

# Upravljanje mobilnim robotima u formaciji

---

**Pažanin, Ivan**

**Master's thesis / Diplomski rad**

**2017**

*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:235:022601>

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

*Download date / Datum preuzimanja:* **2024-07-14**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Ivan Pažanin**

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

Izv. prof. dr. sc. Andrej Jokić, dipl. ing.

Student:

Ivan Pažanin

Zagreb, 2017.

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

Zahvaljujem mentoru izv. prof. dr. sc. Andreju Jokiću, dipl. ing. na pomoći i savjetima tijekom pisanja ovog diplomskog rada te Mihaelu Lobroviću, mag. ing. na svojoj pomoći pri izvedbi simulacija. Zahvaljujem i Šimunu Čaglju, prof. na lekturi i jezičnim savjetima. Uz to, želim zahvaliti svojim roditeljima i bratu na potpori za vrijeme mog cjelokupnog studija. Na kraju, osobito zahvaljujem svojoj Mariji na njejoj pomoći i potpori za vrijeme pisanja ovog rada.

Ivan Pažanin



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



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

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

## DIPLOMSKI ZADATAK

Student: **Ivan Pažanin**

Mat. br.: 0035181472

Naslov rada na hrvatskom jeziku: **Upravljanje mobilnim robotima u formaciji**

Naslov rada na engleskom jeziku: **Control of mobile robots in formation**

Opis zadatka:

U posljednje vrijeme sve je više istraživanja na temu upravljanja velikog broja dinamičkih sustava koji su u međusobnoj interakciji i imaju zajednički (globalan) cilj, ali nemaju centralizirano upravljanje/koordinaciju. Umjesto centraliziranog upravljanja koristi se mnogo prikladnija distribuirana upravljačka struktura. Tipičan primjer ovakvog sustava je skup autonomnih mobilnih robota čiji je zajednički cilj gibanje u zadanoj formaciji. Distribuirana upravljačka struktura ovdje znači da svaki robot prima (putem direktnog mjerenja ili putem komunikacijskog kanala) samo informacije od sebi susjednih robota. Izazov je sintetizirati lokalne regulatore (regulatore na razini svakog pojedinačnog robota) i pravila njihove koordinacije tako da ovakva lokalizirana djelovanja nužno na zadovoljavajući način rješavaju globalan cilj.

U radu je potrebno:

- 1) Iz pregleda literature prikazati osnovne pristupe upravljanju grupom mobilnih robota.
- 2) Ponuditi rješenja za upravljanje formacijom u gibanju. Posebno prikazati moguća rješenja u situaciji kad svi roboti nisu nužno u kontaktu s referentnim robotom.
- 3) Na nizu prikladno osmišljenih simulacija ilustrirati efikasnost predstavljanog upravljačkog algoritma. Usporediti rezultate za različit broj mobilnih robota u formaciji kao i za različite topologije grafa komunikacije između robota.

Rad predati u pisanom i elektroničkom obliku s priloženim programskim kodom.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

11. svibnja 2017.

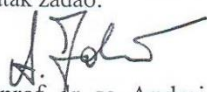
Rok predaje rada:

13. srpnja 2017.

Predviđeni datum obrane:

19., 20. i 21. srpnja 2017.

Zadatak zadao:

  
Izv. prof. dr. sc. Andrej Jokić

Predsjednica Povjerenstva:

  
Prof. dr. sc. Tanja Jurčević Lulić

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS OZNAKA .....	V
SAŽETAK.....	IX
SUMMARY .....	X
1. UVOD.....	1
1.1 Primjene formacija mobilnih robota (vozila).....	2
1.2 Pregled rada.....	5
2. KINEMATIKA MOBILNOG ROBOTA.....	6
2.1 Definiranje koordinatnih sustava i veza između njih.....	6
2.2 Kinematska ograničenja kotača.....	7
2.2.1 Standardni kotač.....	8
2.2.2 Kastorski kotač.....	9
2.3 Kinematika mobilnog robota – kinematički model gibanja.....	11
3. DINAMIKA MOBILNOG ROBOTA.....	14
4. UPRAVLJAČKI SUSTAV .....	17
4.1 Model s jednim integratorom (kinematika).....	17
4.2 Model s dva integratora (dinamika) .....	19
5. FORMACIJE MOBILNIH ROBOTA.....	21
5.1 Praćenje .....	21
5.1.1 Simulacija praćenja .....	22
5.2 Kružno praćenje .....	25
5.2.1 Simulacija kružnog praćenja .....	26
5.3 Susretanje (randezvous) .....	28
5.3.1 Simulacija susretanja.....	29
5.4 Uspostavljanje statičke formacije.....	31
5.4.1 Simulacije uspostavljanja formacije.....	32
5.5 Zaključci.....	33
6. GIBANJE U FORMACIJI.....	34
6.1 Linijsko gibanje u formaciji .....	34
6.1.1 Upravljački zakon .....	34
6.1.2 Kinematika linijskog gibanja u formaciji.....	36
6.1.3 Dinamika linijskog gibanja u formaciji.....	39
6.1.4 Komentar.....	44
6.2 Gibanje formacije po krivulji .....	47
6.2.1 Gibanje formacije po krivulji .....	47
6.2.2 Kinematika gibanja formacije po krivulji .....	50
6.2.3 Dinamika gibanja formacije po krivulji .....	54
6.2.4 Komentar.....	58
6.3 Kombinacija gibanja u formaciji.....	61

---

6.3.1. Upravljački zakon .....	61
6.3.2 Simulacije.....	65
6.3.2.1 Gibanje po liniji.....	65
6.3.2.2 Gibanje po krivulji .....	68
6.3.3. Komentar.....	71
7. ZAKLJUČAK.....	72
PRILOG A - UPUTE ZA KORIŠTENJE APLIKACIJE.....	75
PRILOG B - CD.....	90

## POPIS SLIKA

Slika 1. Formacije životinja; a) Ptice [1], b) Ribe [2] .....	1
Slika 2. Mravi ratnici [6] .....	2
Slika 3. MMS; ilustracija [8] .....	2
Slika 4. Praćenje širenja šumskog požara; a) lansiranje robota s bazne stanice, b) gibanje robota prije idealno uspostavljene formacije, c) i d) praćenje požara u idealnoj formaciji [7, 9] .....	3
Slika 5. Vožnja u koloni [10] .....	3
Slika 6. Formacija autonomnih brodova [11] .....	4
Slika 7. Autonomna žetva [12] .....	4
Slika 8. Globalni i lokalni koordinatni sustav [14] .....	6
Slika 9. Fiksni standardni kotač i njegovi parametri [14] .....	8
Slika 10. Kastorski kotač [13] .....	10
Slika 11. Parametri kotača [14] .....	11
Slika 12. Transformacija modela diferencijalnog pogona (a) u unicycle model (b) [14] .....	13
Slika 13. Ulazi i izlazi iz modela unicycle-a [14] .....	13
Slika 14. Brzina translacije težišta mobilnog robota [14] .....	14
Slika 15. Definiranje točke $P'$ [14] .....	17
Slika 16. Praćenje [14] .....	21
Slika 17. Praćenje usporedba – model točkastog robota .....	22
Slika 18. Usporedba različitih vrijednosti faktora $k$ .....	23
Slika 19. Usporedba rezultata praćenja za <i>unicycle</i> model .....	24
Slika 20. Kružno praćenje [14] .....	25
Slika 21. Simulacija kinematike kružnog praćenja .....	27
Slika 22. Simulacija dinamike kružnog praćenja .....	27
Slika 23. a) robot 1 cilja prema robotu 2; b) robot 1 cilja u svoje susjedstvo [17] .....	28
Slika 24. Susretanje kinematika .....	29
Slika 25. Susretanje dinamika .....	30
Slika 26. Simulacija uspostavljanja formacije (kinematika) .....	32
Slika 27. Simulacija uspostavljanja formacije (dinamika) .....	33
Slika 28. Graf komunikacije za slučaj s 9 robota i najmanje komunikacije .....	36
Slika 29. Graf komunikacije za slučaj s 9 robota i potpunom komunikacijom .....	36
Slika 30. Kinematika V9 formacije s najmanje komunikacije .....	38
Slika 31. Kinematika V9 formacije s najviše komunikacije .....	38
Slika 32. Graf komunikacije s 5 robota uz najmanje komunikacije .....	39
Slika 33. Graf komunikacije s 5 robota uz najviše komunikacije .....	39
Slika 34. Simulacija dinamike s 5 robota s najmanje komunikacije .....	40
Slika 35. Simulacija dinamike s 5 robota i najviše komunikacije .....	41
Slika 36. Simulacija dinamike s 9 robota i najmanje komunikacije .....	42
Slika 37. Simulacija dinamike s 9 robota i najviše komunikacije .....	42
Slika 38. Simulacija dinamike s 15 robota i najmanje komunikacije .....	43
Slika 39. Simulacija dinamike s 15 robota i najviše komunikacije .....	43
Slika 40. Graf komunikacije s 9 robota i malo većom razinom komunikacije .....	44
Slika 41. Poboljšanje očuvanosti formacije i konvergencije željenim krajnjim točkama .....	45
Slika 42. Dinamika s 9 robota u kojoj je važnije doći do željenih krajnjih pozicija .....	46
Slika 43. Simulacija s 5 robota i za upravljački zakon definiran jednačbom (106) .....	48
Slika 44. Simulacija s 9 robota i za upravljački zakon definiran jednačbom (106) .....	48



