

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

*Tomislav Petković, Luka Petrović*

**Decentralizirano upravljanje  
multiagentskim robotskim  
sustavom**

Zagreb, 2016.

*Ovaj rad izrađen je u Laboratoriju za robotiku i inteligentne sustave upravljanja pod vodstvom prof. dr. sc. Stjepana Bogdana i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2015./2016.*

# SADRŽAJ RADA

<b>1. Uvod</b>	<b>1</b>
<b>2. Algoritmi povjerenja u multi-agentskim sustavima</b>	<b>3</b>
2.1. Pojam povjerenja . . . . .	3
2.2. Adaptivni algoritam računanja povjerenja . . . . .	4
2.3. Simulacijski rezultati . . . . .	5
2.3.1. Analiza algoritma kada svaki robot komunicira sa svakim . . .	8
2.3.2. Analiza algoritma uz robote u V formaciji . . . . .	13
<b>3. Upravljanje formacijom multi-agentskog sustava</b>	<b>18</b>
3.1. Funkcija potencijala . . . . .	18
3.2. Upravljanje položajem i orijentacijom robota . . . . .	23
3.2.1. Tehnički aspekti implementacije . . . . .	23
3.2.2. Generiranje reference robota . . . . .	25
3.2.3. Regulacija brzina robota . . . . .	26
<b>4. Simulacija decentraliziranog upravljanja formacijom</b>	<b>28</b>
4.1. Promjena formacije u Gazebo simulatoru . . . . .	28
4.2. Patroliranje u Gazebo simulatoru . . . . .	34
<b>5. Decentralizirano upravljanje formacijom u laboratoriju</b>	<b>45</b>
5.1. Promjena formacije u laboratoriju . . . . .	45
5.2. Patroliranje u laboratoriju . . . . .	51
<b>6. Zaključak</b>	<b>57</b>

<b>7. Literatura</b>	<b>58</b>
<b>8. Sažetak</b>	<b>60</b>
<b>9. Summary</b>	<b>61</b>

# 1. Uvod

U današnje je vrijeme sve više ljudskih poslova zamjenjivo robotima. U automobilskim tvornicama oni zavaruju i lakiraju dijelove, sudjeluju u operacijama poput poznatog Da Vinci kirurškog sustava [1], a u razvoju su i automobili bez vozača [2]. Jedan od tih poslova je zasigurno i patroliranje, odnosno periodičko ispitivanje neke površine s određenim ciljem. Umjesto slanja čovjeka ili tima ljudi kako bi provjerili temperaturu u određenim dijelovima elektrana, vlažnost u tvornicama duhana ili količinu kemikalija u zraku centra za odlaganje toksičnog otpada, moguće je na takve zadatke poslati patrolirajuće robote opremljene odgovarajućim sensorima, odnosno robotski multi-agentski sustav. Multi-agentski sustavi se sastoje od više agenata koji imaju zajednički zadatak. Cilj ovog rada je implementacija gibanja u formaciji na robotima u Gazebo simulatoru i laboratoriju. Koristit će se Pioneer P3-DX roboti čiji je prikaz u simulacijskom okruženju Gazebo dan na Slici 1.1.



**Slika 1.1:** Pioneer robot u Gazebo simulatoru

Način upravljanja robotskog multi-agentskog sustava može biti centraliziran i decentraliziran. Zbog specifičnosti nekih zadataka nije uvijek moguća komunikacija cen-

tralnog računala sa svim agentima. Zato je nužno da agenti tijekom rada donose odluke temeljene na ciljevima zadatka, kvaliteti izvedbe zadane od strane operatera ili prirodni zadatka koja će voditi ka ostvarenju željenog cilja. Nadalje, tijekom vremena može doći do izmjene zadatka, cilja, okoline pa i samog sustava kao i do promjene sposobnosti pojedinog agenta za obavljanjem zadatka. Postavlja se pitanje kako se suočiti s tim problemima pri dizajniranju sustava. Ako promotrimo biološke grupe poput jata, rojeva, krda pa čak i ljudskih društvenih zajednica [3], možemo primijetiti da imaju ugrađene mehanizme za prepoznavanje članova grupe kao prijatelja, odnosno neprijatelja, na osnovu lokalnih interakcija. Motivirano time, u multi-agentske sustave uvode se lokalne interakcije. Upravo na njima temelji se sveukupna informacija o mreži koja je dostupna svakom agentu. Pouzdanost interakcija ovisi o raznim čimbenicima, počevši od kvalitete komunikacijskih kanala i dosljednosti opažanja pa sve do sposobnosti i namjere pojedinca uključenog u interakciju. Kako bi se spomenute interakcije formalno interpretirale, potrebno je uvesti pojam povjerenja između agenata. Na temelju povjerenja agenti mogu odlučivati tijekom rada i mijenjati svoje ponašanje kako bi zadatak bio što uspješnije obavljen.

Algoritmima povjerenja u multi-agentskim sustavima se bavi drugo poglavlje u sklopu kojeg će se objasniti njihova matematička podloga i primjena s naglaskom na primjenu u decentraliziranom upravljanju formacijom. U trećem poglavlju će se reći nešto više o implementiranom načinu decentraliziranog upravljanja formacijom multi-agentskog sustava pomoću funkcije potencijala. Naglasak će biti stavljen na osiguravanje konvergencije (svaki robot mora u konačnom vremenu doći na svoj cilj) i odsustvo sudara. U četvrtom i petom poglavlju se teoretski aspekti algoritama konsenzusa i upravljanja implementiraju te se njihova funkcionalnost ispituje u Gazebo simulatoru i u laboratorijskim uvjetima.

## 2. Algoritmi povjerenja u multi-agentskim sustavima

### 2.1. Pojam povjerenja

Pojam povjerenja uveden je u multi-agentske sustave kako bi se interakcije agenata mogle formalno protumačiti. Kako bi dva agenta obavila zajednički zadatak, rad svakoga od njih treba se oslanjati na određenoj razini sljedećih uvjerenja [4]:

- drugi agent je sposoban izvršiti traženi zadatak
- drugi agent namjerava izvršiti traženi zadatak.

Zapravo, povjerenje je relacija triju članova, agent  $i$  vjeruje agentu  $j$  da će izvršiti zadatak  $a$ . Osnovni algoritam za ažuriranje vektora povjerenja agenata je (2.1) [5]:

$$\dot{\zeta}_i = \sum_{j \in N_i} \zeta_{ij} (\zeta_i - \zeta_j) \quad (2.1)$$

Gdje je  $\zeta_i$  vektor povjerenja  $i$ -tog agenta,  $\zeta_{ij}$  vrijednost povjerenja agenta  $i$  prema agentu  $j$ , a  $N_i$  skup susjeda agenta  $i$  kojeg u kontekstu patroliranja možemo protumačiti kao skup robota čija je udaljenost od robota  $i$  unutar nekog komunikacijskog radijusa  $r$ . U ovom radu elementi vektora  $\zeta_i$  poprimaju vrijednosti iz intervala  $[0, 1]$ , gdje 0 znači nepovjerenje, a 1 maksimalno povjerenje. Glavni nedostatak ovog algoritma je taj da ne uzima u obzir zapažanja agenata. Ako bolje promotrimo (2.1), možemo uočiti kako stacionarna stanja ovise isključivo o početnim uvjetima povjerenja među agentima. Međutim, sposobnost pojedinog agenta za izvršenje zadatka tokom njegovog izvođenja se mijenja u ovisnosti o raznim faktorima poput kvara na mjernom instrumentu robota, greške u komunikaciji s robotom, promjene cilja zadatka za koji robot nije dovoljno

opremljen, itd. Zbog toga je nužno adaptirati algoritam računanja povjerenja i uključiti opažanja agenata  $i$  o sposobnosti izvršenja zadatka njegovih susjeda.

## 2.2. Adaptivni algoritam računanja povjerenja

Ovaj algoritam određivanja povjerenja agenata uzima u obzir opažanja agenata i osigurava konvergenciju vektora povjerenja. Zasniva se na razmijenjenim informacijama između agenata, kao i na agentovim opažanjima [6]. Zato se algoritam (2.1) za ažuriranje vektora povjerenja agenta proširuje na sljedeći način:

$$\begin{cases} \sum_{k \in N_i} (\zeta_{kj} - \zeta_{ij}) + (\tau_{ij}(\sigma_{ij}, \delta_{ij}) - \zeta_{ij}), & j \in N_i \\ \sum_{k \in N_i} (\zeta_{kj} - \zeta_{ij}), & j \notin N_i \end{cases} \quad (2.2)$$

gdje je  $\tau_{ij}$  funkcija opažanja agenta  $i$  o dostojnosti povjerenja njegovog susjeda. Ta funkcija ovisi o parametru  $\sigma_{ij}$ , koji predstavlja pouzdanje agenta  $i$  u njegovog susjeda  $j$  i varijablu  $\delta_{ij}$  koja predstavlja sposobnost izvršavanja zadatka agenta  $j$  kako ju vidi agent  $i$ . Njezino tumačenje ovisi o primjeni i kontekstu zadatka. Kod upravljanja formacijom grupe, koje će biti kasnije obrađeno,  $\delta_{ij}$  je razlika mjerene pozicije i pozicije dobivene komunikacijom sa susjedima. Performansa agenta  $j$  viđena od različitih agenata može biti različita, odnosno općenito je  $\delta_{ij} \neq \delta_{kj}$ . Definicija funkcije  $\tau_{ij}$  dana je s [7]:

$$\tau_{ij} = e^{\frac{-\delta_{ij}^2}{\sigma_{ij}^2}} \quad (2.3)$$

Kako bi se osigurala konvergencija algoritma povjerenja (2.2), potrebno je uvesti mehanizam prilagođavanja u već iznesen algoritam. Pretpostavka iza ovog mehanizma je da su agenti voljni surađivati, odnosno umjesto da susjedi agenta  $j$  samo nameću svoje opažene iznose povjerenja, oni prilagođavaju svoje funkcije opažanja povjerenja  $\tau_{ij}$ . Rezultat prilagođavanja funkcija  $\tau_{ij}$  treba biti ista vrijednost opaženog povjerenja prema agentu  $j$  od strane svih njegovih susjeda jer će u tom slučaju doći do konvergencije povjerenja. Kao parametar prilagođavanja koristi se parametar  $\sigma_{ij}$ . Za promjenu parametra  $\sigma_{ij}$ , odnosno za prilagođavanje funkcije  $\tau_{ij}$  koristi se (2.4) gdje je  $K$  poja-



čanje prilagođavanja.

$$\dot{\sigma}_{ij} = -K \cdot (\tau_{ij}(\sigma_{ij}, \delta_{ij}) - \zeta_{ij}), K > 0 \quad (2.4)$$

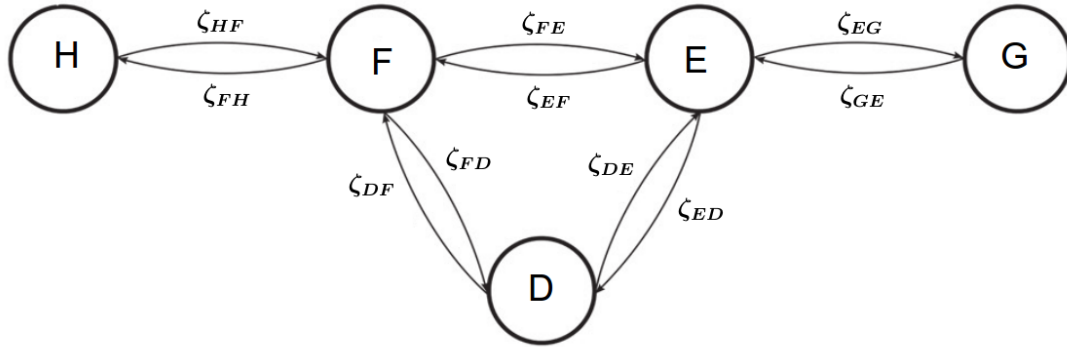
Odabir pojačanja K je proizvoljan, no uvelike utječe na rezultate konvergencije. Odabirom velikih vrijednosti pojačanja K osiguravamo bržu konvergenciju, ali povećavamo osjetljivost algoritma na početne uvjete. Na temelju detaljnih simulacijskih rezultata iz [8], odabrana je vrijednost  $K=0.02$ .

### 2.3. Simulacijski rezultati

Implementacija algoritma (2.2) provodi se na multirobotskom sustavu sastavljenom od pet robota (Gazebo okruženje), odnosno tri robota (laboratorij). Roboti su imenovani su po NATO fonetskoj abecedi [9] kako slijedi:

- Delta
- Echo
- Foxtrot
- Golf
- Hotel

Svaki agent prima vektore povjerenja svojih susjeda, računa svoj vektor povjerenja i šalje ga svojim susjedima. Grafički prikaz komunikacijske mreže nalazi se na Slici 2.1. Agentu Delta susjedi su Foxtrot i Echo koji komuniciraju međusobno dok agenti Hotel i Golf imaju samo jednog susjeda.

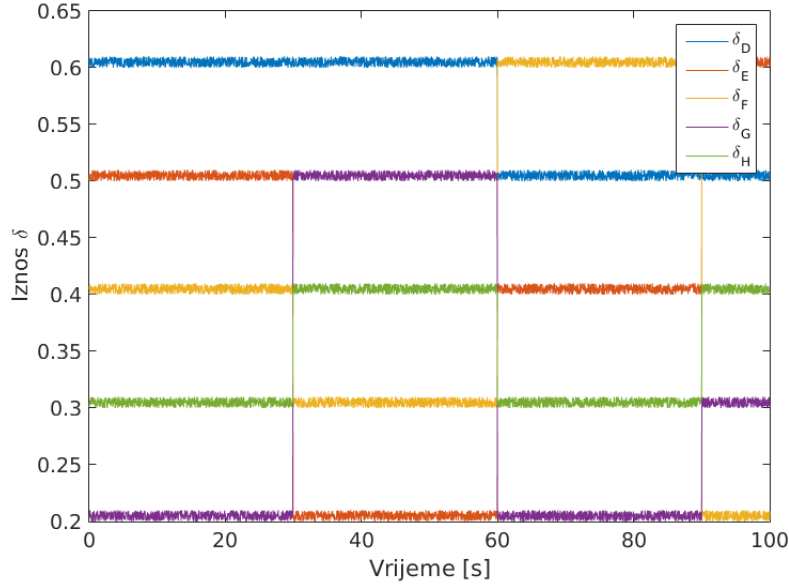


**Slika 2.1:** Prikaz komunikacijske mreže za računanje vektora povjerenja s početnim slovima agenata

Vektor mjerenja  $\delta$  koristi simulirane vrijednosti s obzirom da roboti nemaju ugrađene senzore kojima bi mogli mjeriti udaljenost. Početna vrijednost vektora odabrana je kako slijedi:

$$\delta = \begin{bmatrix} \delta_{Delta} & \delta_{Echo} & \delta_{Foxtrot} & \delta_{Golf} & \delta_{Hotel} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.5 & 0.4 & 0.2 & 0.3 \end{bmatrix} \quad (2.5)$$

gdje svaki stupac odgovara odstupanju od mjerenja odgovarajućeg robota. Kroz cijeli rad su vrijednosti u vektorima poredane tako da podatak u  $i$ -tom stupcu odgovara robotu na  $i$ -tom mjestu u njihovom abecednom poretku. Vektor mjerenja se nasumično permutira ukoliko je potrebno zamijeniti poredak robota unutar formacije, o čemu će biti više govora u narednim poglavljima. Integracija se provodi u diskretnoj domeni, uz primjenu ZOH [10] metode diskretizacije s frekvencijom uzorkovanja 30 Hz. Kako bi se što bolje simulirala stvarna mjerenja, vektoru mjerenja  $\delta$  dodan je vektor redak dimenzije 5 gdje je svaki element slučajna varijabla dana uniformnom razdiobom na intervalu  $[0, 0.01]$ , kojim se simulira mjerni šum. Prikaz mijenjanja vektora  $\delta$  u vremenu uz permutiranje s periodom 30 sekundi može se vidjeti na Slici 2.2. Prvi period najbolje mjeri agent Golf, drugi Echo, a treći opet Golf.



**Slika 2.2:** Permutiranje stupaca vektora  $\delta$  s periodom 30 sekundi

Prilikom patroliranja agente je potrebno držati u raznim formacijama, a u ovom radu položaj agenta unutar formacije određen je vektorom povjerenja  $\zeta$ . Položaj agenta unutar formacije se mijenja tek kad se promijene vektori povjerenja, točnije kad se postigne konvergencija konsenzusa. Kako bi se to moglo postići, agenti u svakom trenutku moraju znati je li konsenzus postignut. Svaki agent postignutost konsenzusa provjerava sljedećom funkcijom:

$$J_i = \sum_{j=1}^n \zeta_{ij}^2 \quad (2.6)$$

Agent  $i$  će pretpostaviti da je došlo do konvergencije ako je  $J_i < \varepsilon$ , gdje je  $\varepsilon$  mali broj, a u simulacijama se koristi  $\varepsilon = 10^{-3}$ . Agenti u pravilu neće istovremeno pretpostaviti da je došlo do konvergencije. Zbog toga se smatra da je konsenzus postignut ako je agent pretpostavio konvergenciju, ako su njegovi susjedi pretpostavili konvergenciju i ako su susjedi njegovih susjeda postigli konvergenciju. To je izvedeno na način da se vektor povjerenja agenta  $i$ ,  $\zeta_i$ , proširuje s tri logičke varijable. Početni iznosi povjerenja i pouzdanja dani su u (2.7). Svaki agent inicijalno sebi vjeruje najviše, dok ostalim agentima vjeruje podjednako, ali manje nego sebi. Logičke varijable koje označuju je

li konsenzus postignut su na nuli, što znači da konsenzus nije postignut.

$$\zeta(0) = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0 & 0 & 0 \\ 0.1 & 1 & 0.1 & 0.1 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 1 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.7)$$

Početni iznosi pouzdanja određeni su proizvoljno i dani u (2.8), gdje element  $\sigma_{ij}$  označava pouzdanje agenta  $i$  u agenta  $j$ , odnosno svaki agent se jednako pouzda u svakog agenta, uključujući sebe.

$$\sigma(0) = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \quad (2.8)$$

Ranije je navedeno kako dva agenta smatramo susjedima ako im je fizička udaljenost manja od nekog komunikacijskog radijusa  $r$ . Uvodimo kvadratnu matricu susjedstva  $\mathbf{A} = \{a_{ij}\}$  za koju vrijedi [11]:

$$a_{ij} = \begin{cases} 1, & 0 < d_{ij} < r \\ 0, & d = 0 \vee d \geq r \end{cases} \quad (2.9)$$

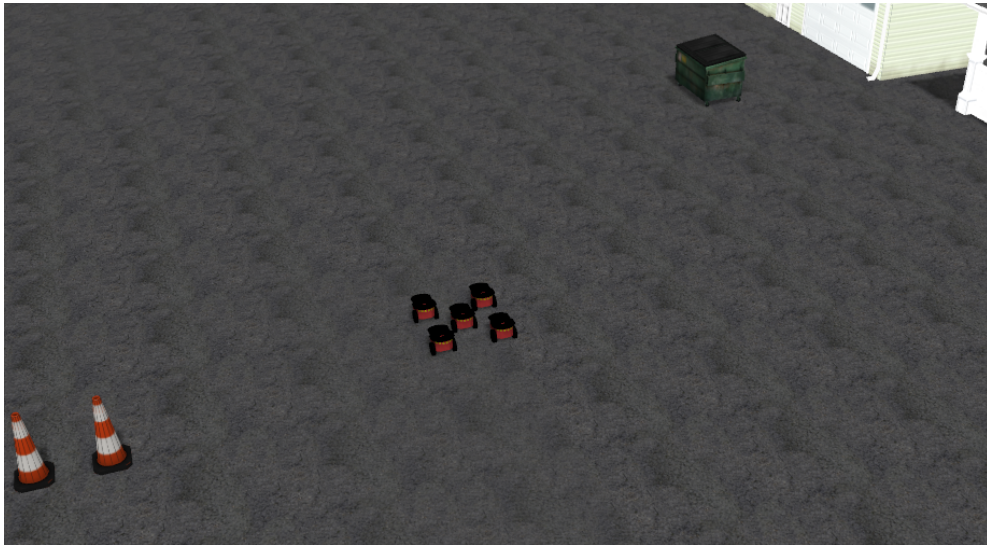
gdje je  $d_{ij}$  fizička udaljenost agenata  $i$  i  $j$ . Valja primijetiti kako su elementi glavne dijagonale matrice  $\mathbf{A}$  jednaki nuli. Ta je matrica funkcija položaja agenata, a samim time i promjenjiva u vremenu. Za potrebe analize rada algoritma (2.2) ćemo razmatrati dva slučaja:

- svaki agent je u komunikacijskom radijusu svakog agenta
- agenti su u V formaciji i imaju jednog ili dva susjeda

### 2.3.1. Analiza algoritma kada svaki robot komunicira sa svakim

Slučaj kada svaki robot komunicira sa svakim možemo predočiti Slikom 2.3. Iako je ta formacija nepraktična zbog male površine koju pokriva i otežanog osiguranja od

sudara, zanimljiva je sa stajališta analize algoritma konsenzusa jer može predstavljati početan položaj.

