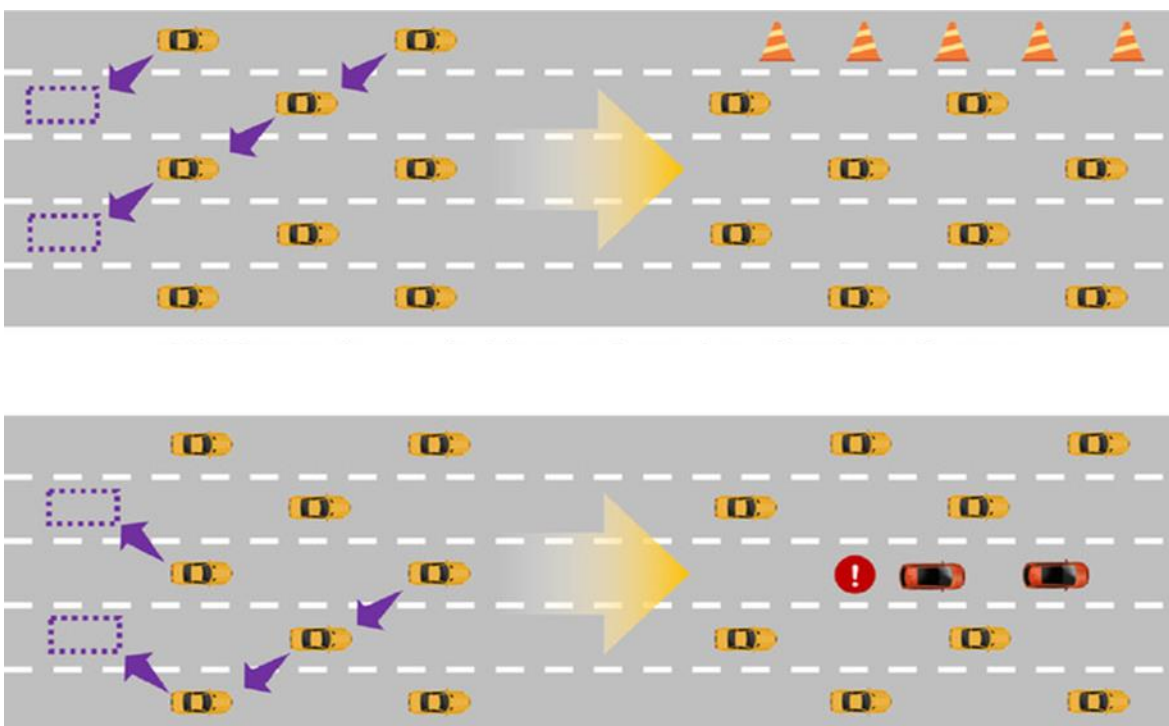
- Dodjela trake
- Spajanje rampi
- Uska grla pri izlasku iz trake

Kod dodjele traka konačni cilj je da se poboljša kapacitet i popunjenost traka na autocestama. Prostornim rasporedom vozila na voznoj traci upravlja se tako da se spriječi ili smanji prometna gužva. Neke metode dodjele traka dodjeljuju trake za pojedinačna vozila, dok druge razmatraju vozila u nizovima i dodjelju trake za takve formacije. Vozila moraju sama izvršiti planiranje uzdužnog i poprečnog gibanja, a tijekom procesa promjene trake može doći do kolizije što ograničava učinkovitost prometa. Metoda spajanja rampi i koordinacije uskih grla pri izlasku iz trake imaju za cilj riješiti takve probleme. Razlika između ova dva scenarija je u tome što je vozilima dopušteno mijenjati vozne trake na bilo kojoj točki bez sudara u scenarijima uskog grla izlaska iz trake, ali moraju čekati do točke spajanja na prilazu. Kontrola formacije na autocesti

(FC) poseban je slučaj kooperativnog planiranja putanje više vozila u scenarijima s više traka koji razmatra grupu autonomno upravljanih vozila i planira njihovo kretanje iz ukupne perspektive.

### 3.1.1 FC metoda upravljanja

U usporedbi s drugim metodama koordinacije s više prometnih traka koje su usredotočene na lokalno izbjegavanje sudara, FC metoda uzima u obzir globalnu koordinaciju i izbjegavanje kolizija. FC metoda usmjerava vozila da voze kao grupa i adaptivno mijenjaju strukturu formacije u skladu s okolinom, što omogućuje primjenu ove metode u više scenarija.



Slika 6 Prikaz FC metode

### 3.1.2 ACC sustavi

Sustav adaptivnog tempomata ( Adaptive cruise control ) je sustav koji automatski upravlja ubrzanjem i kočenjem vozila. Aktivira se preko upravljača i prekida se kočenjem ili pritiskom na gumb. ACC sustav radi na temelju informacija dobivenih od senzorskih sustava koji se sastoje od

radarskih ili laserskih senzora, ili vizijskih sustava koji detektiraju vozilo ispred i smanjuju brzinu. Neke od prednosti takvih sustava jesu da poboljšavaju sigurnost na cesti, tako što vozila opremljena s ovim sustavom drže potreban razmak između drugih vozila. Također, vozač se ne mora fokusirati na brzinu, jer ju sustav automatski regulira te se može usredotočiti na ono što se događa oko njega. Dakako, ovaj sustav ima i neka ograničenja. Jedno od najvećih ograničenja je da sami sustav nije potpuno autonoman jer vozač i dalje mora prakticirati navike sigurne vožnje kako bi sustav dao najbolje rezultate. Nadalje, promjenjivi vremenski uvjeti poput snijega, kiše ili magle bi mogli zbuniti senzore i dati pogrešna očitavanja. Naime, postoje i razine autonomnih auta. Prva razina je ona koja ima sustav adaptivnog tempomata dok druga razina posjeduje i kontrolu trake.