Slika 45. Odstupanje putanje robota od željene trajektorije upravljačkog zakona prikazanog jednadžbom (106).....	49
Slika 46. Kinematika gibanja formacije po krivulji, 5 robota i najmanje komunikacije .....	50
Slika 47. Kinematika gibanja formacije po krivulji, 9 robota i najmanje komunikacije .....	51
Slika 48. Kinematika gibanja formacije po krivulji, 9 robota i najmanje komunikacije, detalj prikaza formacije .....	51
Slika 49. Kinematika gibanja formacije po krivulji, 5 robota i najviše komunikacije.....	52
Slika 50. Kinematika gibanja formacije po krivulji, 9 robota i najviše komunikacije.....	52
Slika 51. Kinematika gibanja formacije po krivulji, 9 robota i najviše komunikacije, detalj prikaza formacije .....	53
Slika 52. Odstupanje putanje robota od željene trajektorije upravljačkog zakona prikazanog jednadžbom (108).....	53
Slika 53. Dinamika gibanja formacije po krivulji, 5 robota i najmanje komunikacije .....	55
Slika 54. Dinamika gibanja formacije po krivulji, 9 robota i najmanje komunikacije .....	55
Slika 55. Dinamika gibanja formacije po krivulji, 9 robota i najmanje komunikacije, detalj prikaza formacije .....	56
Slika 56. Dinamika gibanja formacije po krivulji, 5 robota i najviše komunikacije.....	56
Slika 57. Dinamika gibanja formacije po krivulji, 9 robota i najviše komunikacije.....	57
Slika 58. Dinamika gibanja formacije po krivulji, 9 robota i najviše komunikacije, detalj prikaza formacije .....	57
Slika 59. Kinematika gibanja formacije po krivulji, gibanje po kružnici .....	59
Slika 60. Dinamika gibanja formacije po krivulji, gibanje po kružnici .....	59
Slika 61. Kinematika gibanja po krivulji, gibanje po sinusoidi .....	60
Slika 62. Dinamika gibanja po krivulji, gibanje po sinusoidi .....	60
Slika 63. Definiranje koordinatnih sustava i parametara .....	61
Slika 64. Definiranje kuta $\gamma_2$ .....	62
Slika 65. Definiranje udaljenosti između trenutne i željene pozicije robota.....	63
Slika 66. Graf komunikacije s 5 robota gdje je robot 1 vođa.....	65
Slika 67. Graf komunikacije s 9 robota gdje je robot 1 vođa.....	66
Slika 68. Gibanje formacije s 5 robota po liniji, crveni robot je vodeći robot.....	66
Slika 69. Gibanje formacije s 9 robota po liniji, crveni robot je vodeći robot.....	67
Slika 70. Gibanje formacije s 5 robota po liniji, crveni robot je vodeći robot, kut $\gamma = \pi$ .....	67
Slika 71. Gibanje formacije s 5 robota po krivulji, crveni robot je vodeći .....	68
Slika 72. Gibanje formacije s 9 robota po krivulji, crveni robot je vodeći .....	69
Slika 73. Gibanje formacije s 9 robota po krivulji, crveni robot je vodeći, detaljni prikaz uspostavljene formacije .....	69
Slika 74. Gibanje formacije s 5 robota po krivulji, crveni robot je vodeći robot, kut $\gamma = \pi$ ...	70

## POPIS OZNAKA

Oznaka	Opis
$\vec{\Delta}_{12}$	vektor koji definira razliku između trenutne i željene pozicije robota 2
$\vec{d}_{12}$	vektor koji definira trenutnu udaljenost između robota 1 i 2
$\dot{q}_I$	vektor brzina robota u globalnom koordinatnom sustavu
$\dot{q}_R$	vektor brzina robota u lokalnom koordinatnom sustavu
$\dot{v}_i$	akceleracija robota $i$
$\dot{\tilde{x}}$	brzina točke $P'$ u smjeru osi $X_I$
$\ddot{\tilde{x}}$	ubrzanje točke $P'$ u smjeru osi $X_I$
$\dot{x}_C$	brzina točke $C$ u smjeru osi $X_I$
$\dot{x}_I$	brzina robota u smjeru osi $X_I$
$\dot{x}_R$	brzina robota u smjeru osi $X_R$
$\dot{x}_i$	brzina robota $i$ u smjeru osi $x$
$\dot{\tilde{y}}$	brzina točke $P'$ u smjeru osi $Y_I$
$\ddot{\tilde{y}}$	ubrzanje točke $P'$ u smjeru osi $Y_I$
$\dot{y}_C$	brzina robota u smjeru osi $Y_I$
$\dot{y}_I$	brzina robota u smjeru osi $Y_I$
$\dot{y}_R$	brzina robota u smjeru osi $Y_R$
$\dot{y}_i$	brzina robota $i$ u smjeru osi $y$
$\dot{z}_i$	vektor brzine robota $i$
$\vec{\delta}_{12}$	vektor koji definira željenu udaljenost između robota 1 i 2
$\dot{\theta}_R$	kutna brzina u lokalnom koordinatnom sustavu
$\dot{\phi}_L$	kutna brzina lijevog kotača
$\ddot{\phi}_L$	kutno ubrzanje lijevog kotača
$\dot{\phi}_R$	kutna brzina desnog kotača
$\ddot{\phi}_R$	kutno ubrzanje desnog kotača
$\dot{\omega}_i$	kutno ubrzanje robota $i$
$\Delta_{12}$	razlika između trenutne i željene pozicije robota 2
$\Delta_{\text{lok},12}$	razlika između trenutne i željene pozicije robota 2 u lokalnom koordinatnom sustavu
$D_f$	pojačanje koje definira važnost uspostavljanja formacije
$D_g$	pojačanje koje definira važnost dolaska robota do krajnjih pozicija
$E_f$	greška formacije
$E_g$	greška između trenutne pozicije robota i željene krajnje pozicije
$F_i$	primijenjena sila na robot $i$
$I_A$	moment inercije kotača i pogonskog motora
$I_P$	moment inercije cijelog robota s obzirom na točku $P$
$J_i$	moment inercije robota $i$

$K_f$	matrica koja definira važnost uspostavljanja formacije
$K_g$	matrica koja definira važnost dolaska robota do krajnjih pozicija
$M_L$	pogonski moment lijevog kotača
$M_R$	pogonski moment desnog kotača
$N_i$	broj susjeda robota $i$
$Q_i$	generalizirana sila
$T_R$	kinetička energija rotacije
$T_T$	kinetička energija translacije
$T_W$	kinetička energija rotacije kotača i rotora motora
$a_{12}$	trenutni kut između robota 1 i robota 2
$a_{2,12}$	vrijednost trenutnog kuta između robota 1 i 2 u lokalnom koordinatnom sustavu robota 2
$a_{ij}$	komponenta matrice susjedstva
$a_{j,i,j}$	vrijednost trenutnog kuta između robota $i$ i $j$ u lokalnom koordinatnom sustavu robota $j$
$d_{12}$	trenutna udaljenost između robota 1 i robota 2
$d_{ij}$	trenutna udaljenost između robota $i$ i robota $j$
$g_{ij}$	faktor postojanja komunikacije između robota $i$ i $j$
$k_v$	viskozni koeficijent trenja
$k_{ij}$	pojačanje sustava
$m_i$	masa robota $i$
$q_I$	vektor položaja i orijentacije robota u globalnom koordinatnom sustavu
$q_R$	vektor položaja i orijentacije robota u lokalnom koordinatnom sustavu
$q_i$	generalizirane koordinate
$u_{lok}$	upravljački ulaz izražen u lokalnom koordinatnom sustavu
$u_i$	upravljački ulaz
$v_C$	translacijska brzina težišta mobilnog robota
$v_i$	brzina translacije robota $i$
$\tilde{x}$	pozicija točke $P'$ u smjeru osi $X_I$
$x_{0,12}$	razlika između pozicije robota 1 i robota 2 u smjeru osi $X_0$
$x_{2,12}$	razlika između pozicije robota 1 i robota 2 u smjeru osi $X_2$
$x_I$	pozicija robota u smjeru osi $X_I$
$x_R$	pozicija robota u smjeru osi $X_R$
$x_i$	pozicija robota $i$ u smjeru osi $x$
$\tilde{y}$	pozicija točke $P'$ u smjeru osi $Y_I$
$y_{0,12}$	razlika između pozicije robota 1 i robota 2 u smjeru osi $Y_0$
$y_{0,12}$	razlika između pozicije robota 1 i robota 2 u smjeru osi $Y_2$
$y_I$	pozicija robota u smjeru osi $Y_I$
$y_R$	pozicija robota u smjeru osi $Y_R$
$y_i$	pozicija robota $i$ u smjeru osi $y$
$\dot{z}$	brzina robota