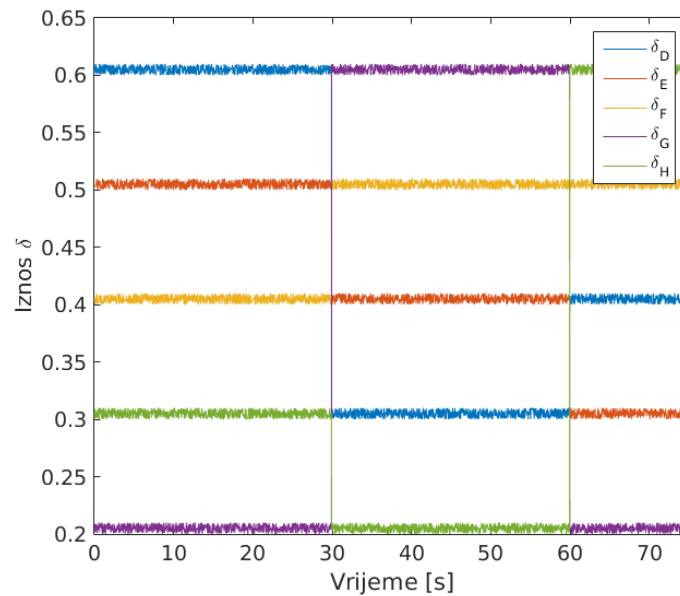


**Slika 2.3:** Pioneer roboti na okupu u simulacijskom okruženju Gazebo

Matrica susjedstva  $\mathbf{A}$  za ovaj slučaj iznosi:

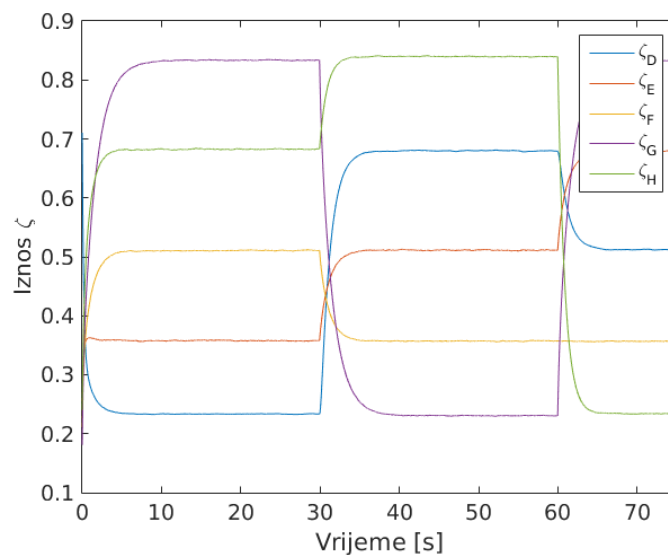
$$\mathbf{A}(t) = \mathbf{A}(0) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (2.10)$$

Analizirat će se signali koje šalje agent Delta jer su svi agenti jednaki, imaju jednak broj susjeda i početne uvjete. Na Slici 2.4 mogu se vidjeti kretanja vrijednosti stupaca vektora  $\delta$ .



**Slika 2.4:** Vrijednosti mjerenja. Prvu periodu najbolje mjeri agent Golf, a drugu Hotel.

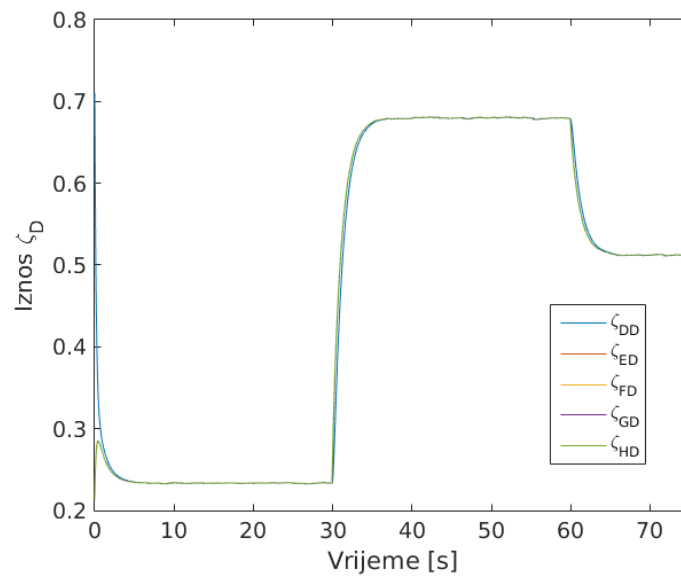
Već nakon pet sekundi vrijednosti povjerenja uđu u stacionarno stanje na Slici 2.5. Agenti su poredani u skladu s mjerenjima na Slici 2.4. Može se vidjeti kako je utjecaj šuma zanemariv.



**Slika 2.5:** Povjerenje agenta Delte prema ostalim agentima

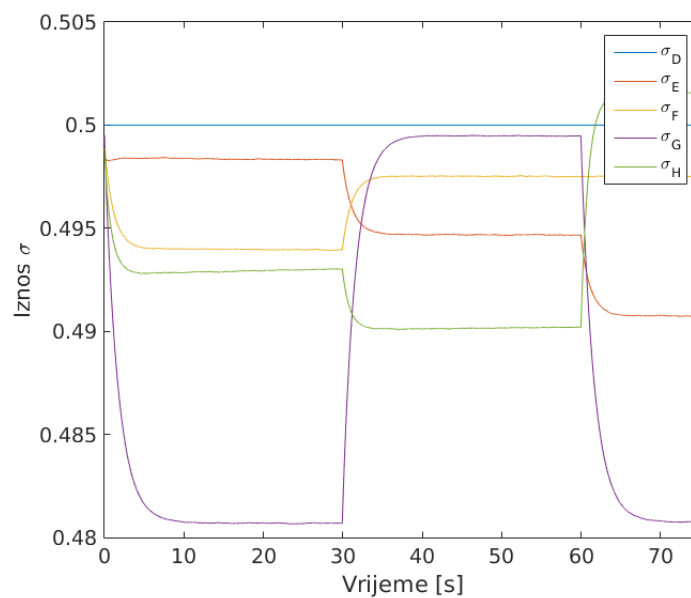
Na Slici 2.6 vidi se kako krivulje povjerenja ostalih agenata prema agentu Delta imaju prilično sličan oblik zbog jednakih početnih uvjeta. Svi su agenti blizu, pa kaš-

njenje ne postoji.



**Slika 2.6:** Povjerenje agenata prema agentu Delta

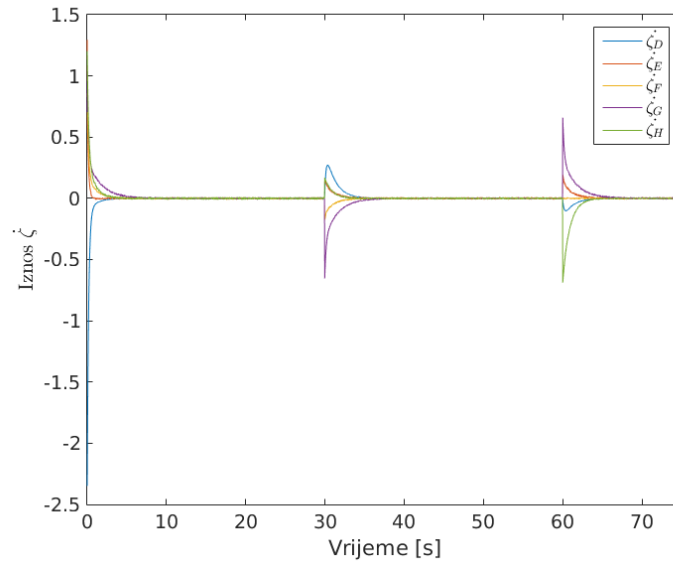
Pouzdanje agenta Delta prema ostalim agentima se mijenja u ovisnosti o mjerenjima, dok se samopouzdanje ne mijenja, što se vidi na Slici 2.7.



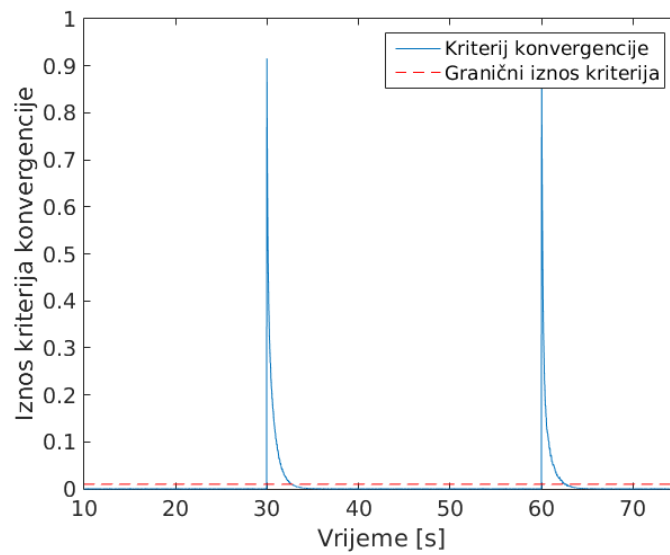
**Slika 2.7:** Pouzdanje agenta Delta prema drugim agentima

Na Slikama 2.8 i 2.9 prikazani su iznosi promjene povjerenja i iznos kriterija konvergencije. Agent Delta pretpostavlja da je došlo do konvergencije kad je iznos krite-

rija konvergencije manji od graničnog iznosa kriterija koji je označen crvenom crtka-  
nom linijom.



**Slika 2.8:** Promjena povjerenja agenta Delte



**Slika 2.9:** Iznos kriterija konvergencije



### 2.3.2. Analiza algoritma uz robote u V formaciji

Slučaj kada se agenti nalaze u V formaciji možemo predočiti Slikom 2.10. Matrica susjedstva  $\mathbf{A}$  za ovaj slučaj iznosi:

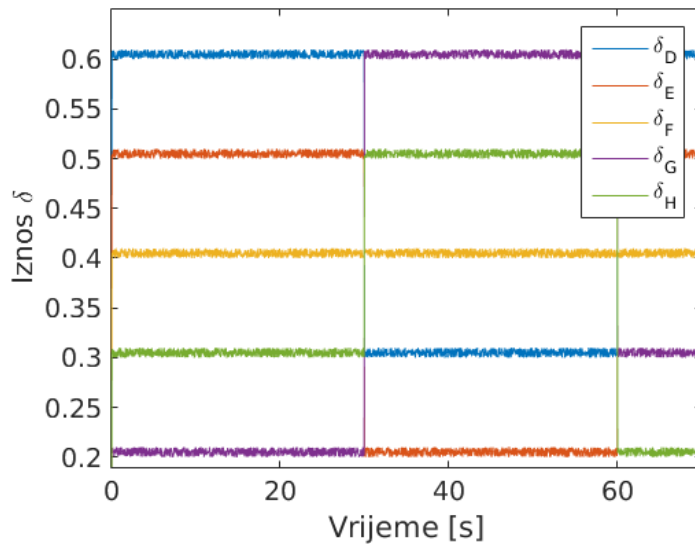
$$\mathbf{A}(t) = \mathbf{A}(0) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.11)$$

Iz (2.11) se može iščitati kako je na čelu formacije Delta, slijede Echo i Foxtrot, te su na začelju Golf i Hotel. Prva tri robota imaju dva susjeda, dok posljednja dva imaju vezu samo s robotom ispred sebe. Treba naglasiti kako će prilikom patroliranja promjena iznosa povjerenja utjecati na položaj unutar formacije, no u ovom poglavlju to ne razmatramo te robote smatramo fiksiranima.



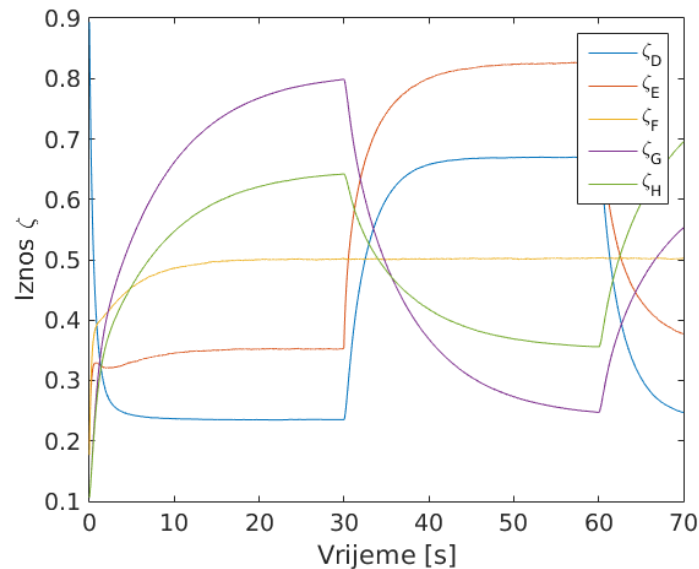
**Slika 2.10:** Pioneer roboti u V formaciji u simulacijskom okruženju Gazebo

Kao i u prethodnom potpoglavlju, analizirat će se signali koje šalje agent Delta iako agente više ne možemo smatrati jednakima zbog nesimetričnosti matrice susjedstva  $\mathbf{A}$ .



**Slika 2.11:** Vrijednosti mjerenja uz period permutiranja od 30 sekundi

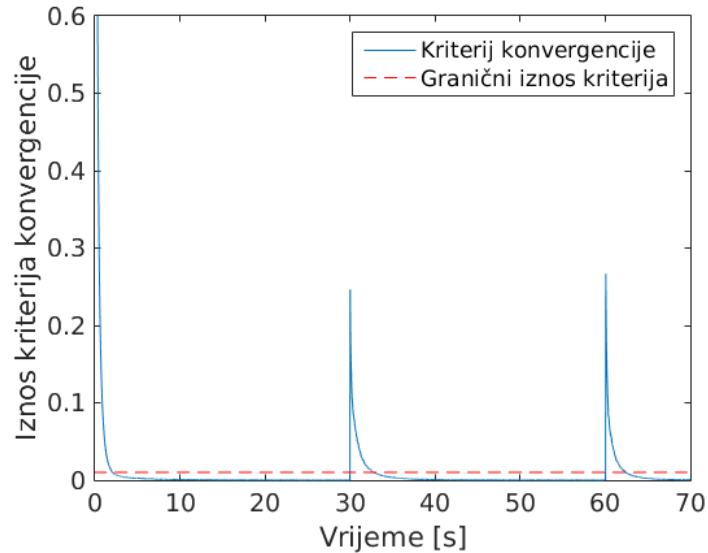
Na Slikama 2.11 i 2.12 se vidi kako kriterij konvergencije nije dobro izabran. Agent Delta puno sporije prima informacije o agentima Golfu i Hotelu zbog toga što mu nisu susjedi. Posljedica toga je sporija konvergencija povjerenja prema njima, te se ona ne postigne unutar 30 sekundi.



**Slika 2.12:** Povjerenje agenta Delte prema ostalim agentima. Može se uočiti kako povjerenja ne uspiju doći u stacionarno stanje nakon 30 sekundi.

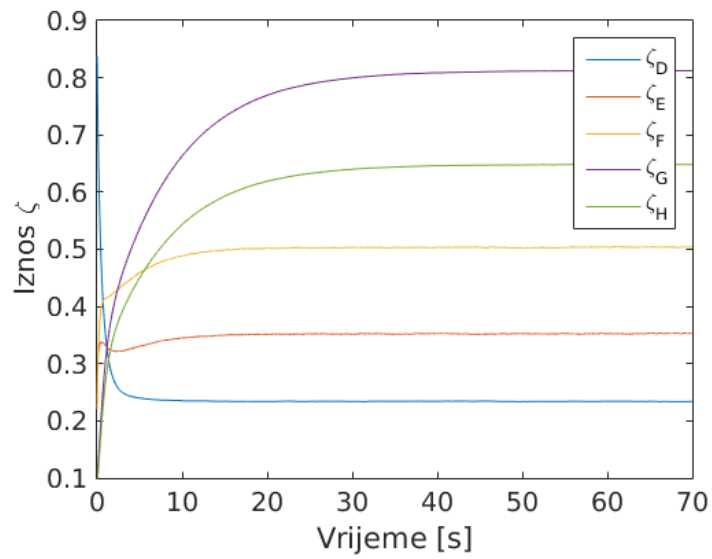
Međutim, na Slici 2.13 se vidi kako je iznos kriterija konvergencije ispod  $\varepsilon =$

$10^{-3}$  već oko treće sekunde kad ni poredak vrijednosti povjerenja prema agentima nije ispravan. Drastično smanjivanje kriterija konvergencije nije moguće jer bi zbog šuma u mjerenju bilo otežano detektiranje stacionarnog stanja. Zbog toga je povećan period permutiranja vektora  $\delta$ .

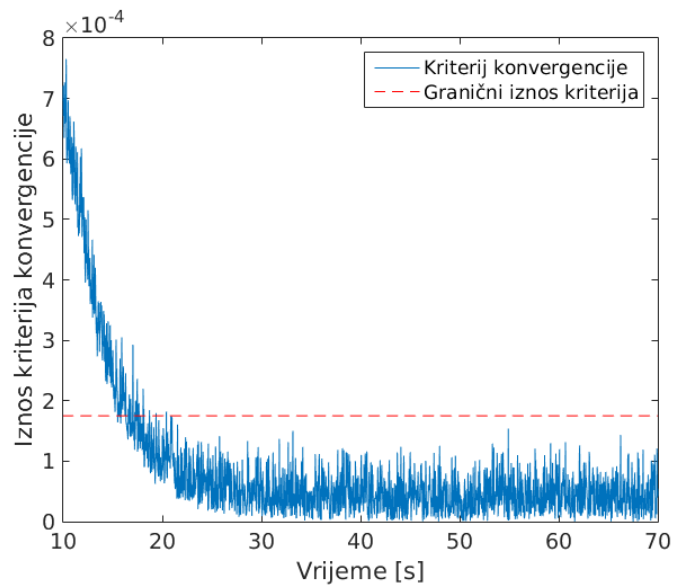


**Slika 2.13:** Iznos kriterija konvergencije za period promjene mjerenja od 30 sekundi.

Iz Slika 2.14 i 2.15, gdje je isključeno permutiranje vektora mjerenja  $\delta$ , se zaključuje kako je u V formaciji opravdano permutirati podatke s minimalnim periodom od 40 sekundi i pritom imati iznos  $\varepsilon = 1.75 \cdot 10^{-4}$ . Iako u 20. sekundi, kad algoritam zaključi kako je konsenzus postignut, povjerenja nisu u stacionarnom stanju, vrlo su mu blizu i vidljivo je kako dodatno smanjivanje kriterija konvergencije nije moguće zbog detekcije stacionarnog stanja.