ACC sustave dijelimo na:

- Sustavi zasnovani na radaru-rade tako što postavimo radarske senzore na vozilo koji detektiraju okruženje. Skup radara daje potpunu sliku o blizini drugih vozila i ostalih potencijalnih opasnosti. Ova vrsta senzora može izgledati različito u ovisnosti o dizajnu i modelu vozila.
- Sustavi zasnovani na laseru-ovaj sustav radi na temelju velike crne kutije postavljene u rešetci vozila. Koristi lasersku tehnologiju da detektira blizinu objekata. Ne radi dobro u promjenjivim vremenskim uvjetima.
- Dalekozorni računalni sustav vida (optički)-koristi malu kameru postavljenu na stražnji dio retrovizora vozila za otkrivanje objekata okrenutih naprijed.
- Sustavi s više senzora-u ACC sustave ponekada ugrađujemo više vrsta senzora kako bi olakšali rad sustava. Senzori u ovakvim sustavima mogu sadržavati podatke sa GPS opreme ili kamere koje sadržavaju informacije o geografskoj okolini.
- Prediktivni sustavi-koriste podatke sa senzora kako bi predvidili ponašanje susjednih vozila. U principu, auto će usporiti ako susjedno mijenja trake te tako povećava sigurnost putnika.

ACC sustavi napreduju iz godine u godinu. Automobilske kompanije kontinuirano rade na usavršavanju ove tehnologije te tako stvaraju cijenom prihvatljivije modele koji bi se jednog dana mogli naći u svim vozilima.

### 3.2. Poljoprivreda

Unatoč činjenici da je gibanje u formaciji najzastupljenije u prometu i transportu, nalazimo ga i u ostalim granama industrije. Razni poslovi u poljoprivredi poput obrade zemlje, žetve, zalijevanja mogu biti obavljani pomoću mobilnih robota. Ako želimo da isti poslovi budu efikasnije obavljani možemo primijeniti gibanje u formaciji. Poljoprivredni roboti mogu zamijeniti tradicionalne ljudske napore tako da se uključe u sve vrste radno intenzivnih i kompliciranih poljoprivrednih aktivnosti. Nadalje, njihovo korištenje može smanjiti pad proizvodnje uzrokovan nepravilnim ljudskim radom, nesarom te smanjuje mogućnost ozljede samih radnika. Kako bi se prilagodili sve većem opsegu proizvodnje i zadovoljili potrebe društvenog razvoja potrebno je koristiti tehnologiju višestrukih robotskih sustava. U ovisnosti o dijeljenju informacija sa susjednim robotima postoje tri različita tipa pozicioniranja:

- Aktivno pozicioniranje-u nedostatku informacija od drugih robota oslanja se na informacije sa senzora kako bi odredio svoju putanju.
- Pasivno pozicioniranje-komunikacija između robota postoji te se pozicioniranje provodi oslanjajući se na susjedne robote.
- Interaktivno pozicioniranje

U posljednjem desetljeću, poljoprivredni sustavi s više robota su se usredotočili na centraliziranje ili distribuirane arhitekture za postizanje suradničkih operacija pod unaprijed definiranim uvjetima. Često je potrebno koordinirati dva ili više robota kako bi povećali efikasnost obavljanja nekog posla. Primjerice, pri žetvi robotski kombajn u svojoj pratnji treba imati robota koji će obaviti transport. Drugi primjer bi bila suradnja dva traktora, od kojih bi vodeći kosio travu dok bi traktor koji ga slijedi izvršio radnju kupljenja pokošene trave. Sustav s više robota je kompleksan sustav te nepravilno izvođenje bi samo dovelo do komplikacija ako je isti posao moguće obaviti sa samo jednim robotom.



**Slika 7 Primjena formacijskih gibanja u poljoprivredi**

### **3.3. Logistika**

Mobilni roboti mogu biti izvrsno rješenje za mnoge probleme u logističkoj industriji. Posebno uzbudljiv podskup toga je korištenje mobilnih robota, autonomnih vozila i bespilotnih letjelica za automatizaciju zadataka temeljenih na kretanju. Ovo područje obuhvaća sve vrste mobilnih robotskih uređaja koji se koriste u logistici, kao što su robotska kolica/vozila, autonomni kamioni na cesti i bespilotne letjelice, koji pomažu robi na njenom putu od polazišta do odredišta. Radi povećanja efikasnosti u skladištima se koriste gibanja u formaciji. Velike tvrtke poput Amazona ulažu sve više u svoje pogone koji rade na tom principu.



**Slika 8 Mobilni roboti u skladištima**

## 4. eMiR ROBOT

Praktični dio u ovom radu biti će izvršen pomoću eMiR robota konstruiranih na Katedri za strojarstvu automatiku. Izrađena su tri identična robota označena sa žutom, plavom i crvenom bojom. Kretanje robota je ostvareno pomoću tzv. diferencijalne strukture. Duljina robota iznosi 300 mm, a širina 250 mm. Roboti su izrađeni od pleksiglasa, a debljina donje ploče je 10 mm. Izrađena je i gornja ploča od istog materijala debljine 6 mm tako da su sve komponente zaštićene. Masa robota iznosi 3,5 kg, a napaja se standardnom baterijom od 12 V/2 Ah koja se može puniti. Ako uzmemo u obzir maksimalnu brzinu motora, maksimalna brzina robota iznosi 0.5 m/s, a maksimalna brzina vrtnje 240 °/s. Kotači su prekriveni gumom kako bi se spriječilo klizanje robota.

### 4.1 Energetski sustav robota

Robot se pokreće pomoću dva istosmjerna motora s mehaničkim prijenosim omjerom 1:27. Maksimalni okretni moment iznosi 0,14 Nm. Sami motori mogu raditi na 24 V, ali u samom radu ih koristimo na 12 V jer time eliminiramo potrebu za hlađenjem. Frekvencija PWM signala iznosi 3,6 kHz a maksimalna struja baterije sa svim uključenim senzorima i kamerom iznosi 450 mA.