---

$z_i$	vektor pozicije robota $i$
$z_i^d$	željena konstantna lokacija robota $i$
$\alpha_{12}$	željeni kut između robota 1 i robota 2
$\alpha_{ij}$	željeni kut između robota $i$ i robota $j$
$\dot{\beta}(t)$	brzina skretanja
$\gamma_2$	kut orijentacije lokalnog koordinatnog sustava robota 2
$\gamma_j$	kut orijentacije lokalnog koordinatnog sustava robota $j$
$\ddot{\delta}(t)$	druga derivacija referentne trajektorije
$\delta_{12}$	željena udaljenost između robota 1 i robota 2
$\delta_{ij}$	željena udaljenost između robota $i$ i robota $j$
$\delta_{ij}(t)$	željeni razmak između pozicija robota $i$ i $j$
$\dot{\theta}$	kutna brzina
$\theta_R$	kut koji os robota zatvara sa lokalnim koordinatnim sustavom
$\theta_i$	orijentacija robota $i$
$\xi_i$	željena pozicija robota $i$
$\xi_j$	željena pozicija robota $j$
$\tau_i$	primijenjeni moment na robot $i$
$\dot{\varphi}$	kutna brzina kotača
$\omega_i$	kutna brzina robota $i$
$\{X_0, Y_0\}$	globalni nepomični koordinatni sustav
$\{X_1, Y_1\}$	globalni koordinatni sustav
$\{X_R, Y_R\}$	lokalni koordinatni sustav
$\{X_i, Y_i\}$	lokalni koordinatni sustav robota $i$
$A$	pozicija kotača
$B$	točka dodira podloge i kastorskog kotača
$C$	težište mobilnog robota
$E(t)$	ukupna greška
$F$	izgled formacije
$L$	razmak između kotača
$P$	referentna točka robota
$P'$	točka na udaljenosti $l$ od točke $P$
$R(\gamma_2)$	matrica rotacije $\gamma_2$
$R(\gamma_j)$	matrica rotacije $\gamma_j$
$R(\theta)$	matrica rotacije $\theta$
$T$	kinetička energija
$V$	potencijalna energija
$d$	razmak između točke dodira i točke okretanja kastorskog kotača
$k$	pojačanje sustava
$l$	udaljenost
$m$	masa robota
$n$	ukupni broj robota

---

---

$r$	radijus kotača
$t$	vrijeme
$v$	brzina translacije
$x$	pozicija robota na osi $x$
$y$	pozicija robota na osi $y$
$z$	pozicija robota
$\Phi$	Rayleigh-ova disipacijska funkcija
$\alpha$	kut koji os kotača zatvara s osi $X_R$
$\beta$	kut između osi kotača i tijela robota
$\beta(t)$	kut skretanja kastorskog kotača
$\gamma$	pojačanje koje definira važnost izjednačavanja brzina robota i trajektorije
$\delta(t)$	referentna trajektorija
$\varepsilon$	pojačanje koje definira važnost izjednačavanja brzina robota
$\theta$	kut kojeg zatvaraju osi između globalnog i lokalnog koordinatnog sustava
$\sigma$	pojačanje koje definira važnost praćenja željene trajektorije
$\varphi(t)$	kut rotacije kotača
$\omega$	kutna brzina

## SAŽETAK

U ovom radu proučavamo problem gibanja skupine mobilnih robota u formaciji. U prvom dijelu rada predstavili smo kinematiku i dinamiku gibanja mobilnog robota s diferencijalnim pogonom. Nakon toga opisali smo upravljačke sustave koje ćemo koristiti u ovom radu, model s jednim te model s dva integratora, koji predstavljaju kinematiku odnosno dinamiku gibanja mobilnih robota. Na nizu simulacija prikazali smo osnovne načine gibanja mobilnih robota te napravili usporedbu takvog gibanja za model s jednim i dva integratora.

Posebnu pozornost u ovom radu posvetili smo različitim načinima gibanja mobilnih robota u formaciji. Predstavili smo tri moguća rješenja te ih pomoću prikladno osmišljenih simulacija ispitali. Prva dva predstavljena rješenja su adaptacija upravljačkih zakona prikazanih u literaturi. Prvo ponuđeno rješenje predstavlja gibanje formacije kod kojeg su zadane željene krajnje pozicije mobilnih robota. Upravljački zakon prilagodili smo za kinematiku i dinamiku gibanja te ispitali na nizu simulacija s različitim brojem robota i grafovima komunikacije. Drugo ponuđeno rješenje odnosi se na gibanje formacije mobilnih robota po unaprijed zadanim putanjama. Predstavljeni upravljački zakon smo unaprijedili za kinematiku, dok smo ga za dinamiku gibanja samostalno razvili. Taj upravljački zakon smo ispitali na nizu simulacija s različitim brojem robota i grafovima komunikacije. Treće ponuđeno rješenje smo samostalno razvili te ispitali na seriji simulacija. U našem upravljačkom zakonu upravljamo udaljenošću i kutom između dva robota. Za razliku od rješenja predstavljenih u literaturi naš način gibanja u formaciji koristi jednostavniji upravljački zakon, omogućuje direktno upravljanje s vodećim robotom, te kao najveću prednost nudi mogućnost jednostavnog mijenjanja orijentacije cijele formacije.

Kao dodatak radu razvili smo i aplikaciju u programu Matlab koja omogućuje jednostavnu izradu simulacija prikazanih u ovom radu. Promjenom zadanih parametara moguće je ispitati kako njihove različite vrijednosti utječu na rezultate simulacije.

Ključne riječi: mobilni roboti, upravljanje formacijom, distribuirano upravljanje

## SUMMARY

In this thesis, we study the problem of coordinated motion of group of mobile robots in formation. In the first part of this thesis we presented kinematics and dynamics of differential drive mobile robots motion. We described single and double-integrator model which represent kinematics and dynamics of mobile robot motion, respectively. Basic mobile robots motions and comparison between single and double-integrator models are presented.

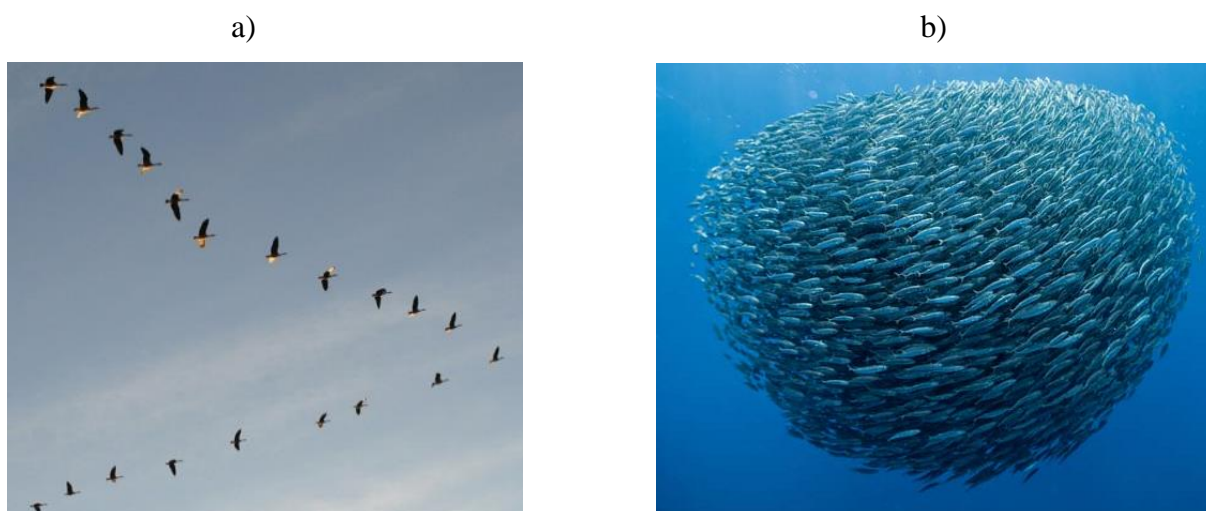
The main part of this thesis deals with different ways of achieving motion of mobile robots in formation. We presented three possible solutions to this problem and each solution is tested in a number of suitable simulations. First two solutions are adaptations of control laws presented in literature. First solution describes formation maneuvers where we define target positions of mobile robots. We adapted the control law from the literature and tested it for various number of robots and various types of communication graphs. Second solution describes formation maneuvers where we define referent time-varying trajectories for all mobile robots. Starting from the known solutions from the literature we have presented a novel control law for kinematics of motion as well as a new control law for dynamics of mobile robots motion. The proposed solutions have been tested for various number of robots and various types of communication graphs. The third solution represents a novel contribution and is capable to control the relative distance and relative angle between the two neighbouring robots. The presented control law is simpler than the similar laws found in the literature. Furthermore, it allows us to directly control the leading robot, while at the same time it offers a possibility to control orientation of the complete formation.