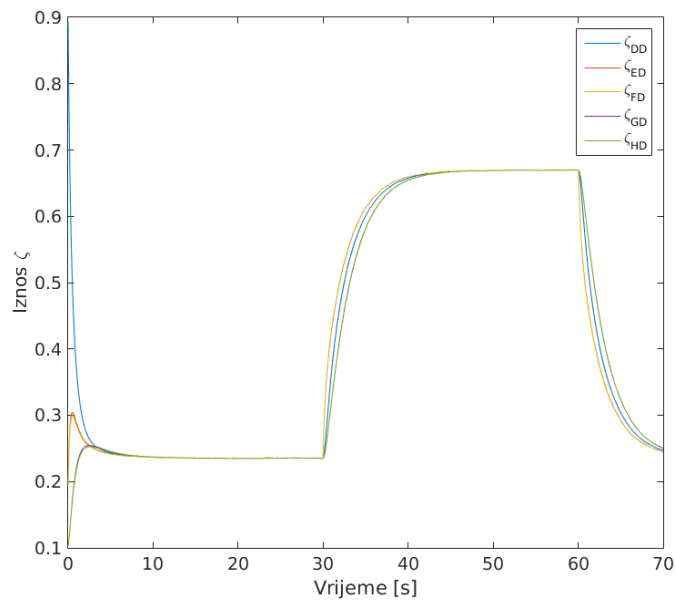


**Slika 2.14:** Povjerenje agenta Delte prema ostalim agentima uz isključeno permutiranje vektora mjerenja

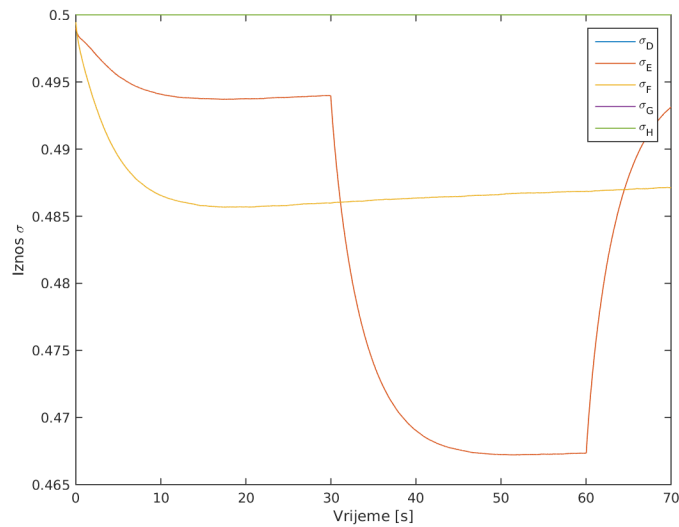


**Slika 2.15:** Iznos kriterija konvergencije uz isključeno permutiranje vektora mjerenja

Sa Slike 2.16 se može očitati kako su oblici odziva povjerenja ostalih agenata prema agentu Delta kvalitativno slični, a razlikuju se u brzini. Povjerenje najbrže računaju susjedi agenta Delta, Echo i Foxtrot, a najsporije susjedi susjeda agenta Delta, Golf i Hotel.



**Slika 2.16:** Povjerenje agenata prema agentu Delta u V formaciji



**Slika 2.17:** Pouzdanje agenta Delta prema drugim agentima

Na Slici 2.17 se vidi promjena pouzdanja agenta Delta u svoje susjede. Pouzdanje u susjede svojih susjeda, kao i samopouzdanje, ostaju jednaka početnima.

# 3. Upravljanje formacijom multi-agentskog sustava

## 3.1. Funkcija potencijala

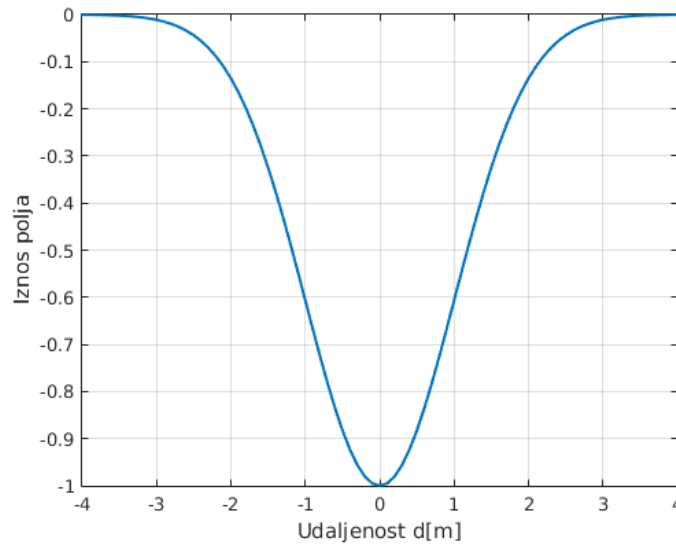
Jedan od osnovnih preduvjeta za koordinirano obavljanje zadataka kod multi-agentskih sustava jest sposobnost postizanja, održavanja i promjene različitih formacija te kretanje u formaciji. Postoji više različitih pristupa upravljanju formacijom, a u sklopu ovog rada formacijama robota upravljano je korištenjem umjetno generiranih polja potencijala. Ideja takvog pristupa jest da se formacijom upravlja koristeći gradijent neke funkcije potencijala - taj gradijent generira privlačnu ili odbojnu silu u odnosu na cilj ili prepreke robota [12]. Za svakog se robota u prostoru generira polje. Izvori polja su prepreke (ostali roboti u prostoru), a ponor polja je cilj robota. Cilj je da se, poput protona i elektrona, dva jednaka entiteta (2 robota) međusobno odbijaju, čime se izbjegava kolizija, a dva različita entiteta (robot i njegov cilj) privlače [13].

Polje potencijala za svakog robota generira se skupom elementarnih zvonolikih funkcija potencijala. Elementarna funkcija potencijala prepreka jest pozitivna, a elementarna funkcija potencijala cilja negativna funkcija. Predznak je stvar odabira; bitno je samo da se predznaci polja cilja i polja robota razlikuju, a ovakav je odabir napravljen jer nakon crtanja polja u 3-D ravnini, kako je prikazano na Slici 3.3, intuitivno djeluje da cilj (negativna funkcija) privlači robota k sebi, a prepreka (pozitivna funkcija) odbija. Gradijent tako definiranog polja potencijala generira privlačnu silu između agenta i cilja te odbojnu silu između dvaju agenata u multi-agentskom sustavu [14].

Elementarna funkcija potencijala cilja definirana je na sljedeći način:

$$u_c(d, \sigma_c) = -e^{\frac{-d^2}{2\sigma_c}} \quad (3.1)$$

gdje je  $d$  udaljenost robota od prepreke. Ovako definirana zvonolika funkcija potencijala cilja prikazana je na Slici 3.1 uz parametar  $\sigma_c = 1$  za parametar  $d$  (udaljenost) u rasponu  $[-4, 4]$  metra od robota.



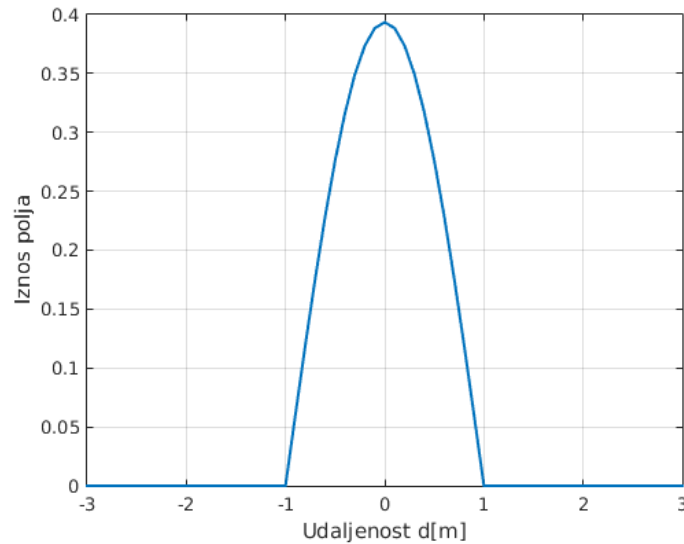
**Slika 3.1:** Funkcija potencijala cilja uz parametar  $\sigma_c = 1$

Elementarna funkcija potencijala prepreke definirana je na sljedeći način:

$$u_{p,i}(d, \sigma_p) = \begin{cases} e^{\frac{-d^2}{2\sigma_p}} - e^{\frac{-1}{2\sigma_p}}, & d < 1 \\ 0, & d \geq 1 \end{cases} \quad (3.2)$$

gdje je  $d$  udaljenost robota od cilja. U slobodnom se prostoru robot treba gibati neome-  
tano; ako u blizini nema prepreka, on mora ići usmjerenom prema cilju. Također, budući  
da dizajniramo decentralizirani sustav, u kojem je svaki agent upravljani samostalno, a  
ne preko jednog centralnog računala, u stvarnosti postoji određen komunikacijski ra-  
dijus izvan kojeg dva robota ne mogu detektirati jedan drugoga. Zbog tih je razloga  
opravdano zanemariti djelovanje polja prepreke na robota izvan određenog radijusa,  
odnosno *odsjeći* polje izvan tog radijusa. Ta je udaljenost postavljena na  $d_{min} = 1[m]$

jer je dovoljno mala da omogućuje konvergenciju formacije u relativno uskim prostorima bez naprednih metoda upravljanja, a ipak dovoljno velika da nikad ne dolazi do sudara. Kada bismo funkciju potencijala definirali kao običnu zvonoliku funkciju za radijus manji od  $d = 1[m]$ , a kao nulu izvan tog radijusa, funkcija potencijala ne bi bila kontinuirana te bi došlo do naglog skoka polja za  $d = 1[m]$  što bi uzrokovalo probleme s upravljanjem. Stoga se funkcija potencijala prepreke *spušta* s članom  $-e^{\frac{-1}{2\sigma_p}}$  čime je osigurana kontinuiranost polja. Ovako definirana funkcija potencijala prepreke prikazana je na Slici 3.2 uz parametar  $\sigma_p = 1$  za parametar  $d$  (udaljenost) u rasponu  $[-3, 3]$  metra od robota.



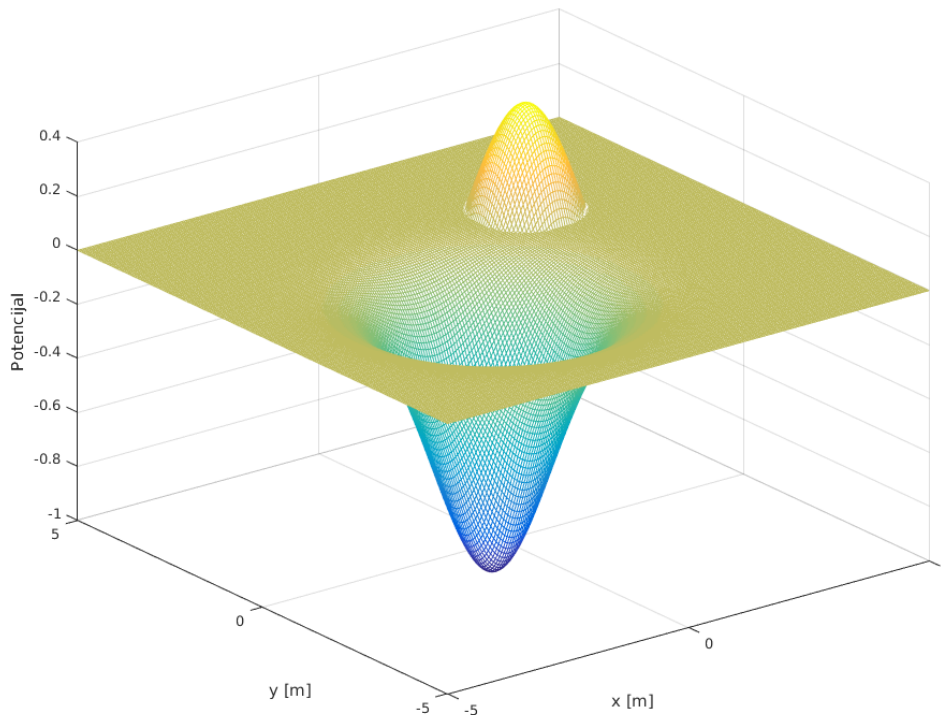
**Slika 3.2:** Funkcija potencijala prepreke uz parametar  $\sigma_p = 1$

Nakon što su definirane elementarne funkcije potencijala prepreke i cilja, lako je kreirati polje potencijala nekog robota; ono je zbroj elementarnih funkcija potencijala svih prepreka u prostoru te elementarne funkcije potencijala cilja tog robota:

$$u(d, \sigma) = \sum_{i=1}^n u_{p,i}(d, \sigma_p) + u_c(d, \sigma_c) \quad (3.3)$$

gdje je  $n$  broj prepreka u prostoru. Udaljenost robota od cilja  $d$  u prostoru jest Euklidska norma vektora udaljenosti  $d = \sqrt{x^2 + y^2}$ . Ovako definirano polje potencijala u prostoru s jednom preprekom prikazano je na Slici 3.3.





**Slika 3.3:** Polje potencijala u prostoru s jednom preprekom

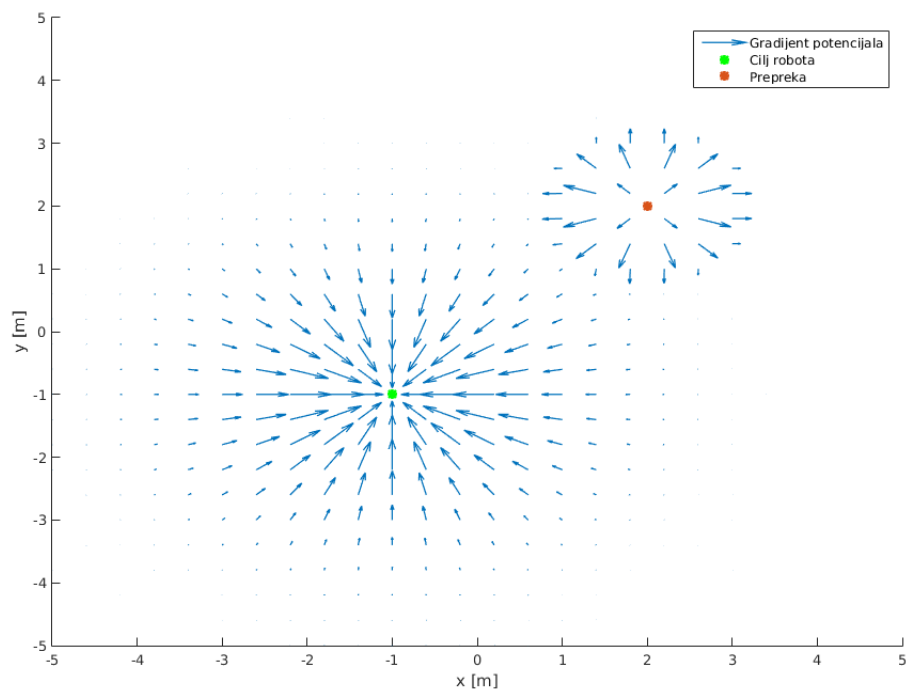
Gibanjem robota prema cilju vrijednost polja potencijala na poziciji robota u prostoru se smanjuje, a gibanjem robota prema prepreci vrijednost polja na poziciji robota se povećava. Vrijedi i obratno; pozitivna promjena vrijednosti polja između neka dva trenutka znači da je robot prišao bliže prepreci, dok negativna promjena vrijednosti polja između neka dva trenutka znači da je robot prišao bliže cilju. Dakle, ako želimo da robot ide prema cilju, potrebno ga je usmjeriti u smjeru negativne promjene polja potencijala. To znači da nam je za dobivanje upravljačkog zakona robota potrebno računati derivaciju, odnosno, budući da je polje potencijala dvodimenzionalno, gradijent polja te mu još okrenuti predznak [11].

$$g(d, \sigma) = -\nabla u(d, \sigma) \quad (3.4)$$

Budući da vrijedi  $d = \sqrt{x^2 + y^2}$ , a parametri  $\sigma_c$  i  $\sigma_p$  su konstantne vrijednosti, može se pisati:

$$g(x, y) = -\nabla u(x, y) = -\frac{\delta u}{\delta x} \mathbf{i} - \frac{\delta u}{\delta y} \mathbf{j} \quad (3.5)$$

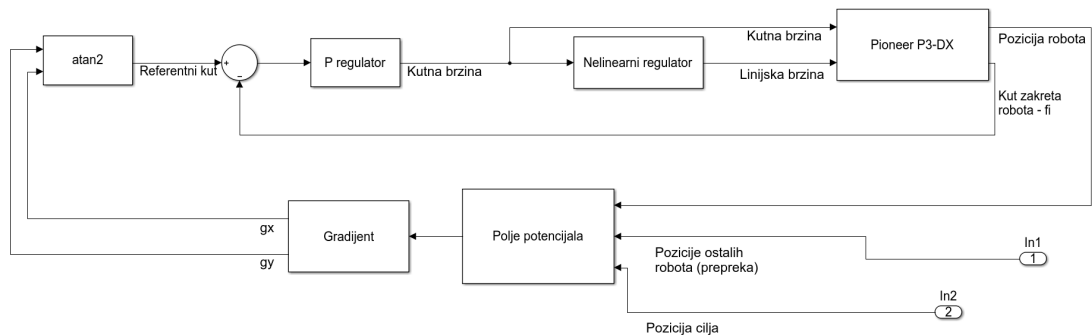
Iz gradijenta polja potencijala možemo odrediti smjer u kojem polje privlači ili odbija robota, odnosno možemo odrediti referentni smjer u kojem se robot treba gibati. Gradijent polja potencijala definiran izrazom (3.5) prikazan je na Slici 3.4. Jasno je vidljivo kako cilj privlači robota k sebi, a prepreka ga odbija od sebe.



**Slika 3.4:** Prikaz gradijenta polja potencijala u xy ravnini

## 3.2. Upravljanje položajem i orijentacijom robota

U prethodnom je potpoglavlju definirano polje potencijala i njegov gradijent. Polje potencijala određuje se pomoću elementarnih funkcija potencijala iz poznatih pozicija cilja i prepreka u prostoru te se računa u točki u kojoj se robot trenutno nalazi. Pozicija cilja robota je ulaz sustava, a pozicija robota i pozicije njegovih prepreka povratna su veza cjelokupnog multi-agentskog sustava. Na temelju tog ulaza i povratne veze generira se polje potencijala iz čijeg se gradijenta određuje referenca smjera robota. Robot se prema cilju giba prateći referentni smjer; od referentnog se kuta oduzima trenutni zakret robota u  $(x, y)$  ravnini da se dobije signal pogreške koji se zatim dovodi na regulator iz kojeg se dobije upravljački signal brzine robota [15]. Cjelokupni je zatvoreni sustav regulacije pozicije i orijentacije robota prikazan na Slici 3.5, a pojedini dijelovi sustava razrađeni su u narednim potpoglavljima.