### 4.2 Sustav za mjerenje

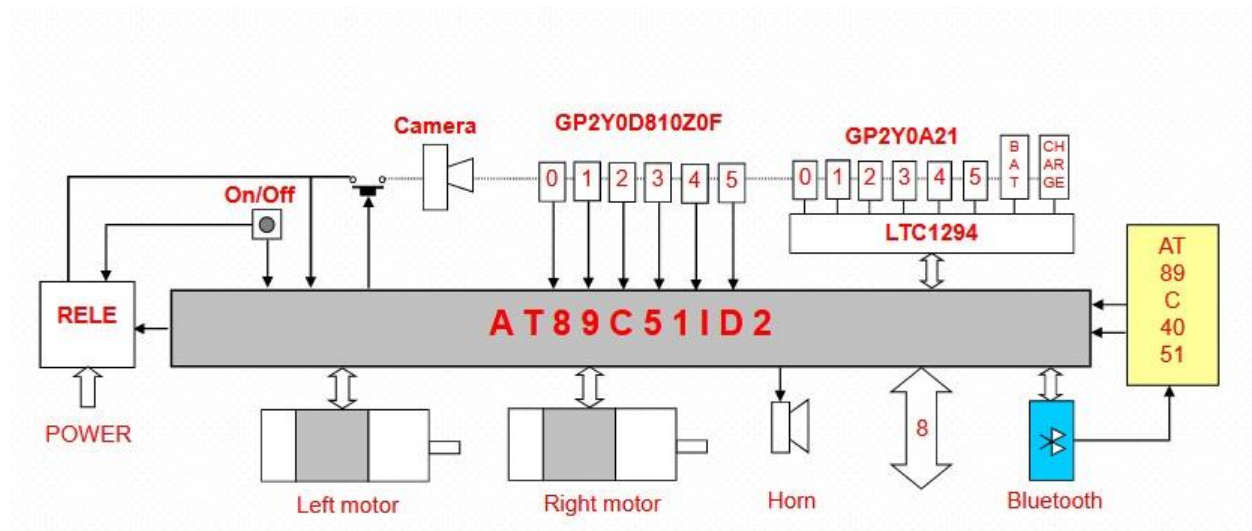
Enkoderi koji su ugrađeni u motore koriste se za svako mjerenje brzine kotača. Svaki enkoder ima dva kanala sa sedam impulsa po okretaju. Množenje impulsa enkodera i prijenosnog omjera mjenjača daje 189 impulsa po okretaju izlazne osi mjenjača. Ako je kretanje robota samo translatorno, tada jedan impuls iznosi 1.33 mm. Sličnom matematikom primjenjenom na rotaciju robota dobijemo rezultat od 556 impulsa enkodera za rotaciju od 360 °.

### 4.3 Upravljački sustav

Upravljačku jedinicu čini mikrokontroler Atmel AT89C51ID2. Ovaj mikrokontroler ima 64 kB programske memorije i frekvenciju od 1 MHz što je sasvim dovoljno za funkcioniranje robota. Funkcionalna shema samog sustava je prikazana na Slici 9. Bežična komunikacija se provodi preko Bluetooth modula Sparkfun WRL-00582 s brzinom prijenosa podataka od 57600 bps. Kako mikrokontroler nema A/D pretvarač koristi se vanjski pretvaral LTC1294. Napajanjem robota se

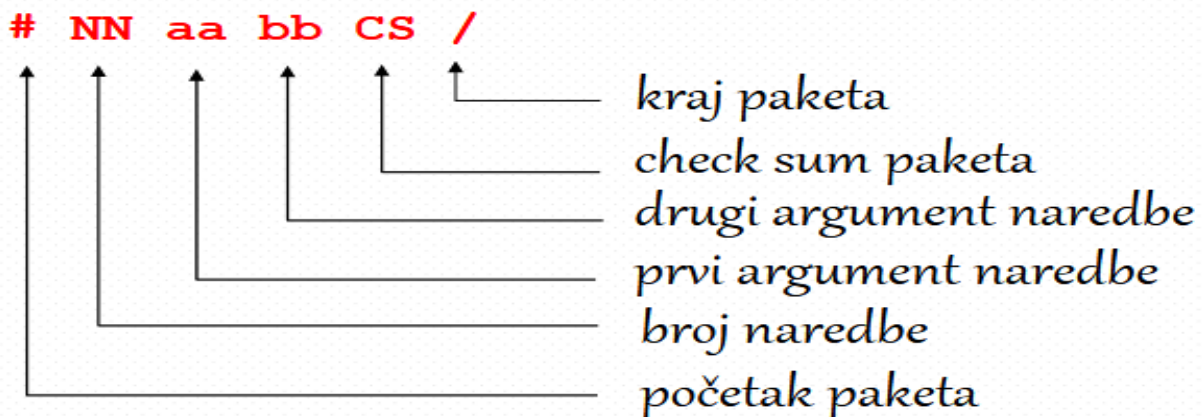


upravlja pomoću bistabilnog releja. Promjena upravljačkog programa provodi se preko Bluetooth bežične komunikacije.



Slika 9 Prikaz upravljačkog sustava robota

Kao što je rečeno, naredbe robotu šalju se bluetooth komunikacijom u obliku paketa točno definiranog oblika koji sadrži oznaku početka naredbe(#), zatim broj naredbe s popratnim argumentima naredbe zapisanim u heksadekadskom sustavu, check sum paketa za provjeru ispravnosti pročitane naredbe, te na kraju oznaku za kraj paketa(/). (iz [1]- Prof.dr.sc. Mladen Crneković- educational Mobile Intelligent Researcher)



Slika 10 Oblik naredbi robotu

U donjoj tablici prikazane su neke naredbe robotu:

NN	Paket	Opis
0	# 00 00 00 /	Zaustavi robota i prekini tekući posao
1	# 01 vv rr CS /	Postavi robota na zadanu brzinu vv i rr(-100 do 100%)
4	# 04 tt 00 CS /	Uključi sirenu tt desetinki sekunde
5	# 05 aa 00 CS /	Uključi(01)/isključi(00) senzore i kameru
255	# FF 00 00 01 /	Isključi robota

**Tablica 1 eMiR naredbe**

(iz [1]- Prof.dr.sc. Mladen Crneković- educational Mobile Intelligent Researcher)