As an addition to this thesis we developed an application in computer program Matlab that enables simple creation of simulations presented in this thesis. We can change different parameters and observe the influence of their values on the results of simulations.

Key words: mobile robots, formation control, distributed control

## 1. UVOD

Ubrzani razvoj i smanjivanje veličine računala, senzora i aktuatora omogućio je i ubrzani razvoj i sve češće korištenje velikog broja autonomnih robota (zračnih, pomorskih i kopnenih) koji radeći zajedno obavljaju različite zadatke. Ideja za zajednički rad (gibanje u formaciji, rad u grupi) autonomnih robota došla je iz prirode, gdje skupine životinja zajedno rade da bi postigle određeni cilj (Slika 1.).



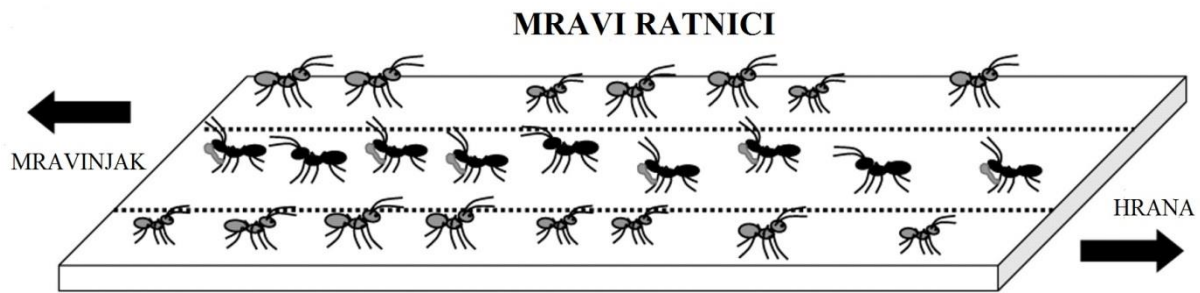
**Slika 1. Formacije životinja; a) Ptice [1], b) Ribe [2]**

Slika 1. a) prikazuje gibanje ptica u V-formaciji. Svaka ptica leti malo iznad ptice ispred sebe što joj omogućuje let uz smanjen otpor zraka, a time i uštedu energije. Ovakav način letenja omogućuje i jednostavnu kontrolu svih ptica u grupi te lakšu koordinaciju i komunikaciju unutar grupe [3].

Slika 1. b) prikazuje jato riba koje se zajedničkom akcijom pokušava obraniti od predatora. Na ovaj način je predatoru otežano fokusiranje na jedan plijen, te predator postaje zbunjen (eng. *predator confusion effect* – Milinski i Heller). Uz obranu od predatora, život u jatu omogućava i lakši pronalazak hrane te olakšano razmnožavanje [4].

Slika 2. prikazuje gibanje mrava ratnika u formaciji. Na ovaj način (gibanje jedan iza drugog poput automobila na cesti, to jest stvaranje "prometnih traka") olakšava se koordinacija i sprječava sudaranje i stvaranje gužvi [5].





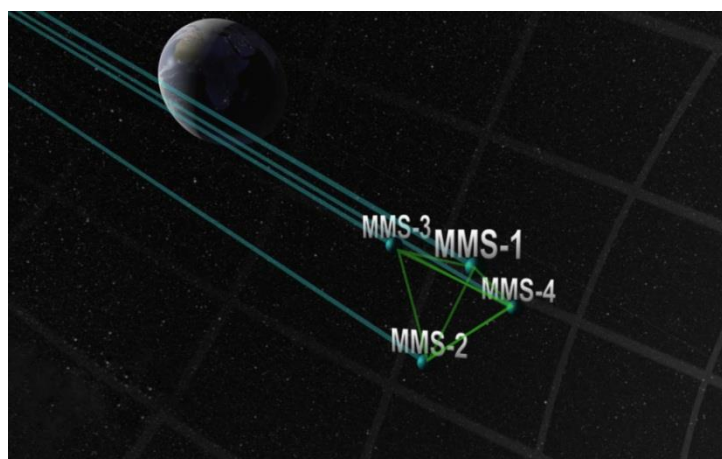
Slika 2. Mravi ratnici [6]

Grupe autonomnih robota omogućuju rad u formaciji koji ima brojne prednosti u odnosu na samostalne autonomne robote, kao što su: niža cijena konstruiranja, proizvodnje i rada takvih sustava, veća robusnost, fleksibilnost, preciznost, efikasnost, energetska učinkovitost i vjerojatnost uspjeha [7].

### 1.1 Primjene formacija mobilnih robota (vozila)

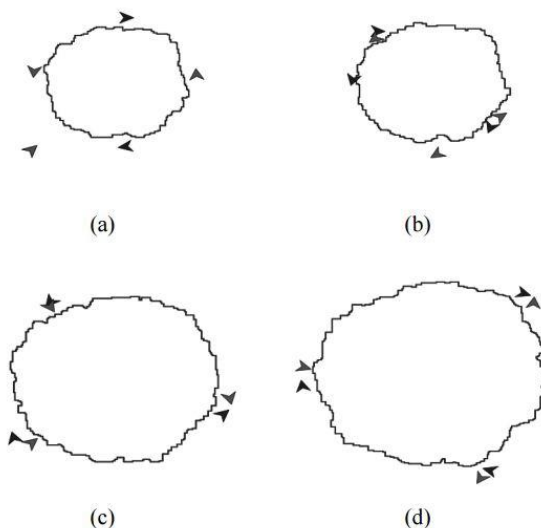
Trenutačno postoji veliki broj mogućih primjena grupe mobilnih robota. Ovdje ćemo nabrojiti i pokazati primjere nekih od njih.

- a) Primjena u svemiru: gibanje letjelica u formaciji, interplanetarna istraživanja, gibanje satelita u formaciji poput MMS (Magnetospheric Multiscale mission) koju je izvela NASA. Slika 3. pokazuje 4 satelita koja lete u formaciji na granici Zemljinog magnetskog polja prateći utjecaj Sunčevog magnetskog zračenja na Zemljino magnetsko polje. Ovakva tetraedarska formacija omogućuje praćenje promjena u 3 dimenzije. Da se sateliti ne gibaju u ovoj formaciji, ne bi mogli pratiti potpune promjene strukture u 3 dimenzije dok lete kroz nju [8].



Slika 3. MMS; ilustracija [8]

- b) Primjena grupe mobilnih robota za rad u opasnim atmosferama poput praćenja razine zračenja oko nuklearnih elektrana, praćenja širenja požara. Simulacija prema [7, 9] prikazuje formaciju sastavljenu od 6 mobilnih robota (bespilotnih letjelica) koji prate širenje šumskog požara (Slika 4.).



**Slika 4. Praćenje širenja šumskog požara; a) lansiranje robota s bazne stanice, b) gibanje robota prije idealno uspostavljene formacije, c) i d) praćenje požara u idealnoj formaciji [7, 9]**

- c) Primjena grupe mobilnih robota u transportu poput vožnje u koloni (eng. *platooning*). Na Slici 5. se vidi vožnja u koloni tri autonomna kamiona tvrtke Volvo. Ovakav način vožnje smanjuje potrošnju goriva, smanjuje emisiju ugljičnog dioksida, smanjuje mogućnost nastajanja gužvi i smanjuje vrijeme potrebno za dostavu, a isto tako može i povećati sigurnost na cesti [10].



**Slika 5. Vožnja u koloni [10]**

- d) Primjena mobilnih robota u vojne svrhe: za pretraživanje terena, snimanje terena, izviđanje neprijateljskog teritorija, patroliranja određenim područjem, zaštita važnog vozila... Na Slici 6. se vidi formacija autonomnih brodova Američke ratne mornarice za vrijeme vojnih vježbi (zaštita važnog plovila od neprijateljskog napada te presretanje potencijalno opasnih plovila). Na ovaj način omogućeno je brže reagiranje na opasnosti uz izbjegavanje ljudskih žrtava [11].



**Slika 6. Formacija autonomnih brodova [11]**

- e) Primjena formacija robota u poljoprivredne svrhe poput autonomnih bespilotnih letjelica za zalijevanje ili kontrolu usjeva, autonomni traktori za obradu zemlje, žetvu, sjetvu... Na Slici 7. se vidi primjer autonomnog sustava za žetvu usjeva. Na ovaj način je omogućena preciznija i brža žetva [12].



**Slika 7. Autonomna žetva [12]**

## 1.2 Pregled rada

Ovaj rad je organiziran na sljedeći način:

U poglavlju 2. bavimo se kinematikom mobilnog robota, definirat ćemo koordinatne sustave, prikazati kakva ograničenja čine kotači te ćemo izvesti kinematičke jednadžbe koje ćemo koristiti za simulacije gibanja mobilnih robota u ovom radu.