Slika 3.5: Regulacijska petlja upravljanja položajem i orijentacijom robota

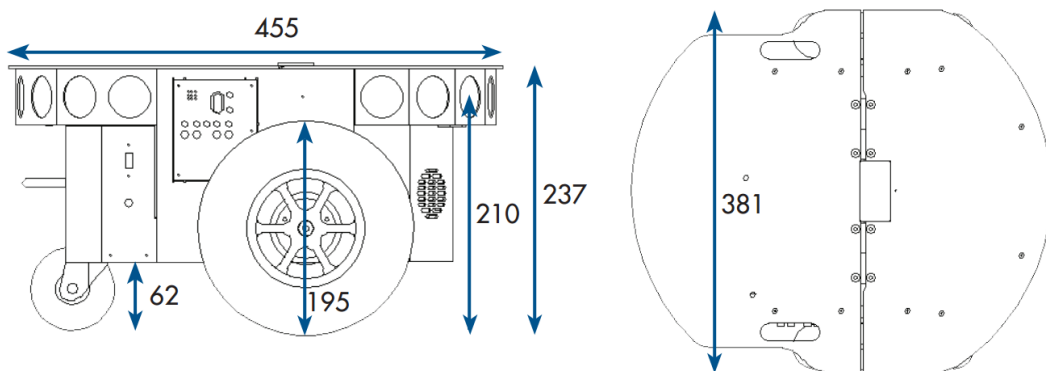
### 3.2.1. Tehnički aspekti implementacije

Robot Pioneer P3-DX jedan je od najpopularnijih mobilnih robota korištenih u istraživanjima [16]. Svaki od robota ima vlastito računalo s Linux operacijskim sustavom i Robot Operating System (ROS) okruženjem što je bio osnovni preduvjet za decentralizirano upravljanje robotima. Fotografija robota prikazana je na Slici 3.6, a tlocrt i bokocrt robota te njegove dimenzije prikazane su na Slici 3.7. Robot u Gazebo simulacijskom okruženju prikazan je na Slici 1.1.

S obzirom da se ovaj rad bavi isključivo upravljanjem robotima, ovdje nije opisan fizikalni model, odnosno dinamika i kinematika robota, već samo informacije bitne



**Slika 3.6:** Slika robota Pioneer P3-DX u laboratoriju



**Slika 3.7:** Bokocrt i tlocrt Pioneer P3-DX robota s navedenim dimenzijama [mm]

za implementaciju upravljačkih algoritama. Korišteni su gotovi upravljački programi robota koji postoje u sklopu ROS okruženja. Robot ima dva kotača s diferencijalnim pogonom, odnosno svaki je kotač zasebno pogonjen, što znači da mu nije moguće direktno zadati brzine u smjeru osi  $x$  i  $y$ , već mu je potrebno zadati smjer odnosno kutnu brzinu u  $(x, y)$  ravnini (oko  $z$  osi) te iznos brzine u tom smjeru. Za potrebe simulacije korišten je gotov model robota u Gazebo simulacijskom okruženju, a upravljanje robotima u simulaciji identično je kao upravljanje u stvarnosti. Važno je naglasiti da je upravljanje robotima decentralizirano, odnosno da svaki robot ima svoj regulator (implementiran na vlastitome računalu).

Za povratnu vezu, odnosno estimaciju pozicija i orijentacija svih robota u multi-

agentskom sustavu u laboratorijskom okruženju korišten je sustav Optitrack. To je sustav od 16 kamera koji na temelju infracrvenog svjetla detektira i prati markere postavljane na robote. Na temelju detektiranih markera, određuje se orijentacija i pozicija robota. Sustav Optitrack povezan je s ROS okruženjem. U simulacijskom okruženju Gazebo svi podaci o pozicijama i orijentacijama su također povezani s ROS okruženjem.

Ukratko, ROS okruženje rješava *low-level* tehničke probleme sustava upravljanja te funkcionira kao svojevrsna *crna kutija*; ulazni signali pojedinog robota su njegova kutna i linijska brzina, a izlazni signali njegova pozicija i orijentacija. Nad tim se ulazima i izlazima implementira sustav upravljanja.

### 3.2.2. Generiranje reference robota

Iz povratne veze sustava, odnosno poznatih trenutnih pozicija i orijentacija pojedinih robota možemo generirati polje potencijala svakog robota definirano prema jednadžbi (3.3). Kako je definirano u potpoglavlju 3.1, gradijent polja potencijala u nekoj točki govori koliko je ukupno djelovanje cilja (privlačenje) i prepreka (odbijanje) na robota u smjeru  $x$  osi i u smjeru  $y$  osi. Robot Pioneer P3-DX kao referencu prima linijsku i kutnu brzinu  $(v_l, v_k)$ , stoga je potrebno iz Kartezijevog  $(x, y)$  koordinatnog sustava prijeći u polarni  $(r, \phi)$  koordinatni sustav. Kut  $\phi$  gradijenta potencijala lako se odredi prema sljedećoj jednadžbi:

$$\phi = \text{atan}\left(\frac{g_y}{g_x}\right) \quad (3.6)$$

gdje su  $g_x$  i  $g_y$  komponente gradijenta potencijala u smjeru  $x$  i  $y$  osi. Taj kut definira točan smjer u kojem polje potencijala privlači ili odbija robota te je upravo to referenca sustava upravljanja. Pri tome je potrebna točnost u sva četiri kvadranta  $(x, y)$  ravnine te je stoga u implementaciji korištena *Python* funkcija  $\text{atan2}(x, y)$  koja uz pomoć predznaka argumenata određuje točan kvadrant.

Važno je i spomenuti da se regulacijsko odstupanje ne generira običnim oduzimanjem stvarnog kuta od referentnog kuta jer tako dobiveno regulacijsko odstupanje može dovesti do redundantnog kruženja robota. Primjerice, ako je referenca kuta  $179^\circ$ , a mjereni kut robota  $-179^\circ$ , običnim oduzimanjem robotu bi bio zadan zakret od  $358^\circ$ ,

dakle gotovo cijeli krug, a robot se u stvarnosti treba zakrenuti svega  $-2^\circ$ . Iz tog je razloga implementirana funkcija koja će računati najbliži potreban zakret između referentnog i mjenog kuta. Taj zakret je regulacijsko odstupanje koje koriste regulatori.

### 3.2.3. Regulacija brzina robota

Za regulaciju kutne brzine robota korišten je obični P regulator. Regulacijsko odstupanje množi se s nekim parametrom  $K$ . Taj smo parametar eksperimentalno utvrdili tako da je odziv kuta robota stabilan, relativno brz i bez nadvišenja. Povećavanjem iznosa parametra  $K$  postiže se agresivnije ponašanje robota; brže vladanje, ali uz mnogo nepotrebnog okretanja robota, a smanjenjem iznosa parametra  $K$  postiže se suprotni učinak [17].

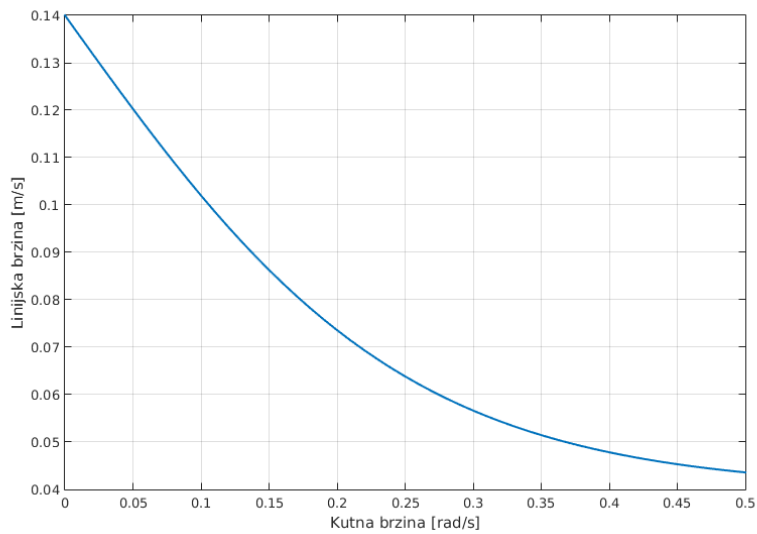
Budući da želimo postići što brži dolazak robota u cilj uz uspješno izbjegavanje prepreka, želimo da se robot po apsolutnom iznosu brzine giba brzo u prostoru gdje nema prepreka, a relativno sporo u prostoru gdje ima prepreka, kako bi ih uspješno izbjegao. Kada je kutna brzina mala, robot se u većini slučajeva giba usmjerenom prema cilju što znači da u njegovoj okolini nema prepreka. Tada želimo veliki iznos linijske brzine. Kada je kutna brzina velika, robot je vrlo vjerojatno ušao u polje drugog robota, odnosno prepreke, te ga to polje odbija od sebe - zakret robota se mora promijeniti. U tom slučaju želimo mali iznos linijske brzine kako bi robot uspješno skrenuo i izbjegao prepreku.

Ovakvo smo ponašanje modelirali uz pomoć funkcije tangens hiperbolički, prema izrazu (3.7):

$$v_l = v_{l,max} - K_1 \tanh(|K_2 \omega_k|) \quad (3.7)$$

gdje su  $v_l$  linijska brzina,  $\omega_k$  kutna brzina,  $v_{l,max}$  maksimalna dozvoljena linijska brzina, a  $K_1$  i  $K_2$  promjenjivi parametri. Parametri  $K_1$  i  $K_2$  određeni su eksperimentalno, ponavljanjem pokusa za različite parametre sve dok nisu dobiveni zadovoljavajući rezultati, odnosno sve dok nije postignuto sigurno skretanje kako bi se izbjegla prepreka te relativno visok iznos linijske brzine u području bez prepreka. Iako na prvi pogled ovako definiran odnos kutne i linijske brzine izgleda dosta slično linearnom, eksperimentalno je utvrđeno da je ponašanje robota s odnosom linijske i kutne brzine defini-

ranom prema (3.7) superiorno ponašanju robota s linearnim odnosom linijske i kutne brzine. To slijedi iz činjenice da se ovako definiranim regulatorom za velike kutne brzine linijska brzina jače smanjuje nego kod linearne ovisnosti, što znači sigurnije skretanje robota i sigurno izbjegavanje kolizije. Prikaz ovisnosti linijske brzine robota o kutnoj brzini robota vidljiv je na Slici 3.8. Dobivena kutna i linijska brzina robota šalju se preko ROS okruženja upravljačkim programima robota koji informacije prosleđuju motorima (aktuatorima) koji izvršavaju traženi zadatak.



**Slika 3.8:** Nelinearna karakteristika ovisnosti iznosa linijske brzine o iznosu kutne brzine robota

# 4. Simulacija decentraliziranog upravljanja formacijom

## 4.1. Promjena formacije u Gazebo simulatoru

U ovome je potpoglavlju na jednom primjeru razrađena i objašnjena statička promjena formacije multi-agentskog sustava u Gazebo simulacijskom okruženju kako bi se bolje razjasnilo upravljanje formacijom opisano u 3. poglavlju. Radi se o pet robota (agenta) koji međusobno mijenjaju mjesta unutar postojeće trokutaste ( $\mathbf{V}$ ) formacije. U početnoj je formaciji vođa agent Delta, njemu s desne strane agenti Golf i Foxtrot, a s lijeve strane agenti Echo i Hotel. Tražena je formacija u kojoj je vođa agent Echo, s desna su mu agenti Hotel i Delta, a s lijeva agenti Foxtrot i Golf. Početni je položaj robota unutar trokutaste formacije dan vektorima  $x_0$  i  $y_0$ , a završni je položaj robota dan vektorima  $x_c$  i  $y_c$ .

$$\mathbf{x}_0 = \begin{bmatrix} x_D & x_E & x_F & x_G & x_H \end{bmatrix} = \begin{bmatrix} 0 & -2 & -4 & -2 & -4 \end{bmatrix} \quad (4.1)$$

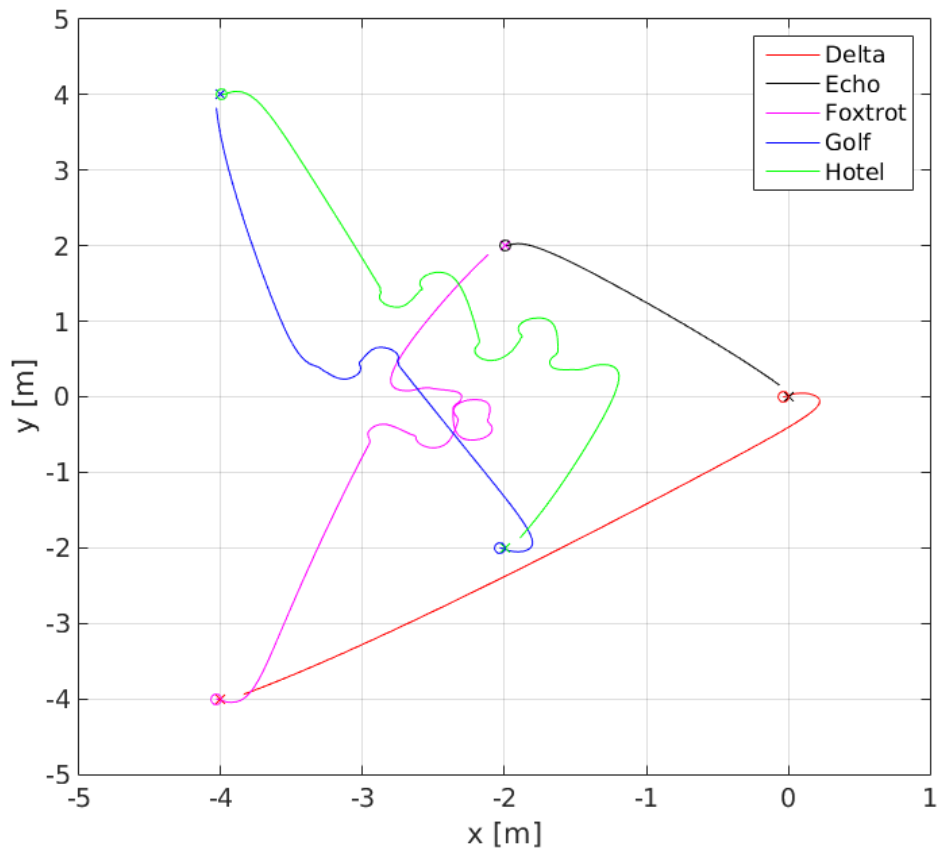
$$\mathbf{y}_0 = \begin{bmatrix} y_D & y_E & y_F & y_G & y_H \end{bmatrix} = \begin{bmatrix} 0 & 2 & -4 & -2 & 4 \end{bmatrix} \quad (4.2)$$

$$\mathbf{x}_c = \begin{bmatrix} -4 & 0 & -2 & -4 & -2 \end{bmatrix} \quad (4.3)$$

$$\mathbf{y}_c = \begin{bmatrix} -4 & 0 & 2 & 4 & -2 \end{bmatrix} \quad (4.4)$$

Putanje robota, zajedno s početnim i krajnjim pozicijama robota, mogu se vidjeti na Slici 4.1. Početne pozicije robota na Slici označene su simbolom  $\mathbf{o}$ , a završne pozicije označene su simbolom  $\mathbf{x}$ .





**Slika 4.1:** Putanje robota prilikom promjene formacije

Analizu gibanja agenata pri promjeni formacije započet ćemo od najjednostavnije prema najsloženijoj putanji pojedinog robota. Najjednostavnija putanja je ona agenta Echoa. Budući da se on u svakom trenutku nalazi na udaljenosti većoj od jednog metra u odnosu na ostale agente, komponente drugih agenata - prepreka u polju potencijala robota Echoa jednake su nuli. Dakle, Echovo polje potencijala generira isključivo njegov cilj. To znači da, sukladno tvrdnjama i definicijama iznesenim u 3. poglavlju, negativni gradijent tog polja potencijala daje referencu gibanja Echou točno u smjeru njegovog cilja. Upravo tako se agent Echo gibao u simulatoru, njegova putanja gotovo u potpunosti izgleda kao pravac. Kutnu brzinu, odnosno nelinearno kretanje u prostoru, ima samo u početnim trenucima. Jednom kada je robot usmjeren prema cilju kreće se po pravcu.

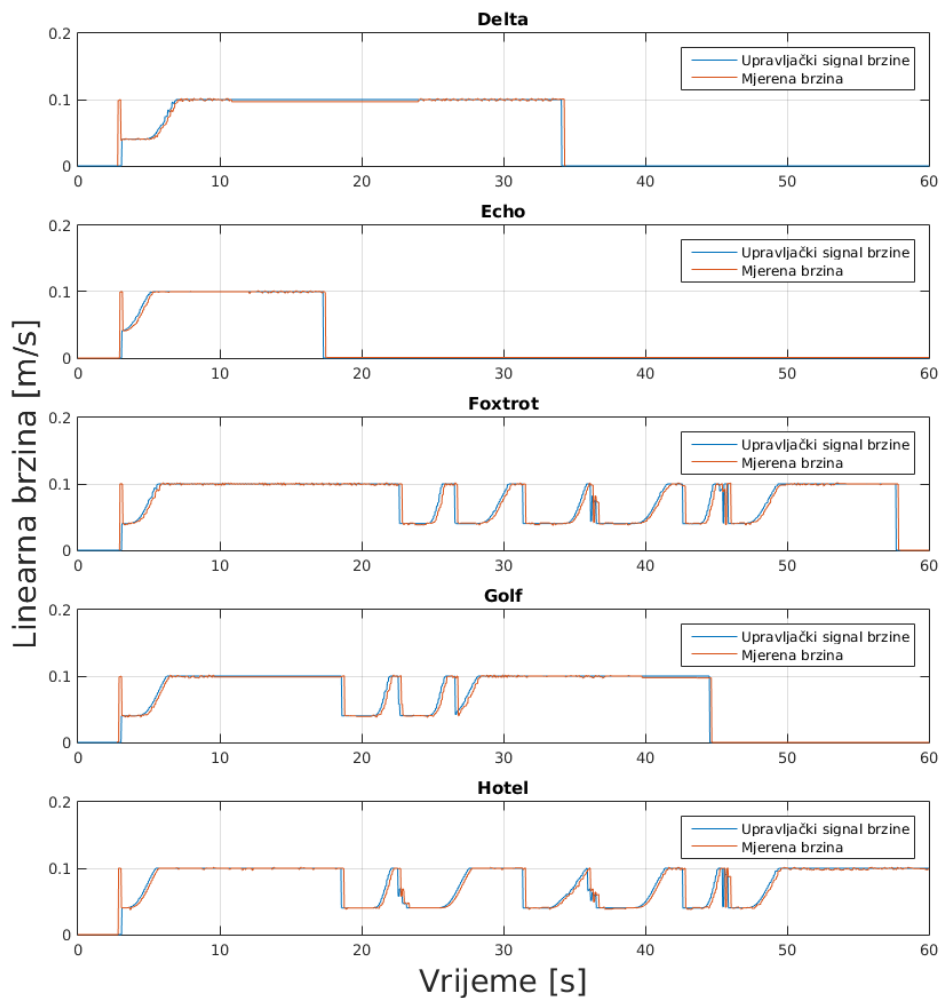
Putanja agenta Delte također većim dijelom sličí pravcu. Razlozi su jednaki kao i kod agenta Echoa; niti u jednom trenutku prepreke robota nisu u radijusu detekcije te mu je zbog toga referenca gibanja, određena gradijentom polja potencijala, cijelo vrijeme usmjerena izravno prema cilju. Na početku putanje, prije nego što se robot počne gibati po pravcu, može se zamijetiti polukružno skretanje. To se događa zato što robot ima dinamička ograničenja, odnosno ne može trenutno postići traženi kut. Dok se okreće dobije referencu linearne brzine različitu od nule te se stoga ne okreće u mjestu već radi kratko polukružno gibanje. Takvo je ponašanje očekivano u radu sa stvarnim robotima (odnosno njihovim modelima) budući da oni nisu točkasti, svesmjerni, i nije značajnije utjecalo na ponašanje cijelog multi-agentskog sustava. Takvo je ponašanje prisutno i kod ostalih agenata, primjerice kod već analiziranog agenta Echoa, samo što je kod agenta Echoa to bilo manje izraženo jer je razlika njegovog referentnog i početnog kuta bila manja nego kod agenta Foxtrota.