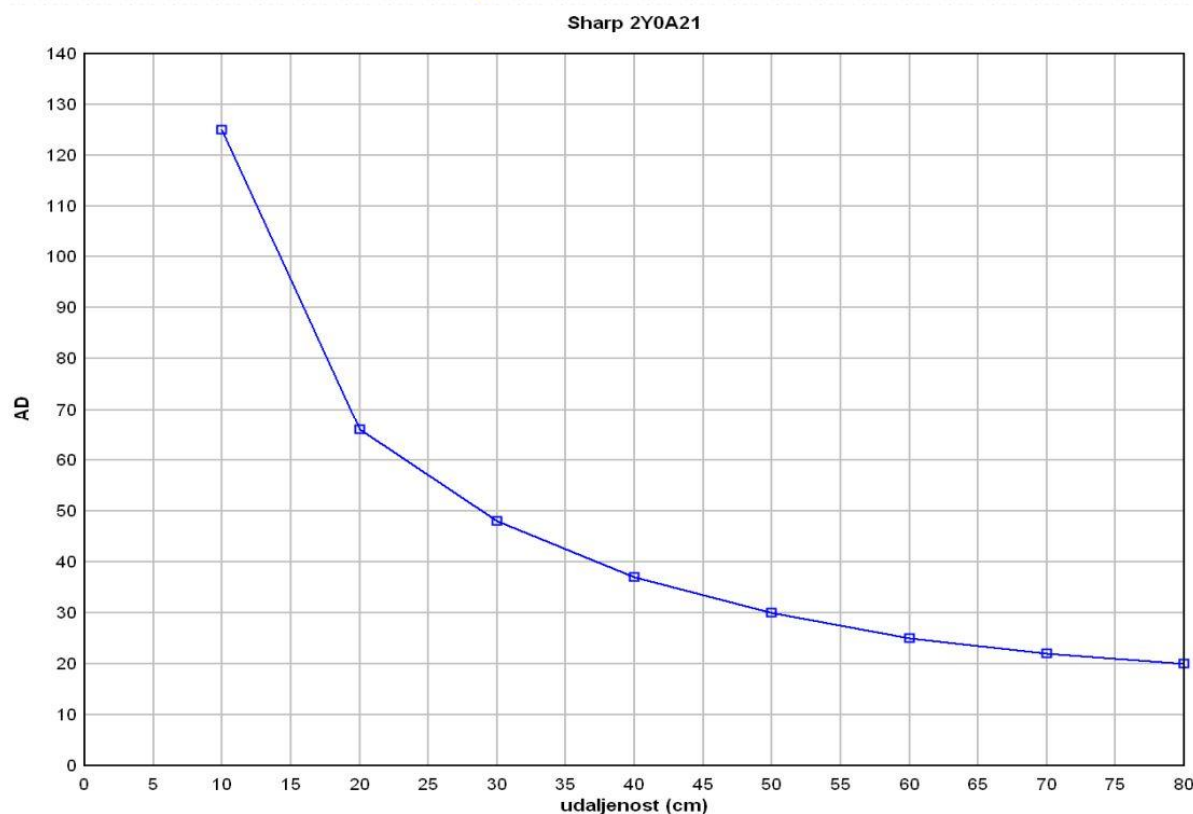
Edukacijski mobilni roboti su dobra osnova za proučavanje gibanja robota pojedinačno ili u formaciji. Roboti su opremljeni kamerom i sensorima tako da su dobivene informacije dovoljne za mnoge metode upravljanja. Nadalje, roboti podržavaju bilo koji programski jezik tako da je potencijalni broj korisnika jako velik. Još jedna prednost je da je vanjski vizijski sustav neovisan o upravljačkom sustavu robota. Robot je jednostavan za upravljanje i reagira brzo bez velikih kašnjenja i oscilacija. Kotači imaju dobar kontakt sa podlogom tako da nema proklizavanja, a komunikacija preko Bluetooth veze je pouzdana i radi dobro u stvarnom vremenu.



**Slika 11 Roboti eMiR**

#### 4.4. Infracrveni senzori

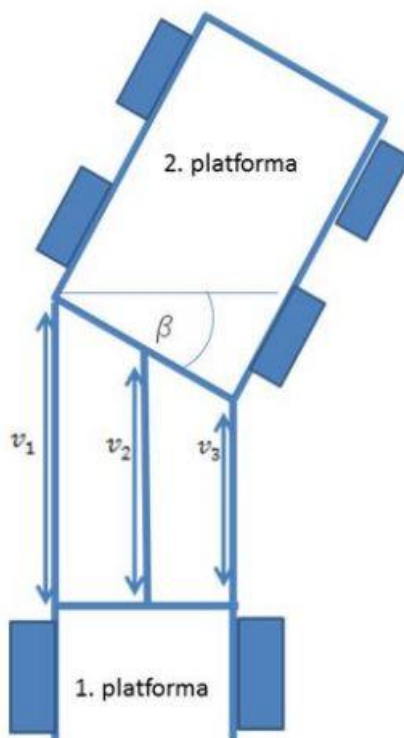
Robot se sastoji od 6 infracrvenih senzora udaljenosti tipa GP2Y0A21 s mjernim područjem od 10 do 80 cm. Poznato je da ti senzori imaju nelinearne karakteristike koje se preračunavaju kako dobili udaljenost u centimetrima. Mjerenje se vrši svakih 10 milisekundi. Budući da opisani senzori udaljenosti ne mjere manje od 10 cm, postoje i mjeraci udaljenosti s paralelnom kontrolom GP2Y0D810Z0F, koje također proizvodi Sharp.



Slika 12 Karakteristika senzora

## 5. ALGORITAM UPRAVLJANJA

Za upravljanje robotom koristit će se metoda voditelj-pratilac, u ovom slučaju, robot žute boje se upravlja pomoću joysticka i programa Robot Manager dok ga crveni prati.