Poglavlje 3. rezervirano je za dinamiku mobilnog robota, pa ćemo u tom poglavlju napraviti detaljan izvod dinamičkog modela robota. Za potrebe simulacija, u ovom radu nećemo koristiti ovaj detaljan model, već ćemo koristiti pojednostavljeni prikaz dinamike koji za naše potrebe dovoljno dobro prikazuje stvarno stanje.

U poglavlju 4. obradit ćemo upravljačke sustave koje ćemo koristiti za izradu simulacija modela točkastog robota, a to su model s jednim integratorom (predstavlja kinematiku točkastog robota) i model s dva integratora (predstavlja dinamiku točkastog robota).

U poglavlju 5. su prikazane 4 skupine simulacija: praćenje, kružno praćenje, susretanje i uspostavljanje formacije. Objasniti ćemo detaljno sve simulacije te ćemo rezultate prikladno grafički prikazati. Također ćemo objasniti i koje su razlike među njima i koji upravljački zakoni su potrebni da bi roboti izvršili naše naredbe.

Poglavlje 6. predstavlja najvažnije poglavlje ovog rada, u kojem prikazujemo tri načina za gibanje robota u formaciji. Kod prvog načina roboti se gibaju linijski u formaciji od nekih početnih do krajnjih točaka. Kod drugog načina roboti se gibaju u formaciji po krivulji. Kod trećeg načina roboti se gibaju po krivulji koja nije nužno pravocrtna, te su moguće rotacije cjelokupne formacije. Za sva tri načina napravljene su i prikladno predstavljene brojne simulacije koje prikazuju efikasnost predstavljениh upravljačkih zakona.

Poglavlje 7. daje zaključke o obrađenoj temi.

U Prilogu A dan je opis i upute za korištenje napravljene aplikacije koja je namijenjena budućim studentima Fakulteta strojarstva i brodogradnje kao pomoć i kao uvod u područje mobilnih robota.

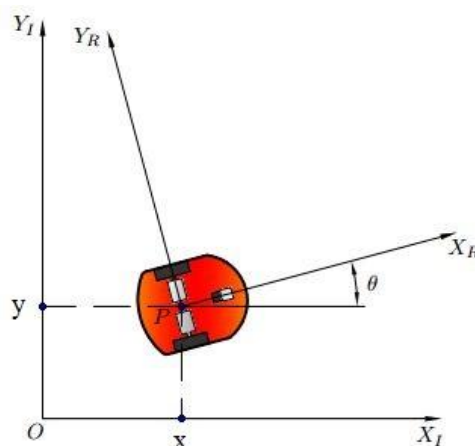
## 2. KINEMATIKA MOBILNOG ROBOTA

U mobilnoj robotici, potrebno je razumjeti mehanička ponašanja robota da bismo konstruirali robote koji su prikladni za zadatke koje trebaju ispuniti i kako bismo pravilno osmislili njihovo upravljanje [13].

Kod robota s kotačima, svaki kotač pridonosi gibanju robota i istodobno postavlja ograničenja tom gibanju. U ovom radu pretpostavit ćemo da se robot ponaša kao kruto tijelo na kotačima, koji su nedeformabilni, i da se robot giba u horizontalnoj ravnini. Ovo je česta pretpostavka u literaturi [13] i za naše potrebe dovoljno dobro aproksimira stvarnu situaciju.

### 2.1 Definiranje koordinatnih sustava i veza između njih

Da bismo pravilno mogli odrediti poziciju robota u ravnini, trebamo matematički opisati odnos između globalnog nepomičnog koordinatnog sustava i lokalnog koordinatnog sustava robota koji su ilustrirani na Slici 8. Osi  $X_I$  i  $Y_I$  definiraju proizvoljni inercijski koordinatni sustav u ravnini kao globalni koordinatni sustav s ishodištem  $O: \{X_I, Y_I\}$ . Da bismo odredili poziciju robota, izaberemo točku  $P$  na tijelu robota kao njegovu referentnu točku.  $\{X_R, Y_R\}$  predstavlja dvije osi čije je ishodište točka  $P$  na tijelu robota, a zajedno predstavljaju lokalni koordinatni sustav robota. Pozicija točke  $P$  u globalnom koordinatnom sustavu određena je pomoću koordinata  $x$  i  $y$ , dok je orijentacija robota određena razlikom između orijentacija globalnog i lokalnog koordinatnog sustava koju označavamo s  $\theta$  [13].



Slika 8. Globalni i lokalni koordinatni sustav [14]

Definirajmo vektor  $q_I$  kao vektor koji sadrži položaj i orijentaciju robota u globalnom koordinatnom sustavu:

$$q_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}. \quad (1)$$

Za transformaciju gibanja iz lokalnog u globalni koordinatni sustav koristit će nam matrica rotacije koja je definirana na sljedeći način:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Budući da je lokalni koordinatni sustav fiksiran na robotu, pozicija robota je u njemu uvijek jednaka nuli pa je i  $\theta_R = \theta$ . Transformacija između komponenti gibanja u lokalnom i globalnom koordinatnom sustavu definirana je relacijom:

$$\dot{q}_I = R(\theta)\dot{q}_R. \quad (3)$$

Jednadžba (3) nam daje mogućnost da shvatimo kako se brzina u lokalnom koordinatnom sustavu transformira u brzinu u globalnom koordinatnom sustavu, tj. ako poznajemo brzinu  $\dot{q}_R = [\dot{x}_R \quad \dot{y}_R \quad \dot{\theta}_R]^T$  u lokalnom koordinatnom sustavu, možemo izračunati komponente gibanja u globalnom koordinatnom sustavu [13].

## 2.2 Kinematska ograničenja kotača

Prvi korak prema modelu kinematike robota predstavlja izvođenje izraza za ograničenja gibanja pojedinih kotača. Prilikom izvođenja uvest ćemo sljedeće pretpostavke:

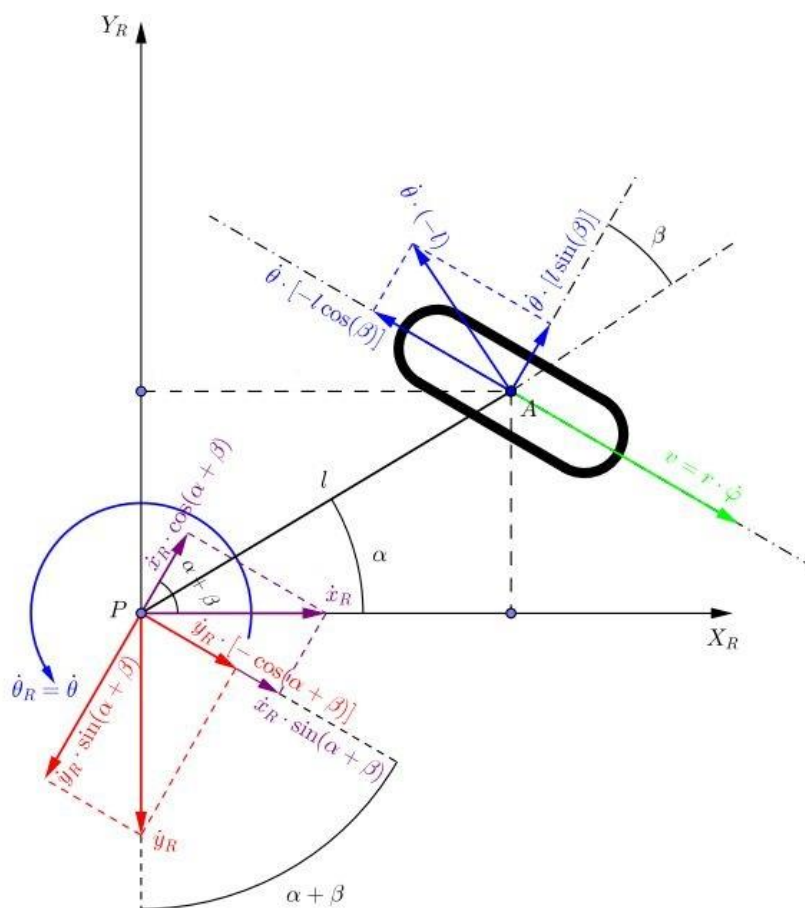
- ravnina kotača je uvijek vertikalna i u svim slučajevima postoji samo jedna točka dodira između kotača i podloge;
- pretpostavljamo da nema klizanja u toj točki dodira, tj. pretpostavljamo čisto kotrljanje.

Pod ovim pretpostavkama predstaviti ćemo dva ograničenja za svaki tip kotača. Prvo ograničenje je da se kotač mora kotrljati u prikladnom smjeru kad je robot u gibanju

(ograničenje kotrljanja). Drugo ograničenje je da nema bočnog klizanja kotača (ograničenje klizanja) [13].

### 2.2.1 Standardni kotač

Prvo ćemo opisati standardni kotač. Njegova pozicija je fiksna i nema vertikalne osi rotacije za skretanje tako da se može gibati samo naprijed i nazad u ravnini. Na Slici 9. prikazan je takav kotač i njegovi parametri. Pozicija kotača  $A$  je određena pomoću udaljenosti  $l$  i kuta  $\alpha$ . Kut između osi kotača i tijela robota je označen sa  $\beta$ , a on je konstantan budući da ovaj kotač ne može skretati. Radijus kotača je označen sa  $r$ , dok je kut rotacije kotača označen sa  $\varphi(t)$  [13].