Agent Golf kreće svoje gibanje kao i agenti Foxtrot i Delta: njegove su prepreke izvan dozvoljenog radijusa te se počinje gibati usmjereno prema cilju. U jednome trenutku, otprilike kada se nalazi na koordinatama  $(-2.7, 0.5)[m]$ , agent Hotel ulazi u radijus detekcije prepreke agenta Golfa. U polju potencijala agenta Golfa sada se, osim utjecaja njegovog cilja, nalazi i utjecaj agenta Hotela. Budući da mu je prepreka bliže od cilja, agent Golf iz gradijenta polja potencijala dobiva referencu kuta koja ga jako tjera od prepreke. Radi naglo skretanje od prepreke sve dok ne bude na dozvoljenoj udaljenosti ( $1[m]$ ). Tek što se odmakne na sigurnu udaljenost od agenta Hotela, u njegov radijus detekcije ulazi agent Foxtrot i stvara promjenu u polju potencijala. Agent Golf se ponovo nastoji maknuti od utjecaja prepreke te radi naglo skretanje. Kada se odmakne od prepreke okreće se prema cilju i sve do dolaska u cilj giba se usmjereno, gotovo po pravcu.

Svaki od pet robota ostala četiri robota vidi kao prepreke. Budući da se polja potencijala svakog robota generiraju na jednak način, ulazak jednog robota u radijus detekcije, odnosno u polje drugog robota znači da je i drugi robot ušao u polje prvog robota. Dakle, roboti se međusobno jednako odbijaju, odnosno analogno 3. Newtonovom zakonu akcija uzrokuje reakciju. To znači da je, u trenucima kada su agenti Hotel i Foxtrot djelovali na agenta Golfa i agent Golf djelovao na njih, odnosno *tjerao*

ih od sebe. To se jasno vidi po putanjama prikazanim na Slici 4.1 i objašnjava putanje preostala dva robota, agenata Foxtrota i Hotela. Dakle, sukladno već objašnjenim ponašanjima, agent Hotel se krene gibati usmjereno prema cilju, kada mu u polje uđe prepreka on se naglim skretanjem nastoji što prije maknuti od nje, zatim se ponovo usmjerava prema cilju i ponavlja te obrasce ponašanja sve dok ne dođe u zadani cilj. Isto vrijedi i za agenta Foxtrota s time da su se u njegovom polju u jednom trenutku sa suprotnih strana našle dvije prepreke (agenti Golf i Hotel) te je on napravio kratko redundantno kružno gibanje. To kružno gibanje ne predstavlja problem niti opasnost od sudara upravo iz razloga da su prepreke dinamičke, a ne statičke; i prepreke se nastoje odmaknuti od robota, a ne samo robot od prepreka.

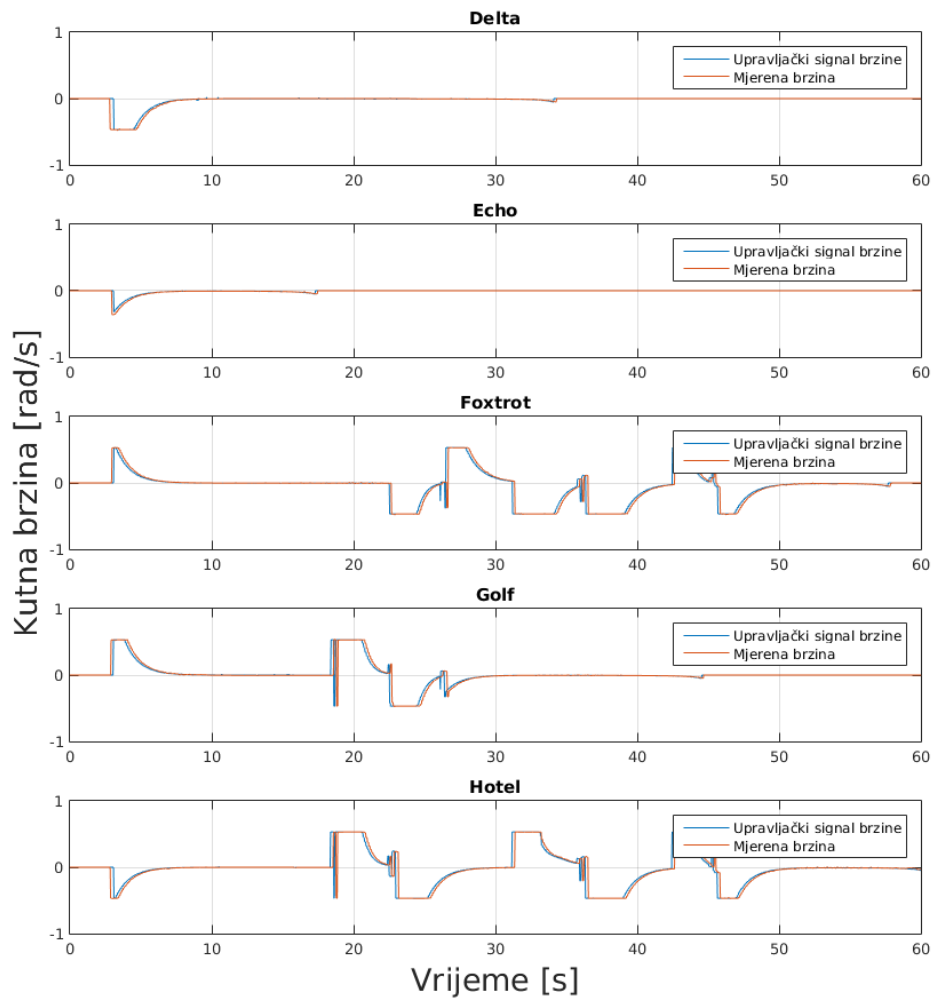
Na Slici 4.2 prikazani su upravljački signali linearne brzine svih agenata te mjerne izlazne brzine. Sa slike se mogu vidjeti trenuci u kojima gibanje prestaje; kada robot dostigne svoj cilj linearna brzina pada na nulu. Vidljivo je i da se robot, kada je gibanje usmjereno prema cilju, odnosno kada su prepreke izvan radijusa detekcije, giba konstantnom linearnom brzinom. U simulaciji su, očekivano, roboti imali stacionarnu točnost brzine tj. pojačanje 1. Sa Slike 4.2 se vidi da mjereni signal iznosom i oblikom gotovo u potpunosti prati upravljački signal, a kašnjenje signala je reda veličine desetinke sekunde, odnosno gotovo zanemarivo. To je očekivano u simulacijskom okruženju gdje se sve izvršava na jednom računalu, no za očekivati je da će realan sustav u laboratoriju imati nešto veće kašnjenje kao posljedicu kašnjenja komunikacije između svih robota međusobno te Optitrack sustava. Zamjetan je i određeni, po amplitudi malen, mjerni šum. Sa slike se također mogu zamijetiti trenuci u kojima prepreka ulazi u polje robota; linearna brzina tada pada sukladno izrazu (3.7).



**Slika 4.2:** Linearne brzine svih robota prilikom promjene formacije

Na Slici 4.3 prikazani su upravljački signali kutne brzine svih agenata te mjerene izlazne brzine. Vidi se kako na početku gibanja svi roboti imaju neku kutnu brzinu; okreću se prema cilju. Zatim kutna brzina iščezava kako agenti idu prema cilju. Tek ako se u polju robota pojavi prepreka dolazi do promjene kutne brzine, ta je promjena brzine skokovita zato što je korišten P tip regulatora, a pri ulasku prepreke u radijus detekcije dogodi se nagli skok u polju, a time i nagla, skokovita promjena reference. Kao i kod odziva linearnih brzina, mjereni signal ima jako malo kašnjenje (reda veličine desetinke sekunde). Mjerena brzina iznosom i oblikom gotovo u potpunosti prati upravljački signal.

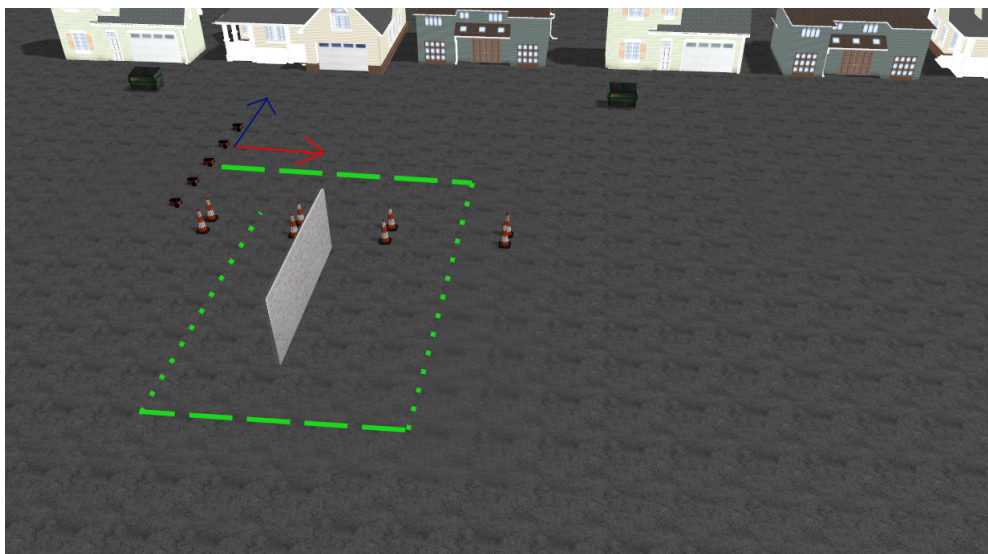
Snimka eksperimenta opisanog u ovom poglavlju nalazi se u priloženom filmskom isječku *zamjena\_simulacija.mp4*.



**Slika 4.3:** Kutne brzine svih robota prilikom promjene formacije

## 4.2. Patroliranje u Gazebo simulatoru

U ovom će se potpoglavlju razraditi patroliranje robota u formaciji u Gazebo simulatoru. Položaji robota unutar formacije bit će određen vrijednostima povjerenja prema algoritmu konsenzusa opisanom u drugom poglavlju (2.2). Za regulaciju linearne i kutne brzine korištena je gradijentna metoda opisana u prethodnom potpoglavlju. Vođa (robot prema kojem je povjerenje najveće) će opisivati pravokutnik dimenzija  $10m \times 20m$ . Prikaz scenarija s putanjom kretanja vođe (zelena crta) i koordinatnim osima se vidi na Slici 4.4. Crvenom bojom označena je  $x$  os, a plavom bojom  $y$  os.



**Slika 4.4:** Scenarij s putanjom kretanja vođe i koordinatnim osima. Crtkano su označene dionice koje roboti prelaze u **V** formaciji, a točkano dionice koje roboti prelaze u **I** formaciji.

S obzirom da se grupom robota upravlja decentralizirano, vođa je jedini agent koji prima informaciju o svom referentnom položaju, željenoj formaciji i zakretu grupe, te ju preko susjeda prosljeđuje ostalim robotima. Zakret grupe je kut koji formacija ima s obzirom na početan položaj na Slici 4.4. Na primjer, prvo skretanje formacije udesno će biti obavljeno na način da se zakret poveća s 0 na  $-\frac{\pi}{2}$  radijana. U ovom će se radu koristiti dvije formacije:

- **V** formacija s poretkom robota kao na Slici 4.5
- **I** formacija s poretkom robota kao na Slici 4.6

Roboti će se kretati u **V** formaciji dok je to moguće zbog toga što tad mogu prekriti više

površine. To je poželjno jer tad mogu bolje obaviti određene zadatke poput mapiranja prostora i raznih mjerenja. Kad roboti dođu do prepreke (u ovom slučaju stošci) više se ne mogu gibati u **V** formaciji jer je razmak između stožaca premalen. Zato prelaze u **I** formaciju i nastavljaju gibanje u toj formaciji do prvog skretanja.



**Slika 4.5:** Agenti u **V** formaciji s označenim redoslijedom po povjerenju

Kada se roboti nalaze u **V** formaciji, njihov je referentni položaj određen koordinatama kako slijedi:

$$x_1 = x_r \quad (4.5)$$

$$y_1 = y_r \quad (4.6)$$

$$x_2 = x_r - 2\sqrt{2} \cdot \cos\left(\frac{\pi}{4} + \phi\right) \quad (4.7)$$

$$y_2 = y_r - 2\sqrt{2} \cdot \sin\left(\frac{\pi}{4} + \phi\right) \quad (4.8)$$

$$x_3 = x_r - 2\sqrt{2} \cdot \cos\left(-\frac{\pi}{4} + \phi\right) \quad (4.9)$$

$$y_3 = y_r - 2\sqrt{2} \cdot \sin\left(-\frac{\pi}{4} + \phi\right) \quad (4.10)$$

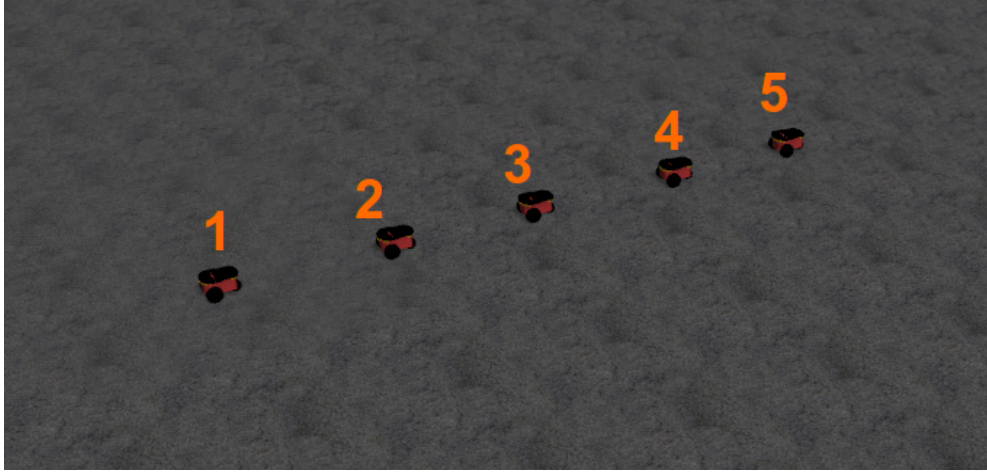
$$x_4 = x_r - 4\sqrt{2} \cdot \cos\left(\frac{\pi}{4} + \phi\right) \quad (4.11)$$

$$y_4 = y_r - 4\sqrt{2} \cdot \sin\left(\frac{\pi}{4} + \phi\right) \quad (4.12)$$

$$x_5 = x_r - 4\sqrt{2} \cdot \cos\left(-\frac{\pi}{4} + \phi\right) \quad (4.13)$$

$$y_5 = y_r - 4\sqrt{2} \cdot \sin\left(-\frac{\pi}{4} + \phi\right) \quad (4.14)$$

Gdje su  $x_i$  i  $y_i$  referentna  $x$  i  $y$  koordinata  $i$ -tog robota u poretku po iznosu povjerenja, a  $\phi$  kut zakreta formacije. Referentne koordinate vođe su označene s  $x_r$  i  $y_r$ .



**Slika 4.6:** Agenti u **I** formaciji s označenim redoslijedom po povjerenju

Kada se roboti nalaze u **I** formaciji, njihov je referentni položaj određen koordinatama kako slijedi, uz iste oznake koje vrijede za **V** formaciju:

$$x_1 = x_r \quad (4.15)$$

$$y_1 = y_r \quad (4.16)$$

$$x_2 = x_r - 1.5\sqrt{2} \cdot \cos(\phi) \quad (4.17)$$

$$y_2 = y_r - 1.5\sqrt{2} \cdot \sin(\phi) \quad (4.18)$$



$$x_3 = x_r - 3\sqrt{2} \cdot \cos(\phi) \quad (4.19)$$

$$y_3 = y_r - 3\sqrt{2} \cdot \sin(\phi) \quad (4.20)$$

$$x_4 = x_r - 4.5\sqrt{2} \cdot \cos(\phi) \quad (4.21)$$

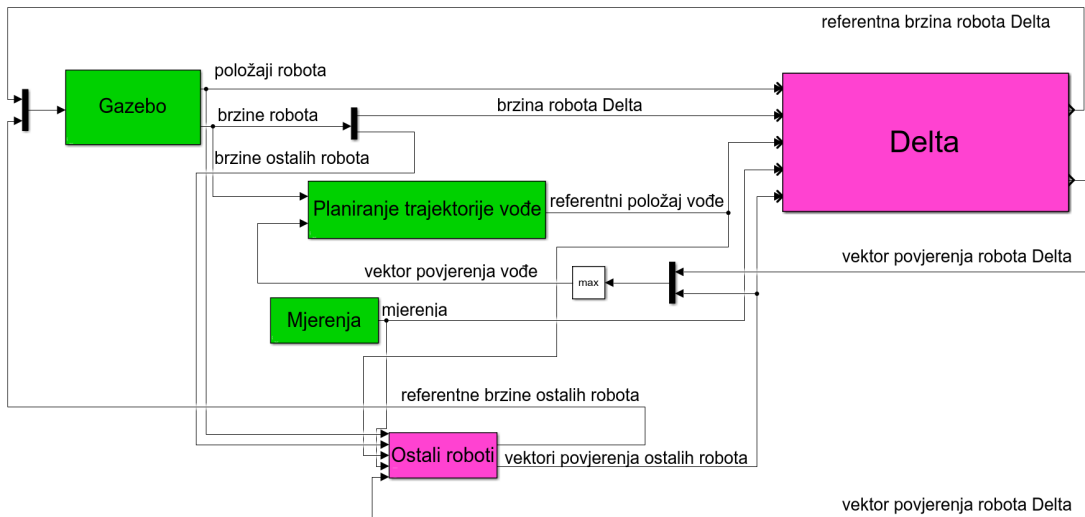
$$y_4 = y_r - 4.5\sqrt{2} \cdot \sin(\phi) \quad (4.22)$$

$$x_5 = x_r - 6\sqrt{2} \cdot \cos(\phi) \quad (4.23)$$

$$y_5 = y_r - 6\sqrt{2} \cdot \sin(\phi) \quad (4.24)$$

Planiranje trajektorije patroliranja je izvršeno programskom petljom s frekvencijom  $f = 30Hz$ . Referentna brzina patroliranja je 0.2 m/s, ukoliko se roboti nalaze u zadanoj formaciji i konsenzus je postignut. Ako konsenzus nije postignut, roboti stanu, pričekaju dok se konsenzus ne postigne i zauzmu nova mjesta unutar formacije. Kad je svaki robot zauzeo svoje mjesto, formacija nastavlja gibanje. Promjena željene formacije iz **V** u **I** ili obratno također zaustavlja formaciju.