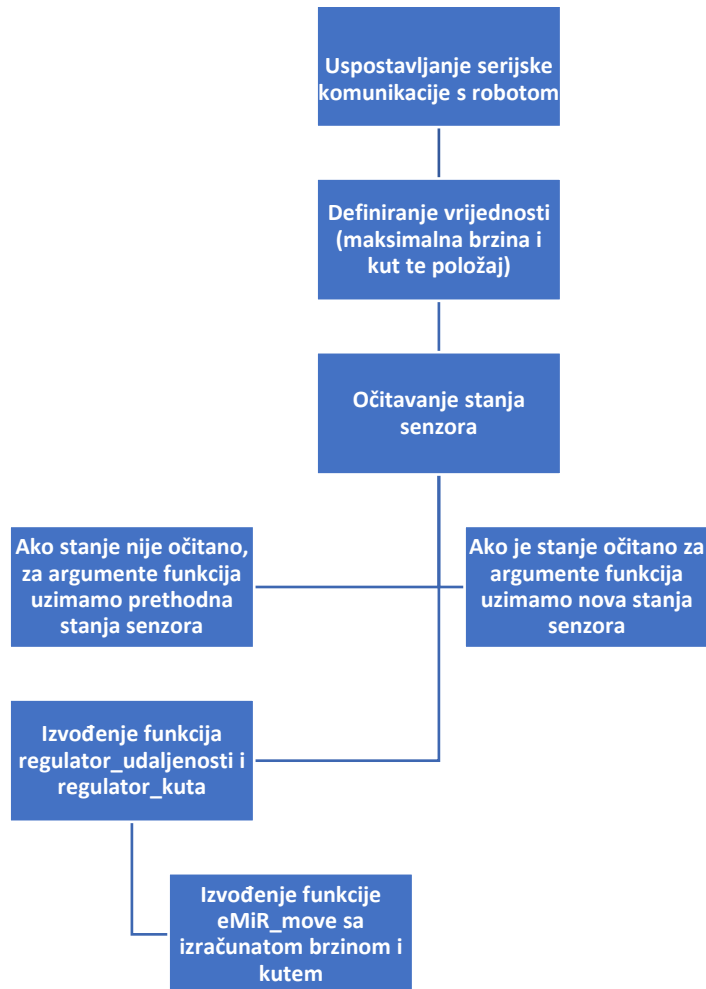


**Slika 13 Prikaz gibanja mobilnih platformi**

Mobilne platforme „eMiR“ se sastoje od tri kotača koja se upravljaju istosmjernim motorima, mikrokontrolera, bluetooth modula te infracrvenih senzora za mjerenje udaljenosti. Komunikacija s mikrokontrolerom se odvija pomoću skripte koja je napisana u programskom jeziku Python.

Funkcija eMiR\_init nam otvara komunikaciju s robotom na serijskom portu. Funkcija eMiR\_move( $v, w$ ) omogućuje da se robot giba translacijskom brzinom  $v$ , te rotacijskom  $w$ .

Također, u samoj skripti koja zapravo izvodi zadani algoritam se nalaze dvije funkcije. Prva funkcija regulator\_kuta ima za argumente kut  $\omega$ , te očitavanja sa 3 senzora na mobilnom robotu. Ta funkcija regulira kut u odnosu na platformu ispred. Analogno, stvorena je funkcija regulator\_udaljenosti koja za argumente ima brzinu, te očitavanja sa senzora. Obe funkcije rade na principu PI regulatora.

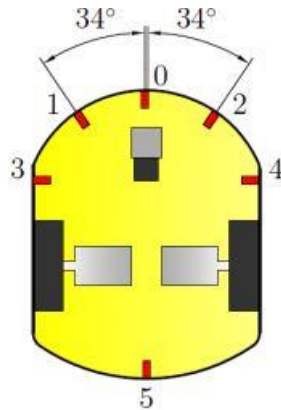


**Slika 14 Tijek upravljanja formacijom robota**

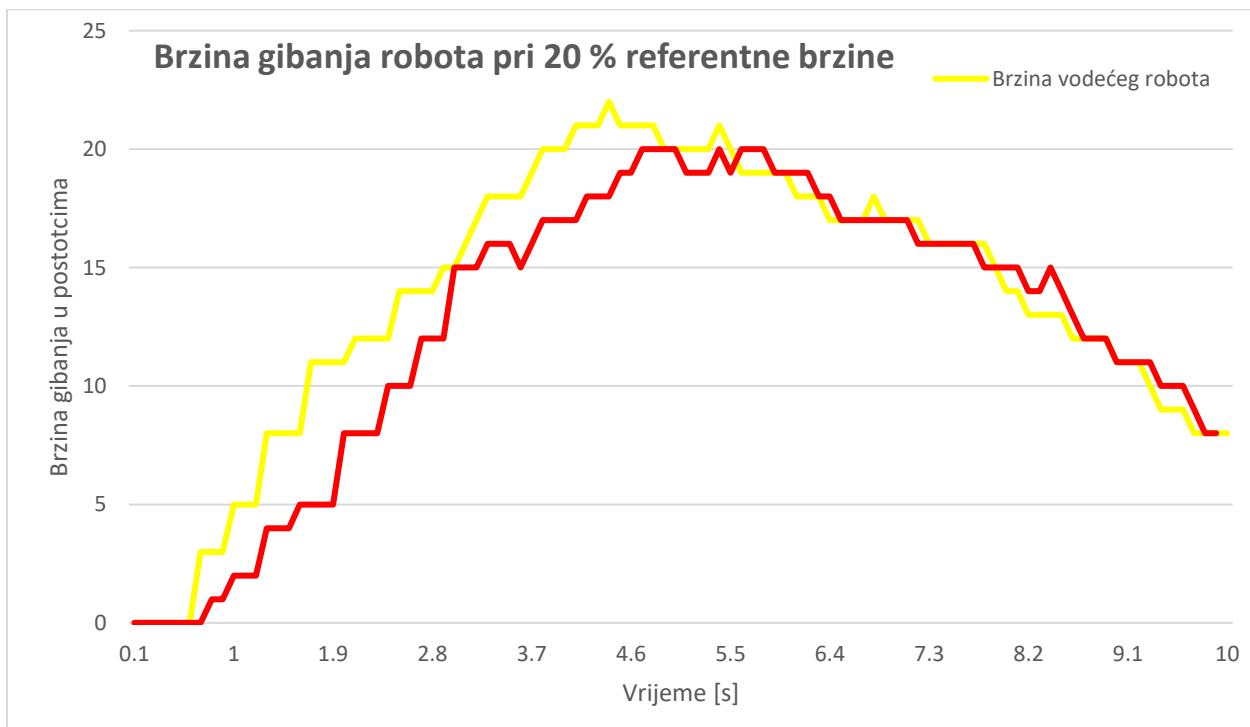
Na Slici 14 prikazan je algoritam kojim se upravljaju mobilni roboti. Na softveru „Robot Manager“ vodećem robotu se zadaje putanja kojom se giba. Nakon pokretanja skripte u Pythonu, slijedeći robot očitava stanje prednjih senzora (0,1,2). Ako je pročitao senzore, kao referentni senzor za upravljanje brzinom uzima se senzor 0 koji se nalazi na sredini robota. Sa dobivenim stanjem senzora 0,1,2 te brzinom robota ulazimo u funkciju regulator\_udaljenosti. U toj funkciji zadajemo parameter regulacije  $K_p$  i  $K_i$ . Udaljenost koju reguliramo je razlika između stanja senzora 0 te referentne udaljenosti koju mi sami zadajemo. Također, u samoj funkciji imamo ograničenja, jedno od njih je da brzina ne može biti veća od zadane maksimalne brzine. Drugi dio koji nam je potreban za regulaciju je smješten unutar funkcije regulator\_kuta. Za računanje kuta uzima se maksimalna vidljivost platforme kojoj prepreke manje smetaju. Ako je stanje senzora 0