Slika 9. Fiksni standardni kotač i njegovi parametri [14]

Ograničenje kotrljanja za ovaj kotač opisano je sljedećom relacijom

$$\dot{x}_R \cdot \sin(\alpha + \beta) - \dot{y}_R \cdot \cos(\alpha + \beta) - \dot{\theta}_R \cdot l \cdot \cos(\beta) - r\dot{\varphi} = 0, \quad (4)$$

a ono je izvedeno iz uvijeta da svako gibanje u smjeru ravnine kotača mora biti popraćeno odgovarajućom količinom okreta kotača tako da postoji čisto kotrljanje u točki dodira. Jednadžba (4) se može zapisati u matričnom obliku kao

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \end{bmatrix} \dot{q}_R - r\dot{\varphi} = 0. \quad (5)$$

Koristeći jednadžbu (3) jednadžba (5) se može preurediti tako da imamo

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \end{bmatrix} R^{-1}(\theta) \dot{q}_I - r\dot{\varphi} = 0. \quad (6)$$

Ograničenje klizanja za ovaj kotač uzrokuje da komponenta gibanja kotača okomito na ravninu kotača mora biti nula, pa stoga imamo

$$\dot{x}_R \cdot \cos(\alpha + \beta) + \dot{y}_R \cdot \sin(\alpha + \beta) + \dot{\theta}_R \cdot l \cdot \sin(\beta) = 0, \quad (7)$$

ili zapisano u matričnom obliku

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R^{-1}(\theta) \dot{q}_I = 0. \quad (8)$$

## 2.2.2 Kastorski kotač

Sljedeće ćemo opisati kastorski kotač koji je prikazan na Slici 10. Kastorski kotač se može okretati oko vertikalne osi koja ne prolazi kroz točku dodira kotača s podlogom. Točka dodira je sada točka  $B$ , koja je kruto povezana s točkom  $A$  oko koje se kastorski kotač okreće. Kastorski kotač ima dva parametra u funkciji vremena:  $\varphi(t)$  predstavlja kut rotacije kotača kao i prije, dok je  $\beta(t)$  kut skretanja i orijentacije kastorskog kotača.

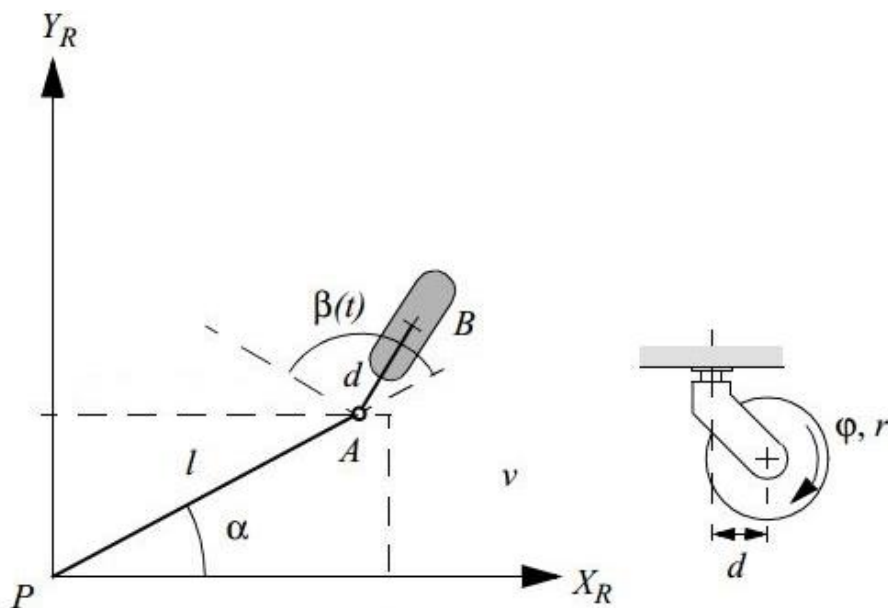
Ograničenje kotrljanja za kastorski kotač identično je onom kod standardnog kotača (jednadžba (6)) uz iznimku da kut  $\beta$  ovisi o vremenu, pa možemo pisati

$$\dot{x}_R \cdot \sin(\alpha + \beta(t)) - \dot{y}_R \cdot \cos(\alpha + \beta(t)) - \dot{\theta}_R \cdot l \cdot \cos(\beta(t)) - r\dot{\varphi} = 0, \quad (9)$$

ili zapisano u matričnom obliku

$$\begin{bmatrix} \sin(\alpha + \beta(t)) & -\cos(\alpha + \beta(t)) & -l \cos \beta(t) \end{bmatrix} R^{-1}(\theta) \dot{q}_I - r\dot{\varphi} = 0. \quad (10)$$





Slika 10. Kastorski kotač [13]

Bočna sila djeluje na kastorski kotač u točki  $A$ . Zbog razmaka između točke dodira i točke okretanja  $A$ , postoji komponenta gibanja kotača okomito na ravninu kotača, pa je ograničenje klizanja slično ograničenju kotrljanja i definiira se sljedećom relacijom

$$\dot{x}_R \cdot \cos(\alpha + \beta(t)) + \dot{y}_R \cdot \sin(\alpha + \beta(t)) - \dot{\theta}_R \cdot [d + l \cdot \sin(\beta(t))] + \dot{\beta}(t) = 0, \quad (11)$$

ili zapisano u matričnom obliku

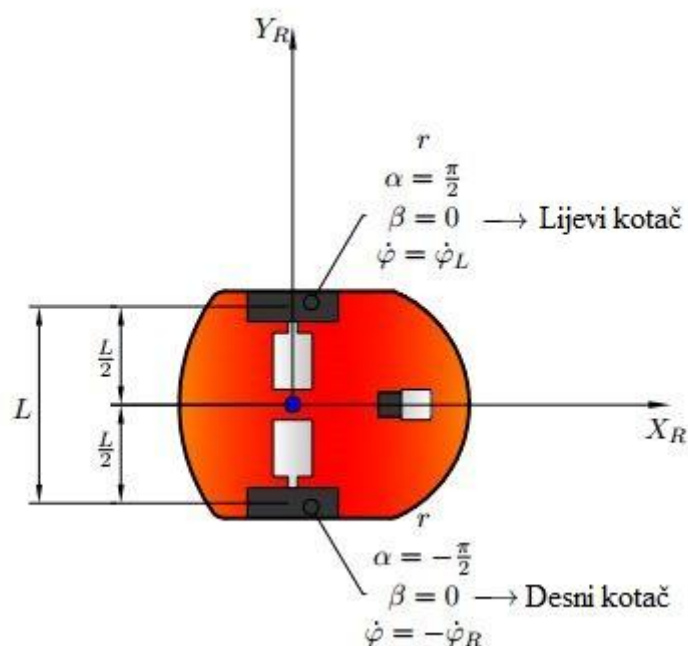
$$[\cos(\alpha + \beta(t)) \quad \sin(\alpha + \beta(t)) \quad d + l \sin \beta(t)] R^{-1}(\theta) \dot{q}_1 + \dot{\beta}(t) = 0. \quad (12)$$

U jednadžbi (12), svako gibanje okomito na ravninu kotača mora biti balansirano s ekvivalentnom i suprotnom vrijednošću okretnog gibanja kastorskog kotača. Ovo je ključno za primjenu kastorskog kotača zato što je postavljanjem vrijednosti  $\dot{\beta}$  svako proizvoljno bočno gibanje prihvatljivo. Zbog toga se može zaključiti da za bilo koje gibanje robota, postoji neka vrijednost brzine okretanja kotača  $\dot{\phi}$  i brzine skretanja  $\dot{\beta}$  takva da su ograničenja zadovoljena. Iako je kinematika kastorskog kotača složena, takav kotač ne predstavlja stvarna ograničenja na kinematiku robota [13].

## 2.3 Kinematika mobilnog robota – kinematički model gibanja

U ovom poglavlju izvest ćemo kinematički model gibanja mobilnog robota, prvo za lokalni koordinatni sustav, a zatim za globalni koordinatni sustav. Nakon toga ćemo izvesti i tzv. *unicycle* model robota.

U našem slučaju imamo robot s dva standardna kotača koji ne mogu skretati ( $\beta = 0$ ) i jedan kastorski kotač. Lokalni koordinatni sustav je definiran tako da os  $Y_R$  prolazi kroz os kotača, a ishodište lokalnog koordinatnog sustava (točka  $P$ ) je na polovini udaljenosti između kotača, kao što se vidi na Slici 11. Kut  $\alpha$  označava kut koji os kotača zatvara s osi  $X_R$ , kut  $\beta$  je kut skretanja kotača,  $r$  je promjer kotača, dok je  $s$   $L$  određen razmak između kotača.