Prikaz komunikacije u decentraliziranom načinu rada vidi se na Slici 4.7. Izdvojen je agent Delta zbog preglednosti. Zelenom bojom je označen dio čvorova koji se izvodi na centralnom računalu, a ljubičastom bojom dio koji se može izvoditi zasebno na svakom robotu. U kontekstu simulacije, decentraliziranost algoritma postignuta je tako da se čvorovi izvode u odvojenim procesima. Komunikacija između čvorova odvija se preko ROS komunikacijskog sučelja.



Slika 4.7: Prikaz komunikacije s fokusom na agenta Deltu

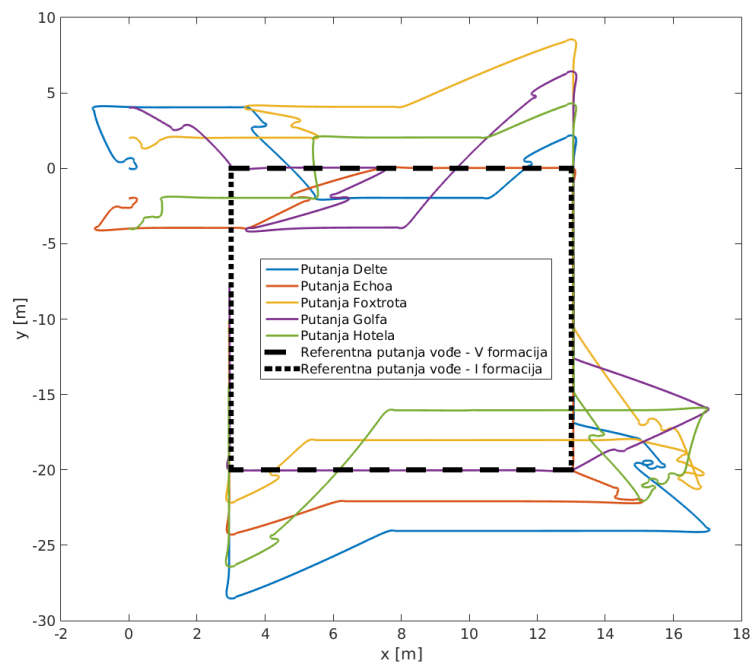
U priloženom filmskom isječku *patrola\_simulacija.mp4* prikazano je patroliranje robota uz dvije promjene vektora mjerenja  $\delta$  nakon jedne i sedam minuta simulacije. Prva se promjena vektora mjerenja  $\delta$  dogodi u  $\mathbf{V}$  formaciji, dok se druga promjena vektora mjerenja  $\delta$  dogodi usred promjene željene formacije iz  $\mathbf{I}$  u  $\mathbf{V}$ .

Prikaz patroliranja robota je dan na Slici 4.8. Početni položaj robota je:

$$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.25)$$

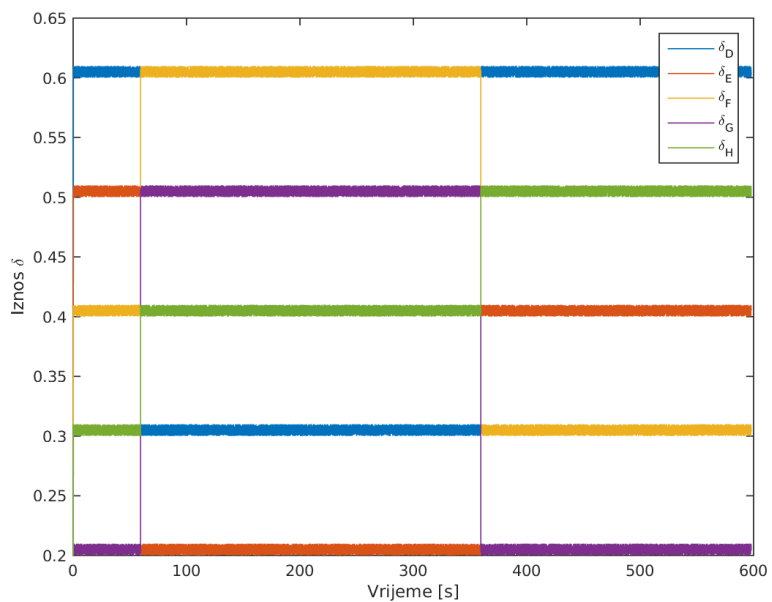
$$\mathbf{y}_0 = \begin{bmatrix} 0 & -2 & 2 & 4 & -4 \end{bmatrix} \quad (4.26)$$

Početni uvjeti vektora povjerenja  $\zeta$ , vektora pouzdanja  $\sigma$  i vektora mjerenja  $\delta$  su postavljeni kao u drugom poglavlju.



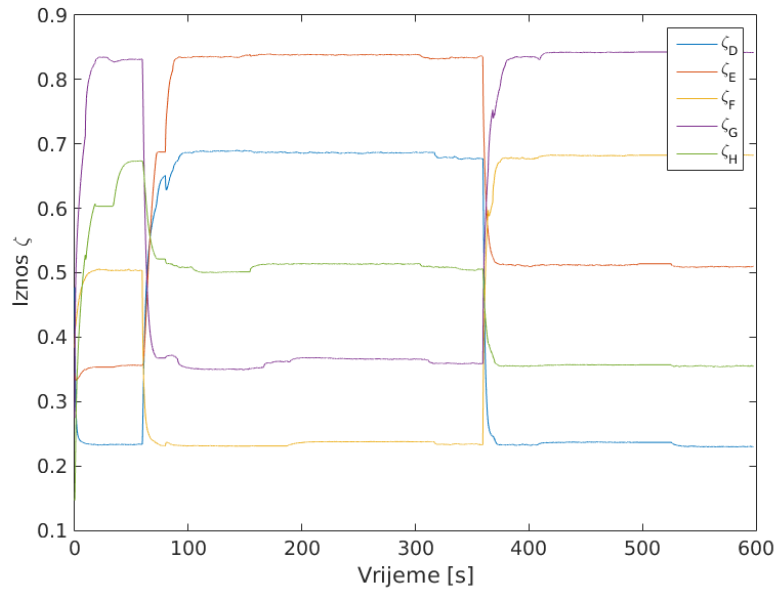
**Slika 4.8:** Prikaz putanja robota prilikom patroliranja

Na Slici 4.9 mogu se vidjeti kretanja vrijednosti stupaca vektora  $\delta$ .



**Slika 4.9:** Vrijednosti mjerenja se permutiraju dva puta. Jednom kad su roboti u **V** formaciji, a drugi put usred promjene iz **I** u **V** formaciju

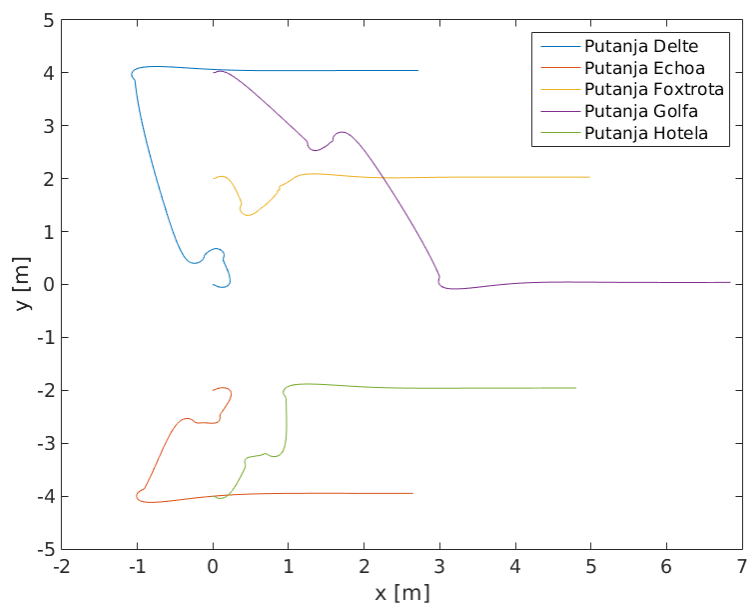
Prikaz iznosa stupaca vektora povjerenja  $\zeta$  dan je na Slici 4.10. Može se uočiti kako algoritam konsenzusa konvergira, a poredak agenata po povjerenju odgovara očekivanju s obzirom na mjerenja.



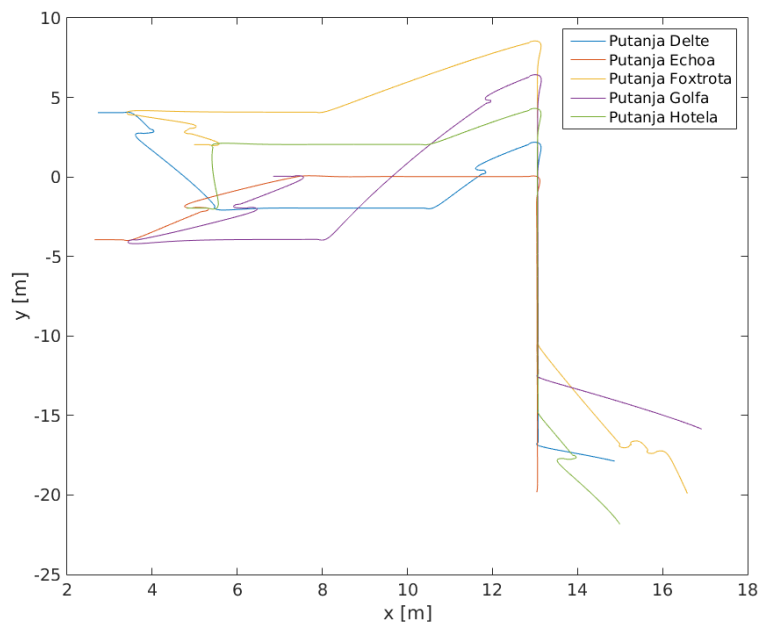
**Slika 4.10:** Vrijednosti povjerenja. Može se uočiti kako i u stacionarnom stanju postoje određeni pomaci zbog promjene vektora pouzdanja  $\sigma$  što ne utječe na poredak agenata po povjerenju

Primjetne su male promjene u stacionarnom stanju vektora povjerenja  $\zeta$  koje se mogu objasniti promjenom matrice susjedstva  $\mathbf{A}$ . Prilikom skretanja i promjene oblika formacije dolazi do nagle promjene u udaljenosti među robotima i samim time matrici susjedstva što za posljedicu ima promjene u matrici pouzdanja  $\sigma$ . Naime, po (2.4), agent  $i$  mijenja samo pouzdanje u svoje susjede koji mu promjenom formacije bivaju zamijenjeni novim agentima prema kojima pouzdanje nije mijenjao. Ta činjenica za posljedicu ima da se u njih agent  $i$  inicijalno pouzda previše ili premalo pa je potrebno neko vrijeme prije nego se vrijednost pouzdanja u njih adaptira.

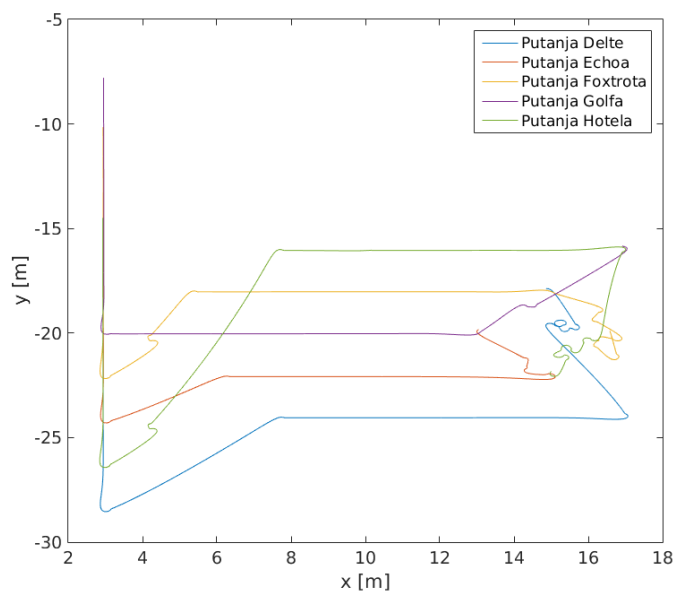
Prikaz putanja robota na Slici 4.8 je zbog preglednosti razložen na tri dijela u kojima se poredak robota unutar formacije ne mijenja, što se vidi na Slikama 4.11-4.13.



**Slika 4.11:** Prikaz prve dionice patroliranja robota uz agenta Hotela na čelu

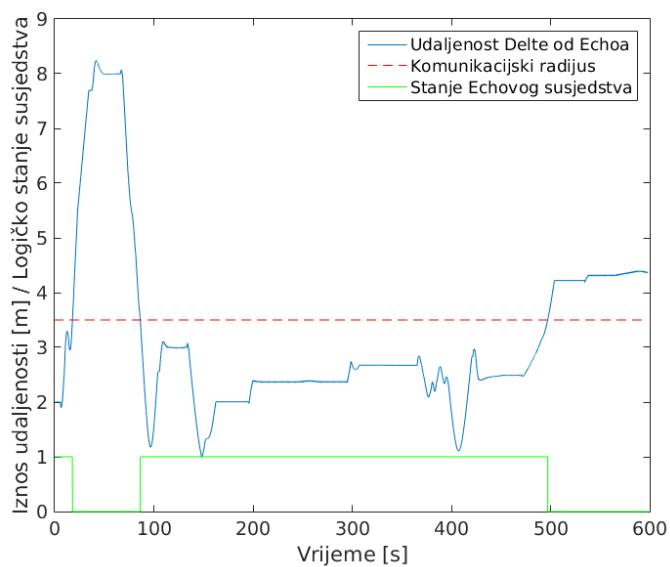


**Slika 4.12:** Prikaz druge dionice patroliranja robota uz agenta Echoa na čelu

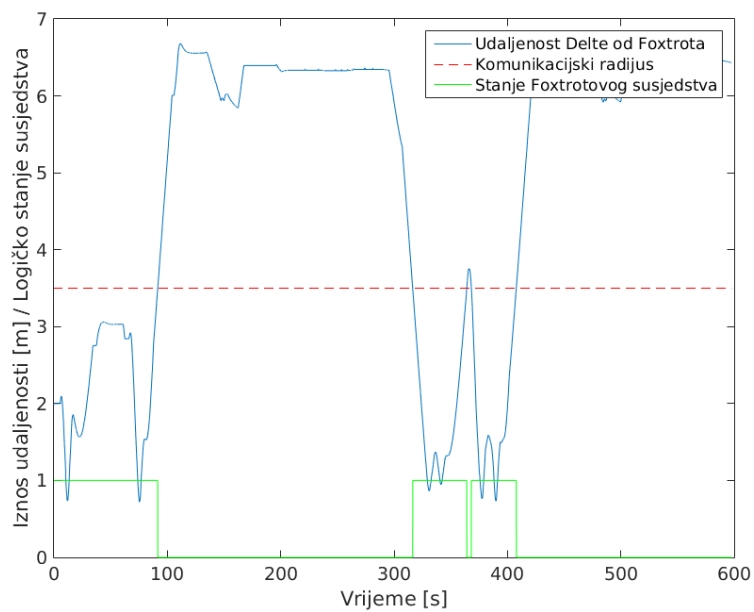


Slika 4.13: Prikaz treće dionice patroliranja robota uz agenta Hotela na čelu

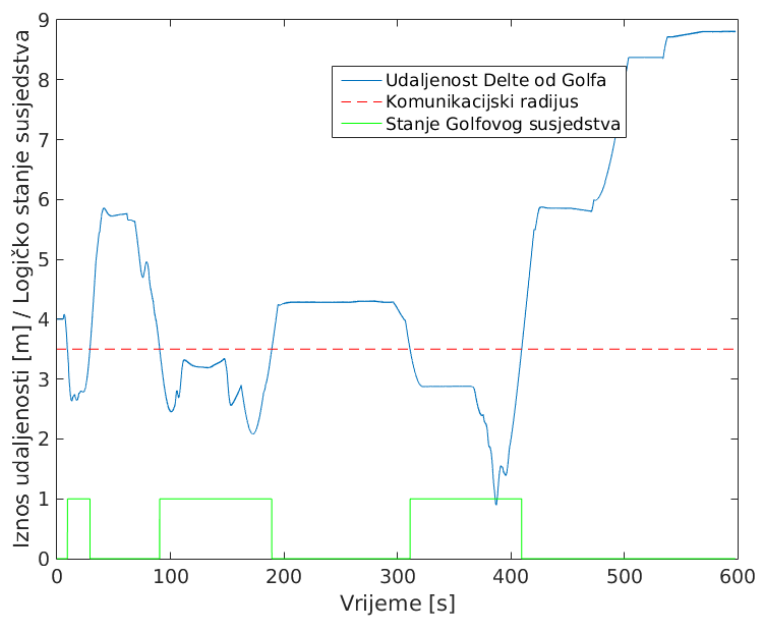
Matrica susjedstva  $\mathbf{A}$  se mijenja u vremenu. Radijus komunikacije  $r$  je postavljen na 3.5 m. Kao što je objašnjeno u prethodnom poglavlju, ako je udaljenost među robotima  $i$  i  $j$  manja od  $r$  vrijednosti  $A(i, j)$  i  $A(j, i)$  su 1, inače 0. Na Slikama 4.14-4.17 prikazana promjena vrijednosti prvog retka matrice  $\mathbf{A}$  u ovisnosti o udaljenosti među agentima.



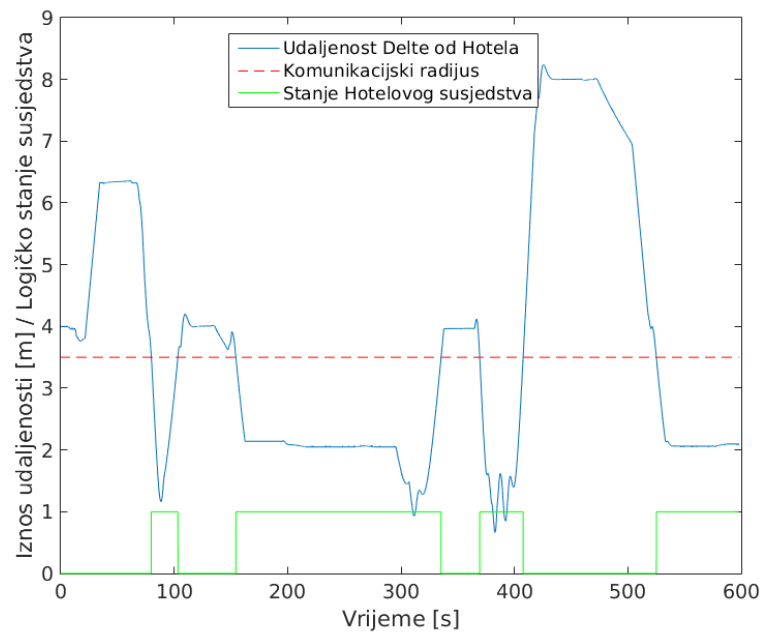
Slika 4.14: Promjena elementa  $A(1,2)$  u ovisnosti o udaljenosti među agentima Delta i Echo



**Slika 4.15:** Promjena elementa A(1,3) u ovisnosti o udaljenosti među agentima Delta i Foxtrot



**Slika 4.16:** Promjena elementa A(1,4) u ovisnosti o udaljenosti među agentima Delta i Golf



**Slika 4.17:** Promjena elementa  $A(1,5)$  u ovisnosti o udaljenosti među agentima Delta i Hotel

Primjetni su trenutci najveće promjene prvog retka matrice susjedstva  $A$  na oko 100, 320, 410 i 520 sekundi. Kao što je ranije objašnjeno, upravo su to trenutci kad su prisutne promjene u iznosu vektora povjerenja  $\zeta$  na Slici 4.10.