veće od maksimalne zadane vidljivosti, za njegovo stanje uzimamo maksimalnu vidljivost. Također, u samom kodu su zadana ograničenja senzora 1 i 2 pomoću kojih izračunavamo mjereni kut. Pogrešku kuta koju reguliramo, analogno kao kod brzine, dobijemo kao razliku izmjereno kuta i kuta od  $90^\circ$ .

Promatrana su dva slučaja, u prvom slučaju pomoću softwera "Robot Manager" zadano je referentno gibanje vodećeg robota, u ovom slučaju žuti, koje u postocima iznosi 20 % od maksimalne brzine. U drugom slučaju, referentna brzina je iznosila 80 % od maksimalne brzine.

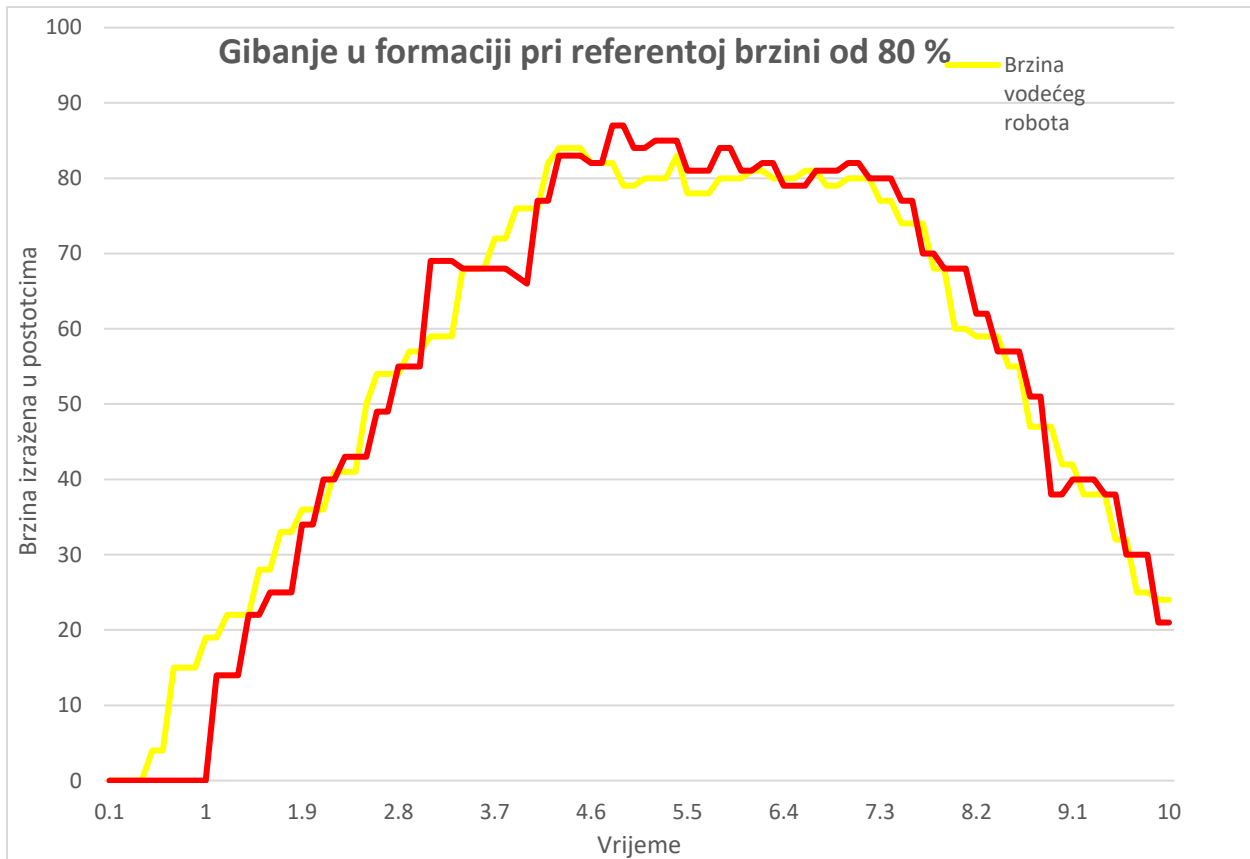


Slika 15 Prikaz mjernog sustava robota



Slika 16 Gibanje robota pri referentnoj brzini od 20 %

Gore navedeni slučaj daje nam zadovoljavajuće rezultate, ako uzmemo u obzir vrstu senzora koji se koriste (infracrveni senzori). Daljnjim podešavanjem parametara regulatora moguće je postići manje oscilacije.



**Slika 17** Gibanje robota pri referentnoj brzini od 80 %

U ovom slučaju dobivene brzine s očekivane zbog nedostatka komunikacije između robota. Sama izvedba bi se mogla poboljšati kada bi se koristio drugi način upravljanja robotima. Crveni robot, u ovom slučaju slijedeći, pokušava uskladiti brzinu s vodećim pa su na grafu vidljiva nagla ubrzavanja i usporavanja.