Slika 11. Parametri kotača [14]

Brzina robota u globalnom koordinatnom sustavu ovisi o kutu  $\theta$  i brzini okretanja kotača  $\dot{\phi}_R$  i  $\dot{\phi}_L$ :

$$\dot{q}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\theta, \dot{\phi}_R, \dot{\phi}_L). \quad (13)$$

Pomoću jednadžbe (3) možemo izračunati gibanje robota u globalnom koordinatnom sustavu iz gibanja u lokalnom koordinatnom sustavu i obratno.

Prvo ćemo pokazati utjecaj svakog kotača na gibanje robota u lokalnom koordinatnom sustavu, a nakon toga će se to gibanje transformirati u globalni koordinatni sustav.

Uvrštavanjem vrijednosti za  $\alpha$  i  $\beta$  definiranih na Slici 11. u jednadžbu (4) dobije se za lijevi kotač

$$\dot{x}_R - \dot{\theta}_R \cdot \frac{L}{2} - r \cdot \dot{\phi}_L = 0, \quad (14)$$

dok se za desni kotač dobije

$$-\dot{x}_R - \dot{\theta}_R \cdot \frac{L}{2} + r \cdot \dot{\phi}_R = 0. \quad (15)$$

Kombinirajući jednadžbe (14) i (15) izvode se sljedeće jednadžbe

$$\dot{x}_R = \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L), \quad (16)$$

$$\dot{\theta}_R = \frac{r}{L}(\dot{\phi}_R - \dot{\phi}_L). \quad (17)$$

Ponavljajući postupak za jednadžbu (7) dolazimo do izraza

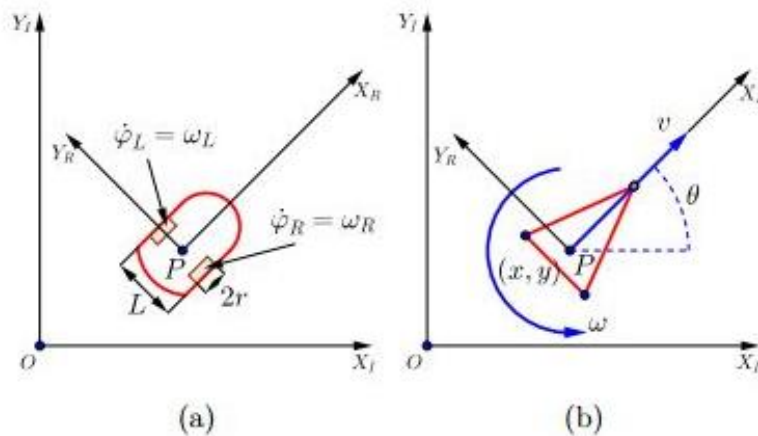
$$\dot{y}_R = 0. \quad (18)$$

Iz jednadžbe (18) je vidljivo da je zbog ograničenja klizanja standardnog kotača, brzina robota u smjeru  $Y_R$  osi uvijek jednaka nuli.

Sada ćemo gibanje opisano jednadžbama (16), (17) i (18) korištenjem jednadžbe (3) prikazati u globalnom koordinatnom sustavu. Imamo

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L) \\ 0 \\ \frac{r}{L}(\dot{\phi}_R - \dot{\phi}_L) \end{bmatrix}. \quad (19)$$

Direktni kinematički model definiran jednadžbom (19) nam pokazuje kako svaki kotač pridonosi brzini u globalnom koordinatnom sustavu. Zbog mogućnosti promjene smjera vožnje promjenom brzine vrtnji lijevog i desnog kotača, ovaj robot je mobilni robot s diferencijalnim pogonom. Zbog nepraktičnosti primjene (19), mobilni robot s diferencijalnim pogonom može biti prikazan pomoću tzv. unicycle modela, kao na Slici 12.



**Slika 12. Transformacija modela diferencijalnog pogona (a) u unicycle model (b) [14]**

Budući da translacijska brzina  $v$  i kutna brzina  $\omega$  točke  $P$  odgovaraju  $\dot{x}_R$  i  $\dot{\theta}_R$  definiranim jednažbama (16) i (17), kinematika modela diferencijalnog pogona dana jednažbom (19) može se zapisati i kao

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix}, \quad (20)$$

ili u pojednostavljenom obliku kao:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix}. \quad (21)$$

Jednažba (21) nam daje komponente brzina duž globalnih osi  $X_I$  i  $Y_I$ , gdje su translacijska brzina  $v$  i kutna brzina  $\omega$  direktno povezane s brzinama kotača kako slijedi:

$$v = \dot{x}_R = \frac{r}{2}(\omega_R + \omega_L), \quad (22)$$

$$\omega = \dot{\theta}_R = \frac{r}{L}(\omega_R - \omega_L). \quad (23)$$

Stoga, kod unicycle modela ulazne upravljačke veličine su  $v$  i  $\omega$ , kao na Slici 13.



**Slika 13. Ulazi i izlazi iz modela unicycle-a [14]**

### 3. DINAMIKA MOBILNOG ROBOTA

Dinamički model mobilnog robota uključuje razmatranje svojstava koja ovise o masama, inercijama, silama trenja, centrifugalnim silama, momentima... Dinamički modeli omogućuju nam bolje shvaćanje strukture i rada sustava u ranoj fazi razvoja/konstrukcije sustava. Priprema i uporaba dinamičkih modela omogućuje pronalazak nedostataka i grešaka modela realnog sustava. Njihovo ispravljanje je jednostavnije i jeftinije u virtualnoj fazi u usporedbi sa stvarnim robotima [15]. Izvod dinamičkih jednažbi prikazanih u ovom poglavlju je napravljen prema [14] i [15].

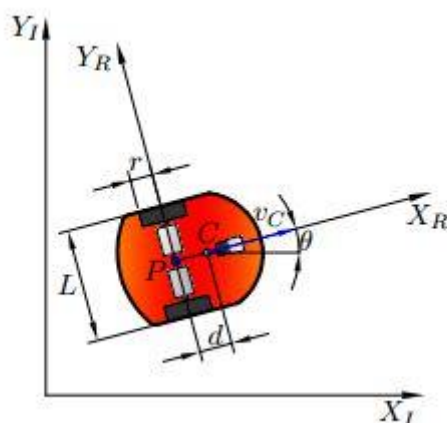
Dinamičke jednažbe gibanja izvest ćemo pomoću Euler-Lagrangeove jednažbe:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial \Phi}{\partial \dot{q}_i} = Q_i, \quad (24)$$

gdje se  $L$  definira kao razlika između kinetičke energije  $T$  i potencijalne energije  $V$ .  $q_i$  su generalizirane koordinate,  $Q_i$  je generalizirana sila koja djeluje na sustav, a  $\Phi$  je Rayleigh-ova disipacijska funkcija. Pod pretpostavkom da se robot giba u horizontalnoj ravnini, potencijalna energija je konstantna pa trebamo pronaći samo kinetičku energiju robota. Ukupna kinetička energija sustava je dana pomoću sljedeće jednažbe:

$$T = T_T + T_R + T_W, \quad (25)$$

gdje je  $T_T$  kinetička energija translacije,  $T_R$  je kinetička energija rotacije mobilnog robota, dok je  $T_W$  kinetička energija rotacije kotača i rotora motora.



Slika 14. Brzina translacije težišta mobilnog robota [14]

Vrijednosti ovih kinetičkih energija su

$$T_T = \frac{1}{2}mv_C^2 = \frac{1}{2}m(\dot{x}_C^2 + \dot{y}_C^2), \quad (26)$$

$$T_R = \frac{1}{2}I_P\dot{\theta}^2, \quad (27)$$

$$T_W = \frac{1}{2}I_A\dot{\phi}_R^2 + \frac{1}{2}I_A\dot{\phi}_L^2, \quad (28)$$

gdje je  $m$  masa cijelog mobilnog robota,  $v_C$  je linearna brzina težišta mobilnog robota,  $I_P$  je moment inercije cijelog mobilnog robota s obzirom na točku  $P$ ,  $\dot{\theta}$  je kutna brzina mobilnog robota,  $I_A$  je moment inercije kotača i pogonskog motora, dok su  $\dot{\phi}_R$  i  $\dot{\phi}_L$  kutne brzine desnog i lijevog kotača s radijusom  $r$ , kako to prikazuje Slika 14.

Brzine točke  $P$  su dane jednadžbom (21). Pomoću jednadžbi (22) i (23) brzine točke  $P$  se mogu izraziti kao:

$$\dot{x} = \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L)\cos(\theta), \quad (29)$$

$$\dot{y} = \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L)\sin(\theta), \quad (30)$$

$$\dot{\theta} = \frac{r}{L}(\dot{\phi}_R - \dot{\phi}_L). \quad (31)$$

Komponente translacijske brzine težišta robota  $C$  definirane su kao:

$$\dot{x}_C = \dot{x} + d\dot{\theta}\sin(\theta), \quad (32)$$

$$\dot{y}_C = \dot{y} + d\dot{\theta}\cos(\theta), \quad (33)$$

gdje je  $d$  udaljenost između točaka  $P$  i  $C$ ,  $\dot{x}_C$  je brzina točke  $C$  u smjeru osi  $X_I$ , a  $\dot{y}_C$  je brzina točke  $C$  u smjeru osi  $Y_I$ .