# 5. Decentralizirano upravljanje formacijom u laboratoriju

## 5.1. Promjena formacije u laboratoriju

U ovome je potpoglavlju na jednom primjeru razrađena i objašnjena statička promjena formacije multi-agentskog sustava u laboratorijskom okruženju kako bi se bolje razjasnilo upravljanje formacijom opisano u 3. poglavlju te kako bi se razmotrilo ponašanje realnog sustava i potencijalni implementacijski problemi. Radi se o tri robota (agenta) koji iz jedne trokutaste formacije međusobnom zamjenom pozicija prelaze u drugu trokutastu ( $\mathbf{V}$ ) formaciju. U početnoj je formaciji vođa agent Delta, njemu s desne strane agent Foxtrot, a s lijeve strane agent Echo. Tražena je formacija koja je zakrenuta za  $180^\circ$  u odnosu na početnu, a u kojoj raspored agenata ostaje isti. Početni je položaj robota unutar trokutaste formacije dan vektorima  $x_0$  i  $y_0$ , a završni je položaj robota dan vektorima  $x_c$  i  $y_c$ .

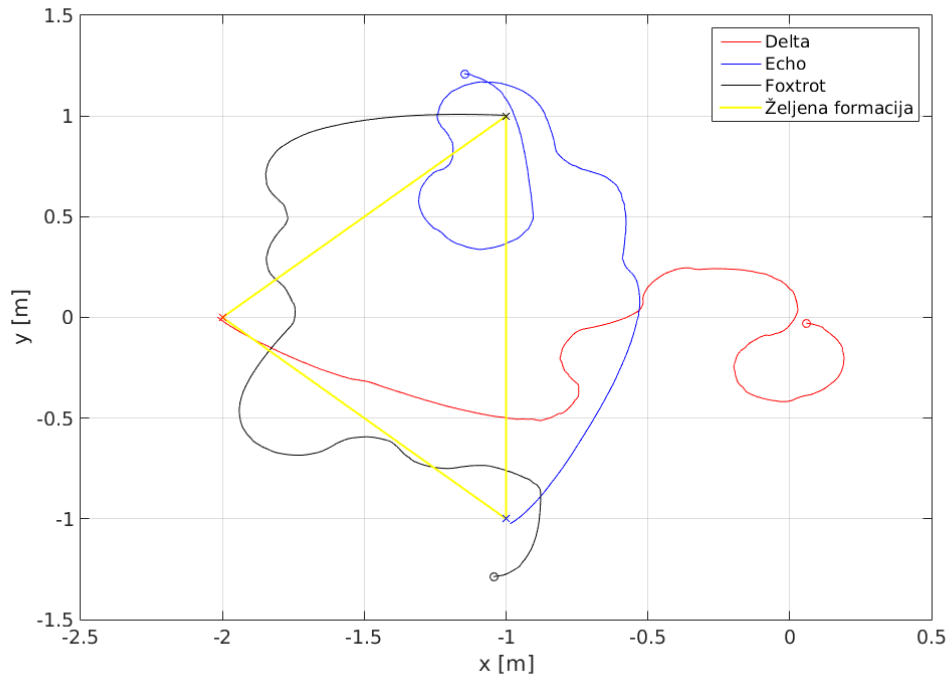
$$\mathbf{x}_0 = \begin{bmatrix} x_D & x_E & x_F \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 \end{bmatrix} \quad (5.1)$$

$$\mathbf{y}_0 = \begin{bmatrix} y_D & y_E & y_F \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \quad (5.2)$$

$$\mathbf{x}_c = \begin{bmatrix} 0 & -1 & -1 \end{bmatrix} \quad (5.3)$$

$$\mathbf{y}_c = \begin{bmatrix} -2 & -1 & 1 \end{bmatrix} \quad (5.4)$$

Putanje robota, zajedno s početnim i krajnjim pozicijama robota, mogu se vidjeti na Slici 5.1. Početne pozicije robota na Slici označene su simbolom  $\mathbf{o}$ , a završne pozicije označene su simbolom  $\mathbf{x}$ .



**Slika 5.1:** Željena formacija i stvarne putanje robota prilikom promjene formacije

Agenti se, zbog ograničenog prostora u laboratoriju, u početnim pozicijama nalaze na međusobnoj udaljenosti  $d = \sqrt{2}[m]$ . Ovo je dvostruko manje nego udaljenost agenata u početnim pozicijama prilikom simulacije promjene formacije (poglavlje 4.1). Iz tog je razloga bilo očekivano da će roboti češće međusobno doći unutar udaljenosti  $d_{min} = 1[m]$  kada jedan drugoga vide kao prepreku. Ovaj je pokus u laboratoriju dao dobar uvid u kvalitetu projektiranog sustava upravljanja brzinom robota koristeći umjetno generirana polja potencijala; i na relativno malim udaljenostima roboti su konvergirali svojim ciljevima sigurno i bez kolizija.

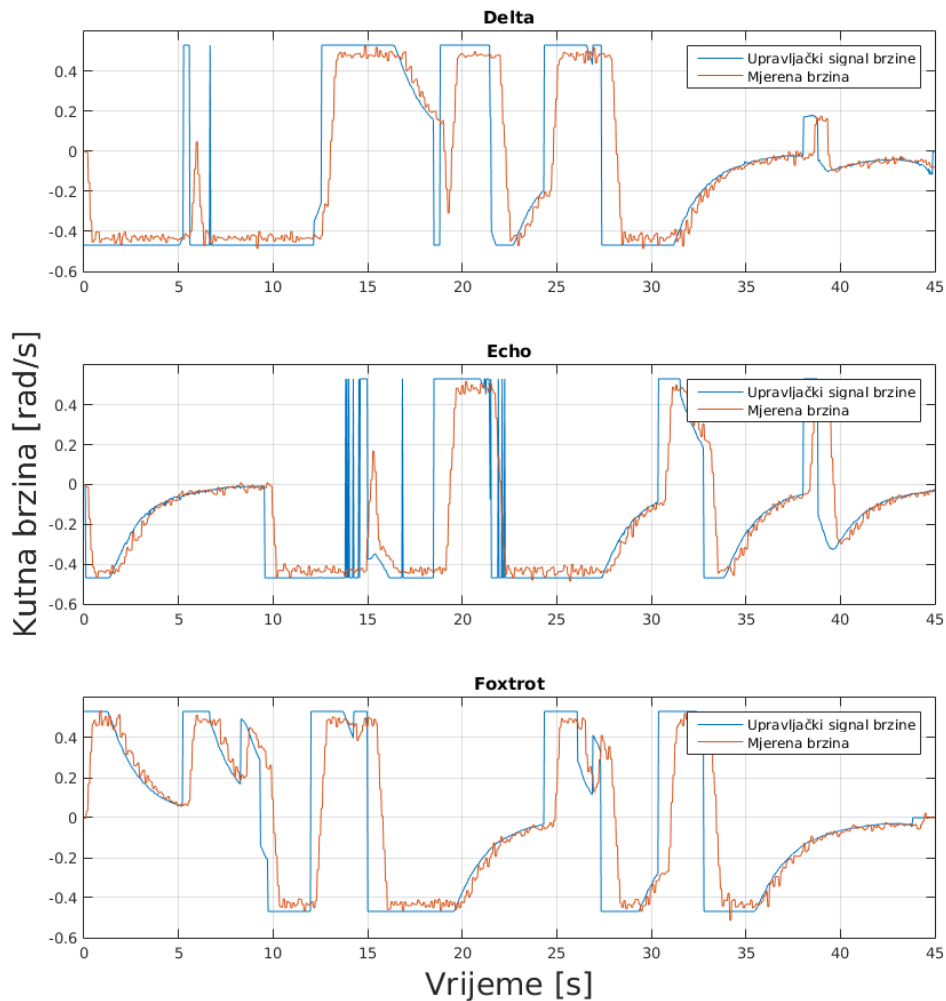
Analizu gibanja agenata pri promjeni formacije napraviti ćemo na putanji vođe, agenta Delte. U početnom je trenutku agent Delta okrenut od svoga cilja. Dobiva referencu u smjeru cilja te se počinje okretati, što se može zamijetiti kao polukružno skretanje na Slici 5.1. To se događa zato što robot ima dinamička ograničenja, odnosno ne može trenutno postići traženi kut. Dok se okreće dobije referencu linearne brzine

različitu od nule te se stoga ne okreće u mjestu već radi kratko polukružno gibanje. Takvo je ponašanje očekivano u radu sa stvarnim robotima budući da oni nisu točkasti, svesmjerni, i nije značajnije utjecalo na ponašanje cijelog multi-agentskog sustava. Pri kraju te polukružne putanje, kada je već gotovo okrenut prema svom cilju, agent Delta ulazi u polje agenta Foxtrota kojeg vidi kao prepreku. Budući da mu je prepreka bliže od cilja, agent Delta iz gradijenta polja potencijala dobiva referencu kuta koja ga jako tjera od prepreke. Radi naglo skretanje od prepreke sve dok ne bude na dozvoljenoj udaljenosti ( $1[m]$ ). Kada izađe iz polja prepreke okreće se prema svome cilju te kreće prema njemu. Ponovo dolazi u polje prepreke, sada agenta Echoa te se postupak ponavlja; skreće od prepreke te se, kada dođe na dozvoljenu udaljenost, ponovo okreće prema svome cilju. Nakon toga još dva puta dolazi pa izlazi iz polja prepreke te se naposljetku okreće cilju i sve do dolaska u cilj giba usmjereno, gotovo po pravcu. Već je pri analizi gibanja jednog agenta vidljiva razlika između gibanja u laboratoriju, gdje je prostor ograničen i tokom gibanja je više puta došlo do ulaska u polje prepreke, te gibanja u simulaciji gdje je prostor bio veći pa je gibanje bilo usmjerenije prema cilju.

Svaki od tri robota ostala dva robota vidi kao prepreke. Budući da se polja potencijala svakog robota generiraju na jednak način, ulazak jednog robota u radijus detekcije, odnosno u polje drugog robota znači da je i drugi robot ušao u polje prvog robota. Dakle, sukladno teorijskim razmatranjima u 3. poglavlju te jednako kao pri promjeni formacije u simulacijskom okruženju u potpoglavlju 4.1, roboti se međusobno jednako odbijaju, odnosno analogno 3. Newtonovom zakonu akcija uzrokuje reakciju. To se jasno vidi po putanjama prikazanim na Slici 5.1 i objašnjava putanje preostala dva robota, agenta Foxtrota i Echoa, koji se gibaju sukladno već objašnjenim obrascima ponašanja; agent se počne gibati usmjereno prema cilju, kada mu u polje uđe prepreka on se naglim skretanjem nastoji što prije maknuti od nje, zatim se ponovo usmjerava prema cilju i ponavlja to sve dok ne dođe u zadani cilj.

Na Slici 5.2 prikazani su upravljački signali kutne brzine svih agenata te mjerene izlazne brzine. Vidi se kako na početku gibanja svi roboti imaju neku referentnu kutnu brzinu; nastoje se okrenuti prema cilju. Zatim kutna brzina iščezava kako agenti idu prema cilju. Tek ako se u polju robota pojavi prepreka dolazi do promjene kutne brzine, ta je promjena brzine skokovita zato što je korišten P tip regulatora, a pri ulasku

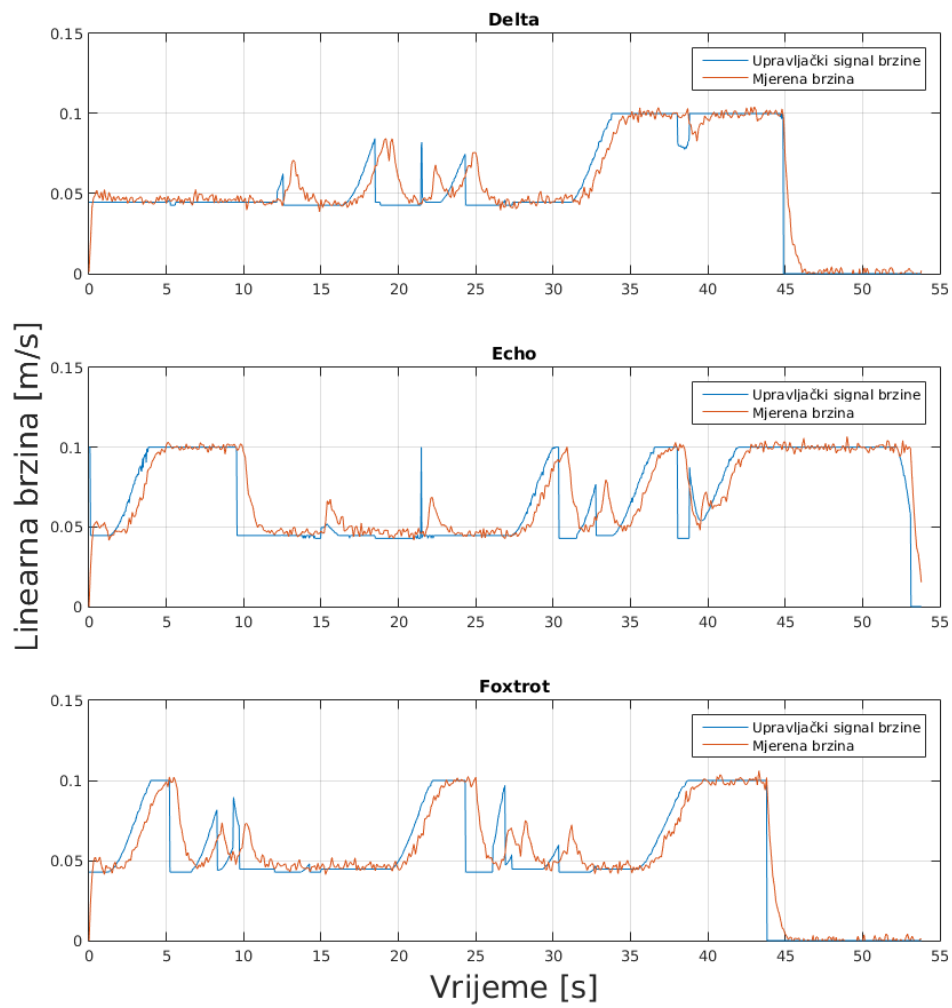
prepreke u radijus detekcije dogodi se nagli skok u polju, a time i nagla, skokovita promjena reference. Mjereni signal, očekivano, ima zamjetno kašnjenje (reda veličine sekunde), uzrokovano kašnjenjem Optitrack sustava, kašnjenjem u komunikaciji računala robota te dinamičkim ograničenjima stvarnih robota. Dinamika robota očituje se i pri praćenju upravljačkog signala; robot ne može trenutno postići zadanu kutnu brzinu već je dostiže kroz neko vrijeme. Ovo je zamjetno pri svakoj promjeni (skoku) upravljačkog signala, a posebno se dobro vidi u početnom trenutku gdje je potrebno neko vrijeme da bi robot ubrzao od nule prema referentnoj kutnoj brzini. Mjereni signal iznosom i oblikom dobro prati upravljački signal. Osim kašnjenja prisutan je i šum koji je posljedica nesavršenosti senzora i aktuatora.



**Slika 5.2:** Kutne brzine svih robota prilikom promjene formacije

Na Slici 5.3 prikazani su upravljački signali linearne brzine svih agenata te mjerene izlazne brzine. Sa slike se mogu vidjeti trenuci u kojima gibanje prestaje; kada robot dostigne svoj cilj referenca linearne brzine pada na nulu. Zbog dinamičkih ograničenja robota mjerena brzina ne pada trenutno već kroz određeno vrijeme usporavanja, reda veličine sekunde. Vidljivo je i da se robot, kada je gibanje usmjereno prema cilju, odnosno kada su prepreke izvan radijusa detekcije, giba konstantnom linearnom brzinom. Roboti su, očekivano, imali stacionarnu točnost linearne brzine tj. pojačanje 1. Sa Slike 5.3 se vidi da mjereni signal iznosom i oblikom dobro prati upravljački signal. Mjereni signal, očekivano, ima zamjetno kašnjenje (reda veličine sekunde), uzrokovano kašnjenjem Optitrack sustava, kašnjenjem u komunikaciji računala robota te dinamičkim ograničenjima stvarnih robota. Zamjetan je i određeni, po amplitudi neznemariv šum kao posljedica nesavršenosti senzora i aktuatora. Promatrajući Sliku 5.3 i Sliku 5.2 može se uočiti kako se linearna brzina zaista ponaša prema upravljačkom zakonu (3.7), sukladno teorijskim razmatranjima u 3. poglavlju.

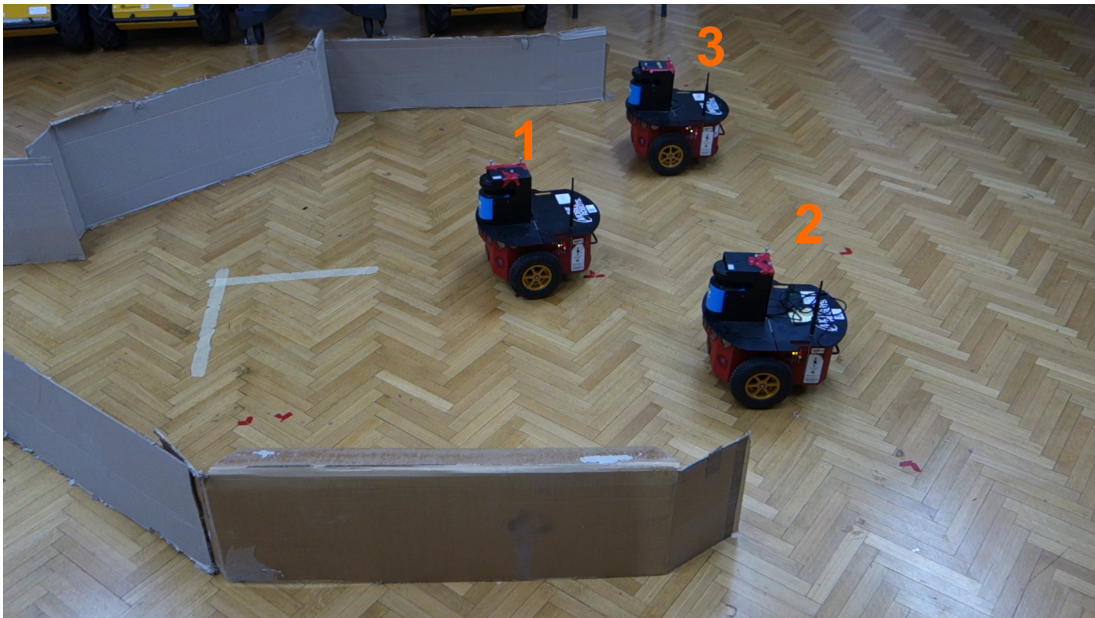
Snimka eksperimenta opisanog u ovom poglavlju nalazi se u priloženom filmskom isječku *zamjena\_laboratorij.mp4*.