## 5.1. Oprema

### 5.1.1 Arduino Mega 2560

Arduino Mega 2560 je mikrokontrolerska pločica zasnovana na mikrokontroleru Atmega2560. Ima 54 ulazna/izlazna pina ( od kojih se 15 može koristiti kao PWM izlazi). Ima također i 16

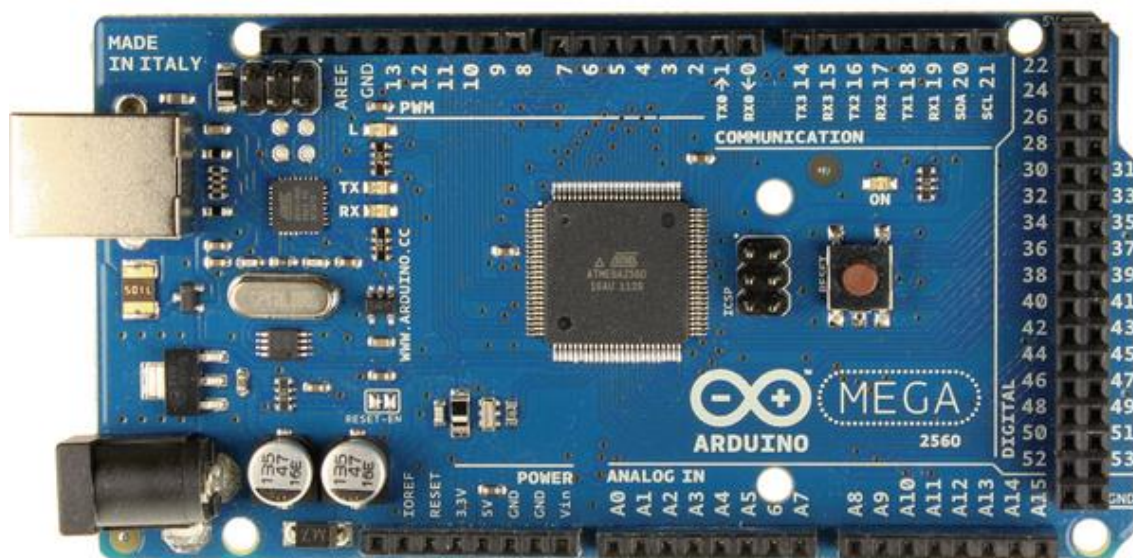
analognih izlaza, te dobru brzinu koju postiže sa 16 MHz kristalom. Najveće prednosti jesu da dolazi s više memorije te je veći veličinom. Nadalje, velika prednost samih pločica jest da je sama sintaksa i programski jezik jednostavan. Ova mikropločica se jako brzo pokreće te ima jako malu potrošnju energije. Ova Arduino pločica nudi i 4 hardverska porta za UART komunikaciju što nam je omogućilo jednostavnu bluetooth komunikaciju s robotima.

Specifikacije pločice su iduće:

<b>Mikrokontroler</b>	<b><a href="#">ATmega2560</a></b>
<b>Radni napon</b>	5V
<b>Ulazni napon (preporučeni)</b>	7-12V
<b>Ulazni Voltage (limit)</b>	6-20V
<b>Digitalni I/O pinovi</b>	54
<b>Analogni ulazni pinovi</b>	16
<b>Istosmjerna struja po I/O pinu</b>	20 mA
<b>Istosmjerna struja za 3.3V pin</b>	50 mA
<b>Flash Memorija</b>	256 KB
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Takt</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Duljina</b>	101.52 mm
<b>Širina</b>	53.3 mm
<b>Težina</b>	37 g

**Tablica 2. Specifikacije ATmega 2560 pločice**





Slika 18 Arduino ATmega 2560 pločica

### 5.1.2. Bluetooth modul WRL-00582

Robot je opremljen modulom WRL-00582 proizvođača SparkFun. Komunikacija je podešena na 57600 bps (bita u sekundi), ali može raditi pri brzinama od 9600 bps do 115200 bps. Domet modula iznosi 106 m.

Tehničke specifikacije su sljedeće:

- Mala potrošnja električne energije: 25mA u prosjeku
- Enkriptirana konekcija
- Frekvencija: 2.4~2.524 GHz
- Radni napon: 3.3V-6V
- Serijska komunikacija: 2400-115200bps
- Radna temperatura: -40 ~ +70C
- Ugrađena antenna

## 6. ZAKLJUČAK

U radu je pokazano kako je uz dobro implemeniran algoritam moguće obavljanje gibanja u formaciji. Gibanje u formaciji voditelj-pratilac je osnova za mnoga kompleksnija gibanja korištena u industriji, poljoprivredi i transportu. Zbog ograničenosti broja platformi algoritam je izveden za dva mobilna robota. Problemi u formaciji nastaju pri nagloj promjeni smjera gibanja vodećeg robota gdje robot koji ga slijedi zbog ograničenosti svoje konstrukcije i razmještaja senzora gubi vezu s vodećim robotom te dolazi do raspada formacije. Ovaj problem mogao bi se riješiti uz korištenje više senzora ili korištenjem neke druge opreme kao što su kamere ili LIDAR.

## LITERATURA

- [1]- Prof.dr.sc. Mladen Crneković- educational Mobile Intelligent Researcher
- [2]-[https://www.researchgate.net/publication/332370346\\_A\\_Survey\\_of\\_Multi-mobile\\_Robot\\_Formation\\_Control](https://www.researchgate.net/publication/332370346_A_Survey_of_Multi-mobile_Robot_Formation_Control)
- [3]-Madhevan, Sreekuman, Tracking Algorithm Using Leader Follower Approach for Multi Robots. International Conference on Design And Manufacturing, IConDM 2013.
- [4]- <https://repozitorij.fsb.unizg.hr/islandora/object/fsb:2987>
- [5]- <https://repozitorij.fsb.unizg.hr/islandora/object/fsb%3A5679>
- [6]- MELLODGE P., Feedback Control for a Path Following Robotic Car. Master degree thesis, Faculty of the Virginia Polytechnic Institute and State University ,2002.