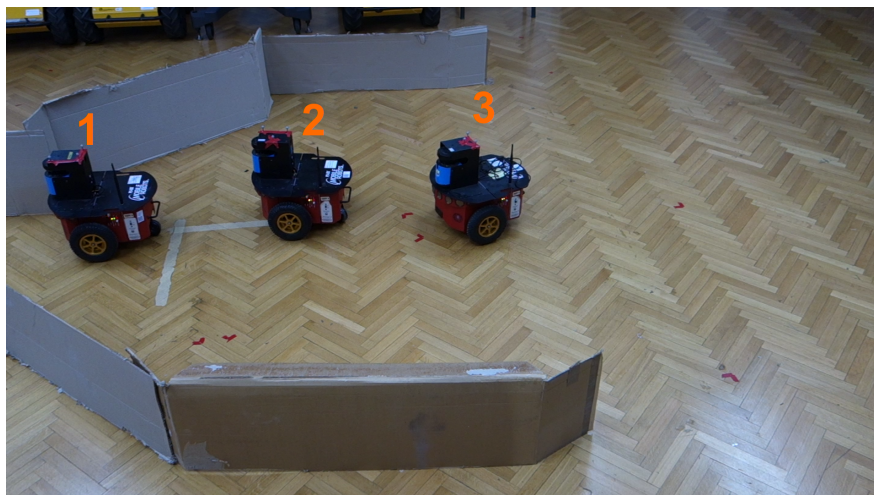
Slika 5.3: Linearne brzine svih robota prilikom promjene formacije

## 5.2. Patroliranje u laboratoriju

Patroliranje u laboratoriju izvedeno je s tri robota. Upravljački algoritam je istovjetan kao i prilikom patroliranja u Gazebo simulatoru. Regulacija linearne i kutne brzine je istovjetna kao i u prethodnom poglavlju uz smanjen radijus detekcije robota na  $r = 0.5m$  i smanjenje maksimalne linearne brzine robota na  $v_{max} = 0.1m/s$ . To je učinjeno kako bi se povećala duljina patroliranja vođe u laboratoriju, a maksimalna brzina je smanjena zbog osiguranja od sudara. Za razliku od patroliranja u simulatoru, u laboratoriju je referentna putanja vođe ravna linija duga 4 metra. Početna formacija robota je **V** koja se može vidjeti na Slici 5.4, a zbog prepreka ju je nužno promijeniti u **I** kao na Slici 5.5.



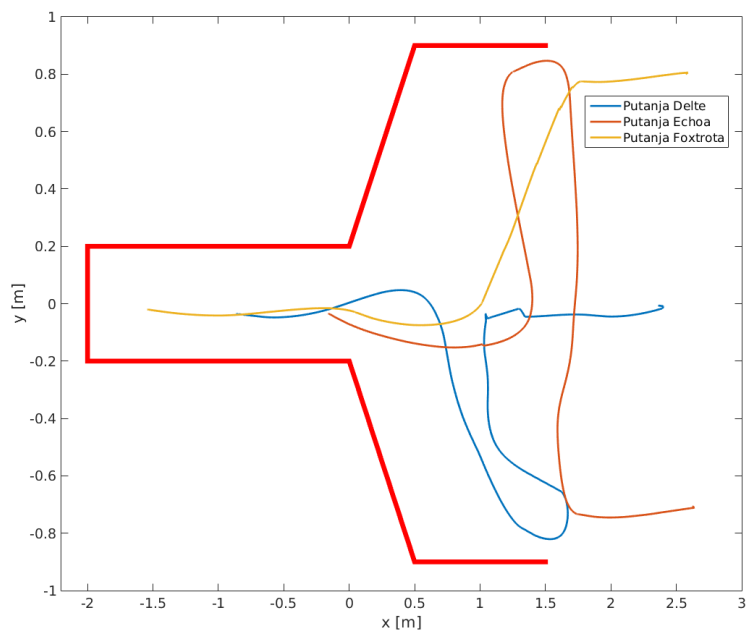
**Slika 5.4:** Agenti u **V** formaciji s označenim redoslijedom po povjerenju



**Slika 5.5:** Agenti u **I** formaciji s označenim redoslijedom po povjerenju

Također, agenti se u formaciji u laboratoriju nalaze bliže nego u simulatoru, jer je apsolutna udaljenost vođe od ostalih agenata u **V** formaciji smanjena s 4 na 1 metar, a udaljenost između susjeda u **I** formaciji s 1.5 na 0.7 metara.

Prikaz patroliranja dan je na filmskom isječku *patrola\_laboratorij.mp4*, a putanje su prikazane na Slici 5.6.



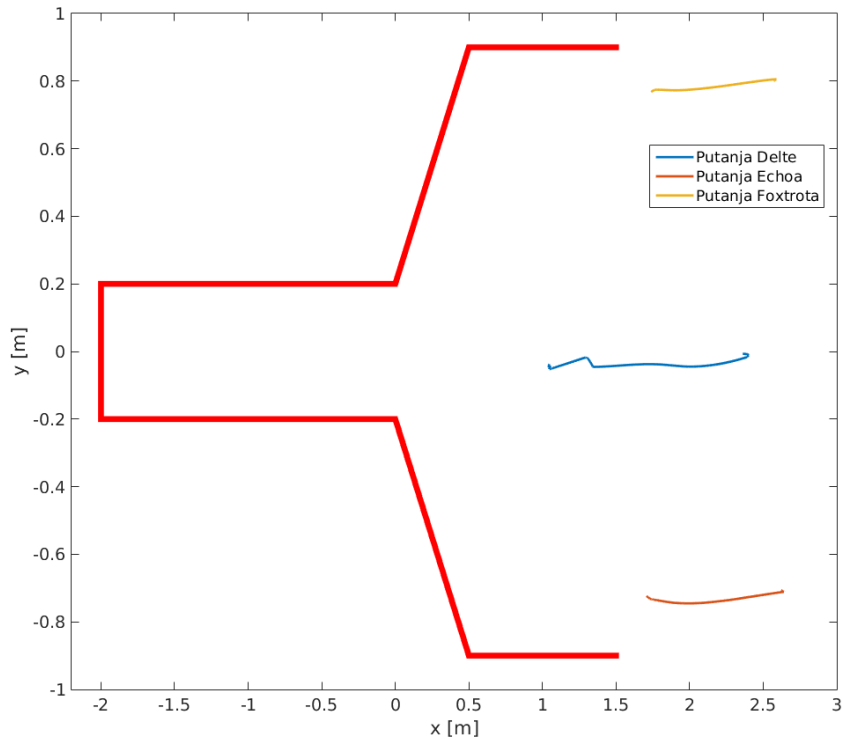
**Slika 5.6:** Prikaz putanja robota prilikom patroliranja



Crvenom bojom je označena kartonska prepreka. Početan položaj robota je dan s:

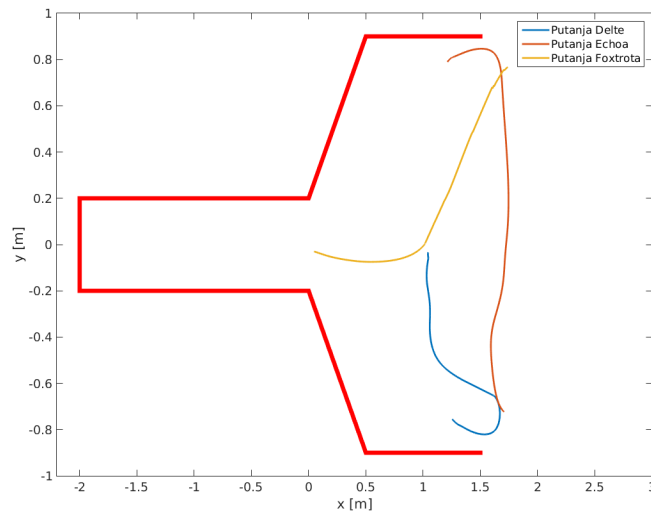
$$\mathbf{x}_0 \approx \begin{bmatrix} 2.6 & 2.6 & 2.6 \end{bmatrix} \quad (5.5)$$

$$\mathbf{y}_0 \approx \begin{bmatrix} 0 & -0.7 & 0.7 \end{bmatrix} \quad (5.6)$$



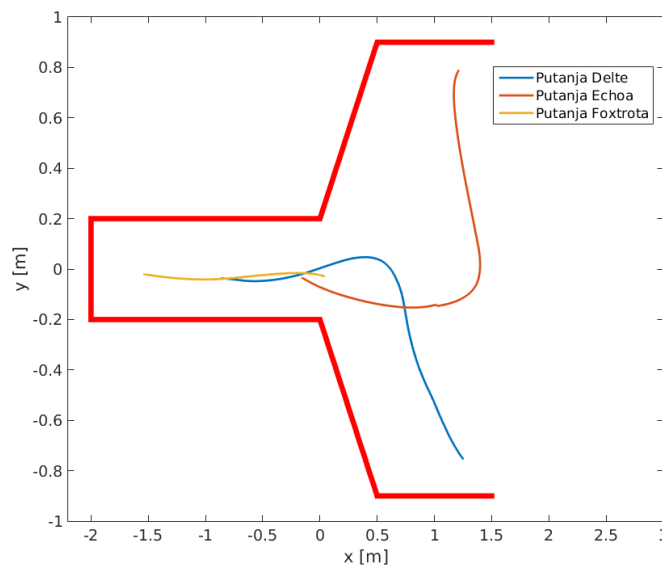
**Slika 5.7:** Prikaz prve dionice patroliranja robota uz agenta Delta na čelu u **V** formaciji

U početnoj poziciji su svi agenti poredani u liniju. Nakon postizanja konsensusa, agent s najvećim povjerenjem, Delta, zauzima vodeći položaj. Kad ga zauzme, referenca položaja vođe se počne mijenjati i svi se roboti gibaju u **V** formaciji.



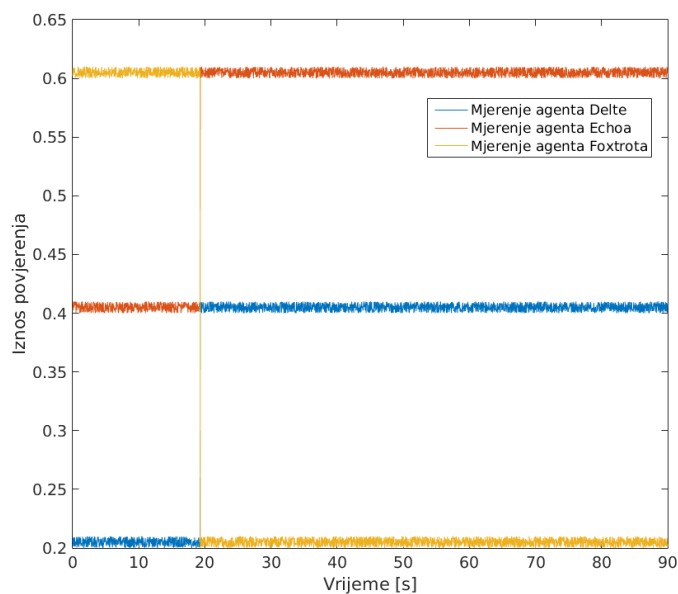
**Slika 5.8:** Prikaz druge dionice patroliranja robota uz agenta Foxtrota na čelu u **V** formaciji

Kad se postigne konsenzus oko drukčijih vrijednosti povjerenja, Foxtrot odlazi na čelo i formacija se nastavlja gibati.



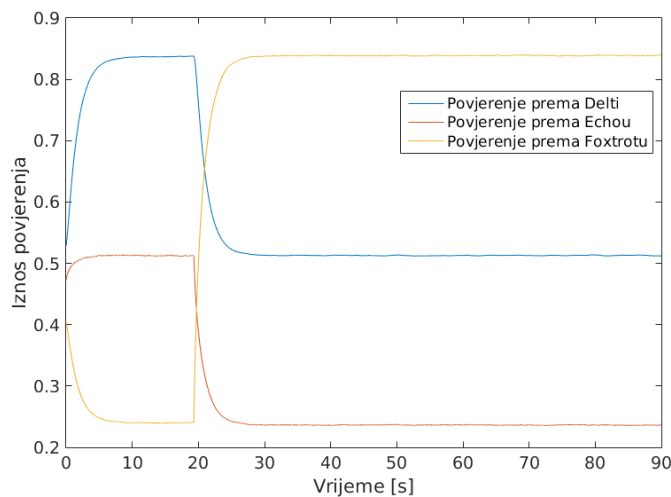
**Slika 5.9:** Prikaz treće dionice patroliranja robota uz agenta Foxtrota na čelu u **I** formaciji

Kad širina prepreke postane premala za gibanje u **V** formaciji roboti se, na čelu s Foxtrotom, preslažu u **I** formaciju i nastavljaju gibanje do cilja. Mjerenja agenata se vide na Slici 5.10. Vrijednosti vektora mjerenja  $\zeta$  se permutiraju samo jednom na 20 sekundi jer je zamjena agenata nemoguća kad se prepreka počne sužavati.



**Slika 5.10:** Vrijednosti mjerenja agenata

Prikaz iznosa stupaca vektora povjerenja  $\zeta$  je dan na Slici 5.11. Može se uočiti kako algoritam konsenzusa konvergira, a poredak agenata po povjerenju odgovara očekivanju s obzirom na mjerenja.



**Slika 5.11:** Vrijednosti povjerenja

Odzivi povjerenja u laboratoriju su sličniji odzivima koji su dobiveni u drugom poglavlju, gdje su se razmatrali algoritmi konsenzusa, nego u simulacijskom okruženju Gazebo. Prijelazna pojava je glađa i nema turbulencija u stacionarnom stanju. Razlog

tome je taj da je u laboratoriju matrica susjedstva  $\mathbf{A}$  konstantna i iznosi:

$$\mathbf{A}(t) = \mathbf{A}(0) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (5.7)$$

Odnosno, s obzirom da vođa u svakom trenutku komunicira sa svojim prvim susjedima, kao i oni međusobno, svaki robot je susjed sa svim ostalim robotima. Zbog toga se cijela matrica pouzdanja  $\sigma$  mijenja u svakom trenutku (osim na glavnoj dijagonali jer se samopouzdanje po (2.4) ne mijenja) pa nema naglih promjena u vektoru povjerenja  $\zeta$ .

## 6. Zaključak

U ovom radu je implementirano decentralizirano upravljanje multiagentskim robotskim sustavom zasnovano na algoritmima povjerenja u simulacijskom i laboratorijskom okruženju. S obzirom na činjenicu da je upravljanje multiagentskim sustavom decentralizirano, odnosno pojedini robot nema nužno informaciju o položaju svih ostalih robota, odabran je sustav upravljanja pomoću gradijenta zvonolikih funkcija potencijala.

Gibanjem robota upravlja se linearnim P regulatorom kutne brzine i nelinearnim regulatorom linijske brzine čiji je referentni signal generiran iz gradijenta funkcije potencijala. Parametri regulatora odabrani su tako da se garantira konvergencija formacije i odsustvo sudara robota. Raspored robota unutar formacije određen je vektorom povjerenja oko kojeg je postignut konsenzus. Povjerenje je mjera u kojoj agent  $i$  vjeruje agentu  $j$  da želi obaviti zadatak  $a$ . U kontekstu ovog rada, posao je robotsko patroliranje. Vrijednosti povjerenja se stalno osvježavaju u ovisnosti o mjerenjima koji predstavljaju mjeru sposobnosti agenata.

Iznesena teorijska razmatranja potvrđena su eksperimentima patroliranja robota u I i V formaciji u simulacijskom okruženju i u laboratoriju. Treba naglasiti kako su korišteni Pioneer P3-DX roboti koji imaju diferencijalan pogon i zbog toga nisu svesmjerni kao roboti korišteni u publikacijama koje su motivirali ovaj rad [18].

## 7. Literatura

- [1] Erico Guizzo. How google's self-driving car works. *IEEE Spectrum Online*, October, 18, 2011.
- [2] Sergio Maeso, Mercedes Reza, Julio A Mayol, Juan A Blasco, Mercedes Guerra, Elena Andradas, and María N Plana. Efficacy of the da vinci surgical system in abdominal surgery compared with that of laparoscopy: a systematic review and meta-analysis. *Annals of surgery*, 252(2):254–262, 2010.
- [3] Karen Cook. *Trust in society*. Russell Sage Foundation, 2001.
- [4] Tomislav Haus, Ivana Palunko, Domagoj Tolic, Stjepan Bogdan, Frank L Lewis, and Dariusz G Mikulski. Trust-based self-organising network control. *Control Theory & Applications, IET*, 8(18):2126–2135, 2014.
- [5] Kevin T Mantel and Christopher M Clark. Trust networks in multi-robot communities. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 2114–2119. IEEE, 2012.
- [6] Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [7] Tomislav Haus, Ivana Palunko, Domagoj Tolić, Stjepan Bogdan, and Frank L Lewis. Decentralized trust-based self-organizing cooperative control. In *Control Conference (ECC), 2014 European*, pages 1205–1210. IEEE, 2014.
- [8] Ivana Jerončić. Evaluacija multi-agentskih sustava temeljenih na povjerenju. Master's thesis, FER, 2015.

- [9] NS Zelinska. International significance of icao alphabet for flight safety. 2013.
- [10] Alejandro G Yepes, Francisco D Freijedo, Jesus Doval-Gandoy, Oscar Lopez, Jano Malvar, and Pablo Fernandez-Comesa. Effects of discretization methods on the performance of resonant controllers. *Power Electronics, IEEE Transactions on*, 25(7):1692–1712, 2010.
- [11] Tomislav Burić, Luka Korkut, Mario Krnić, Josipa Pina Milišić, Mervan Pašić, and Igor Velčić. *Vektorska analiza*. Element, 2014.
- [12] Kristian Hengster-Movrić, Stjepan Bogdan, and Ivica Draganjac. Multi-agent formation control based on bell-shaped potential functions. *Journal of Intelligent and Robotic Systems*, 58(2):165–189, 2010.
- [13] Yong K Hwang and Narendra Ahuja. A potential field approach to path planning. *Robotics and Automation, IEEE Transactions on*, 8(1):23–32, 1992.
- [14] Shuzhi S. Ge and Yun J Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3):207–222, 2002.
- [15] Zdenko Kovačić, Stjepan Bogdan, and Vesna Krajči. *Osnove robotike*. Graphis, 2002.
- [16] *Pioneer 3 Operations Manual*, 2006.
- [17] Zoran Vukić and Ljubomir Kuljača. *Automatsko upravljanje - analiza linearnih sustava*. Kigen, 2005.
- [18] Charles Pippin and Helen Christensen. Trust modeling in multi-robot patrolling. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 59–66. IEEE, 2014.

## 8. Sažetak

U radu je implementirano decentralizirano upravljanje multiagentskim robotskim sustavom zasnovano na algoritmima povjerenja u simulacijskom i laboratorijskom okruženju. Korišteni su roboti Pioneer P3-DX. Raspored robota unutar formacije određen je vektorom povjerenja oko kojeg je postignut konsenzus, a gibanje robota upravljano je sustavom upravljanja pomoću gradijenta zvonolikih funkcija potencijala. Provedeni su eksperimenti u simulacijskom okruženju i u laboratoriju.

**Ključne riječi:** multi-agentski sustavi, algoritmi konsenzusa, decentralizirano upravljanje, upravljanje formacijom, sustavi temeljeni na povjerenju



## 9. Summary

In this work, the decentralized trust-based control algorithm of a multi-agent robotic system, is given. Pioneer P3-DX robots have been used. Arrangement of robots inside of a formation has been determined using trust vector for which all the agents have reached consensus. Movement of the robots has been controlled by control system with an artificial potential based on bell-shaped functions. Finally, multiple experiments have been done in the simulation environment and in the laboratory.

**Keywords:** multi-agent system, consensus algorithm, decentralized control, formation control, trust-based systems