## PRILOZI

- I. CD disk
- II. Python skripta i kod za izvođenje

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

```
def eMiR_citaj():
    global RS232

    i=0
    znak=''
    string=''

    if RS232.inWaiting()>0:
        znak=RS232.readline(1)
        if znak=='*':
            while znak<>'/' and i<26:
                string=string+znak
                znak=RS232.readline(1)
                i=i+1
            if i>26:
                string=''
            else:
                string=string[0:24]
                RS232.flushInput()
            return string
        RS232.flushInput()
    return ""
```

```
def eMiR_stop():
    global RS232
```

```
RS232.write('#010000FF/')
RS232.flushOutput()
RS232.close()

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

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

    if len(Senzori)==24:
        #sen0
        sen0=Senzori[1:3]
        sen0=int(sen0,16)
        #sen1
        sen1=Senzori[3:5]
        sen1=int(sen1,16)
        #sen2
        sen2=Senzori[5:7]
        sen2=int(sen2,16)
        #sen3
        sen3=Senzori[7:9]
        sen3=int(sen3,16)
        #sen4
        sen4=Senzori[9:11]
        sen4=int(sen4,16)
        #sen5
        sen5=Senzori[11:13]
        sen5=int(sen5,16)
        #sen6
        sen6=Senzori[13:15]
        sen6=int(sen6,16)
        #bat
        bat=Senzori[15:17]
        bat=int(bat,16)
        #vtrans
        vtrans=Senzori[17:19]
        vtrans=int(vtrans,16)
        if vtrans > 127 :
            vtrans = vtrans - 256
        #vrot
        vrot=Senzori[19:21]
        vrot=int(vrot,16)
        #if vrot > 127 :
```

```
        # vrot = vrot - 256
    izlaz=[]
    for i in range(0,len(lista)):
        if lista[i] == 0:
            izlaz.append(sen0)
        if lista[i] == 1:
            izlaz.append(sen1)
        if lista[i] == 2:
            izlaz.append(sen2)
        if lista[i] == 3:
            izlaz.append(sen3)
        if lista[i] == 4:
            izlaz.append(sen4)
        if lista[i] == 5:
            izlaz.append(sen5)
        if lista[i] == 6:
            izlaz.append(sen6)
        if lista[i] == 7:
            izlaz.append(vtrans)
        if lista[i] == 8:
            izlaz.append(vrot)
    return izlaz
else:
    return []

def eMiR_move(b,h):

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

    W=''
    if h<0 :
        h = h+256
    W = hex(h)
    d = W.split('x')
    if h<16 :
        W1='0'+d[1]
```

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

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

Programski kod za izvođenje praćenja

```
def regulator_kuta(preth_omega, sen0, sen1, sen2):
    beta=0.59341195
    global preth_kut_err
    Kp=20
    Ki=2

    vidljivost=35

    epsilon=0.001

    #odredivanje kuta prethodne platforme
    if sen0==0:
        sen0=10
    if sen1==0:
        sen1=10
    if sen2==0:
        sen2=10

    # za racunanje kuta uzima maksimalna vidljivost platforme
    kojoj prepreke manje smetaju
    if sen0>vidljivost:
        sen0=vidljivost
```

```

    if sen1<vidljivost and sen1<sen2:
        mjer_kut=pi-asin((sen1*sin(beta))/sqrt(sen1**2+sen0**2-
2*sen1*sen0*cos(beta)))
    elif sen2<vidljivost and sen2<sen1:
        mjer_kut=asin((sen2*sin(beta))/sqrt(sen2**2+sen0**2-
2*sen2*sen0*cos(beta)))
    else:
        mjer_kut=pi/2

    kut_err=mjer_kut-pi/2
    if abs(kut_err)<epsilon:
        kut_err=0
        preth_omega=0

    #inkrementalni PI regulator
    omega=preth_omega+kut_err*(Kp+Ki*0.001)-preth_kut_err*Kp

    if omega>maks_kutna_brzina:
        omega=maks_kutna_brzina
    if omega<-maks_kutna_brzina:
        omega=-maks_kutna_brzina

    preth_kut_err=kut_err
    preth_omega=omega

    return int(omega)

def regulator_udaljenosti(preth_brzina,sen0,sen1,sen2):
    global preth_er_udaljenost
    Kp=1
    Ki=0.5

    er_udaljenost=sen0-ref_udaljenost

    #inkrementalni PI regulator
    # brzina=preth_brzina+er_udaljenost*(Kp+Ki*0.001)-
preth_er_udaljenost*Kp
    # P regulator
    brzina=Kp*er_udaljenost

    if brzina>maks_brzina:
        brzina=maks_brzina

```



```
    if brzina<0:
        brzina=0

    if sen1==0:
        brzina=0
    if sen2==0:
        brzina=0

    print preth_er_udaljenost
    print er_udaljenost

    preth_er_udaljenost=er_udaljenost
    preth_brzina=brzina

    return int(brzina)

from eMiRcon import*
import time
from math import*
from datetime import datetime
#from pylab import *

eMiR_init(9)

global preth_vrijednosti
global ref_udaljenost
global maks_brzina
global maks_kutna_brzina
global preth_er_udaljenost
global preth_kut_err

preth_er_udaljenost=0
preth_kut_err=0
preth_vrijednosti=[80,80,80]
preth_brzina=0
preth_omega=0

ref_udaljenost=10

maks_brzina=40
maks_kutna_brzina=30
```

```
brzina=0
sen0=0
sen1=0
sen2=0

i=0
duljina=0
while duljina==0:
    sens=eMiR_sensor_status([0,1,2])
    sen0_preth=sen0
    sen1_preth=sen1
    sen2_preth=sen2

    duljina=len(sens)

time.sleep(0.5)

wispis=[]
t=[]
while i<10000:
    sens=eMiR_sensor_status([0,1,2,7,8])
    sen0_preth=sen0
    sen1_preth=sen1
    sen2_preth=sen2
    if len(sens)>0:
#   procitao senzore
        sen0=sens[0]
        sen1=sens[1]
        sen2=sens[2]
        tren_brzina=sens[3]
        tren_omega=sens[4]
        wispis.append(tren_omega)
        t.append(i)
    else:
#   nije procitao senzore
        sen0=sen0_preth
        sen1=sen1_preth
        sen2=sen2_preth

brzina=regulator_udaljenosti(preth_brzina, sen0, sen1, sen2)
omega=regulator_kuta(preth_omega, sen0, sen1, sen2)
eMiR_move(brzina, omega)
```

```
preth_brzina=brzina
preth_omega=omega

#print sen0,' ',sen1,' ',sen2,' ',brzina,'
',tren_brzina,' ',omega,' ',tren_omega
print tren_omega
i=i+1

eMiR_stop()
#plot(t,wispis)
```