Kombinirajući jednadžbe (29), (30) i (31) s jednadžbama (32) i (33) dobiva se izraz za ukupnu kinetičku energiju

$$\begin{aligned} T = & \left[ \frac{mr^2}{8} + \frac{r^2}{2L^2}(md^2 + I_P) + \frac{I_A}{2} \right] \dot{\phi}_R^2 \\ & + \left[ \frac{mr^2}{4} + \frac{r^2}{L^2}(md^2 + I_P) \right] \dot{\phi}_R\dot{\phi}_L \\ & + \left[ \frac{mr^2}{8} + \frac{r^2}{2L^2}(md^2 + I_P) + \frac{I_A}{2} \right] \dot{\phi}_L^2. \end{aligned} \quad (34)$$

Generaliziranu silu  $Q_i$  definiramo kao desni i lijevi pogonski moment  $M_R$  i  $M_L$ , dok su  $\varphi_R$  i  $\varphi_L$  generalizirane koordinate. Rayleigh-ova disipacijska funkcija koja opisuje viskozno trenje motorskog sustava desnog i lijevog kotača definira se kao

$$\Phi = \frac{1}{2} k_v (\dot{\varphi}_R^2 + \dot{\varphi}_L^2), \quad (35)$$

gdje je  $k_v$  viskozni koeficijent trenja. Uvrštavanjem u jednadžbu (24) dobije se

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\varphi}_R} \right) - \frac{\partial T}{\partial \varphi_R} = M_R - k_v \dot{\varphi}_R, \quad (36)$$

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\varphi}_L} \right) - \frac{\partial T}{\partial \varphi_L} = M_L - k_v \dot{\varphi}_L. \quad (37)$$

Uvrštavanjem (34) u (36) i (37) i sređivanjem dobiju se dinamičke jednadžbe koje opisuju gibanje mobilnog robota

$$a\ddot{\varphi}_R + b\ddot{\varphi}_L = M_R - k_v \dot{\varphi}_R, \quad (38)$$

$$a\ddot{\varphi}_L + b\ddot{\varphi}_R = M_L - k_v \dot{\varphi}_L, \quad (39)$$

gdje su parametri  $a$  i  $b$  jednaki

$$a = \frac{mr^2}{4} + \frac{r^2}{L^2} (md^2 + I_p) + I_A, \quad (40)$$

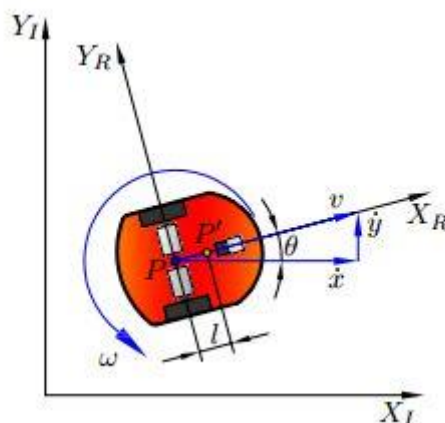
$$b = \frac{mr^2}{4} + \frac{r^2}{L^2} (md^2 + I_p). \quad (41)$$

## 4. UPRAVLJAČKI SUSTAV

Da bismo napravili upravljački sustav mobilnog robota koristit ćemo se kinematičkim modelom *unicycle*-a iz jednadžbe (21). U ovom poglavlju izvest ćemo linearni model prikladan za sintezu upravljačkog sustava. Ovaj model će nam omogućiti da se *unicycle* robot ponaša kao točkasti robot, a to je poželjno zbog jednostavnosti i lakše implementacije jer se tada zanemaruje orijentacija robota. Za izvod koristit ćemo [14], [16] i [7].

### 4.1 Model s jednim integratorom (kinematika)

Cilj je da se *unicycle* ponaša kao:  $\dot{z} = u$  (gdje je  $\dot{z} = [\dot{x} \ \dot{y}]^T$ , dok  $u = [u_1 \ u_2]^T$  predstavlja željene brzine). To se može napraviti na način da definiramo novu točku na robotu, točku  $P'$  koja se nalazi na udaljenosti  $l$  od točke  $P$  kao na Slici 15.



Slika 15. Definiranje točke  $P'$  [14]

Pozicija te nove točke dana je pomoću jednadžbi

$$\tilde{x} = x + l \cos(\theta), \quad (42)$$

$$\tilde{y} = y + l \sin(\theta). \quad (43)$$

Njihovim deriviranjem po vremenu dobije se

$$\dot{\tilde{x}} = \dot{x} - l\dot{\theta} \sin(\theta), \quad (44)$$

$$\dot{\tilde{y}} = \dot{y} + l\dot{\theta} \cos(\theta). \quad (45)$$



Uvrštavanjem jednadžbe (21) u jednadžbe (44) i (45) dobije se

$$\dot{\tilde{x}} = v \cos(\theta) - l\dot{\theta} \sin(\theta), \quad (46)$$

$$\dot{\tilde{y}} = v \sin(\theta) + l\dot{\theta} \cos(\theta). \quad (47)$$

Sada uvodimo pretpostavku da točkom možemo  $P'$  izravno upravljati tako da

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (48)$$

gdje su  $u_1$  i  $u_2$  željene brzine. Kombinirajući jednadžbe (46), (47) i (48) dobijemo

$$u_1 = v \cos(\theta) - l\dot{\theta} \sin(\theta), \quad (49)$$

$$u_2 = v \sin(\theta) + l\dot{\theta} \cos(\theta), \quad (50)$$

ili zapisano u matričnom obliku

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (51)$$

Ako želimo povezati  $v$  i  $\omega$  sa željenim brzinama  $u_1$  i  $u_2$ , jednostavnom matematikom se jednadžba (51) može preurediti u

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = R^{-1}(\theta) \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{l} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (52)$$

Ova transformacija unicycle modela omogućava nam korištenje linearnog modela zapisanog jednadžbom (48) kao osnovu za stvaranje upravljačkog sustava, umjesto kinematičkog modela mobilnog robota s diferencijalnim pogonom koji je dan jednadžbom (19).

## 4.2 Model s dva integratora (dinamika)

Dinamička jednadžba gibanja za  $i$ -ti robot glasi:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} v_i \cos(\theta_i) \\ v_i \sin(\theta_i) \\ \omega_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_i} & 0 \\ 0 & \frac{1}{J_i} \end{bmatrix} \begin{bmatrix} F_i \\ \tau_i \end{bmatrix}. \quad (53)$$

gdje su  $x_i$  i  $y_i$  pozicije  $i$ -tog robota u globalnom koordinatnom sustavu,  $\theta_i$  je orijentacija,  $v_i$  i  $\omega_i$  predstavljaju linearnu i kutnu brzinu,  $\tau_i$  je primijenjeni moment,  $F_i$  je primijenjena sila,  $m_i$  je masa, dok je  $J_i$  moment inercije.

Postupak izvoda modela je identičan kao i u poglavlju 4.1. Definiramo novu točku na robotu, točku  $P'$  koja se nalazi na udaljenosti  $l$  od točke  $P$  kao na Slici 15. Pozicija te nove točke dana je pomoću jednadžbi (42) i (43). Njihovim deriviranjem po vremenu dobiju se jednadžbe (44) i (45). Uvrštavanjem jednadžbe (21) u jednadžbe (44) i (45) dobiju se jednadžbe (46) i (47). Ponovnim deriviranjem tih jednadžbi dobiju se jednadžbe:

$$\ddot{x} = \dot{v} \cos(\theta) - v\omega \sin(\theta) - l\dot{\omega} \sin(\theta) - l\omega^2 \cos(\theta), \quad (54)$$

$$\ddot{y} = \dot{v} \sin(\theta) + v\omega \cos(\theta) + l\dot{\omega} \cos(\theta) - l\omega^2 \sin(\theta). \quad (55)$$

Uvođenjem jednadžbi  $\dot{v} = F/m$  i  $\dot{\omega} = \tau/J$  u jednadžbe (54) i (55) dobije se:

$$\ddot{x} = \frac{F}{m} \cos(\theta) - v\omega \sin(\theta) - l \frac{\tau}{J} \sin(\theta) - l\omega^2 \cos(\theta), \quad (56)$$

$$\ddot{y} = \frac{F}{m} \sin(\theta) + v\omega \cos(\theta) + l \frac{\tau}{J} \cos(\theta) - l\omega^2 \sin(\theta), \quad (57)$$

a to se preglednije može zapisati u matričnom obliku:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} -v\omega \sin(\theta) - l\omega^2 \cos(\theta) \\ v\omega \cos(\theta) - l\omega^2 \sin(\theta) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \cos(\theta) & -\frac{l}{J} \sin(\theta) \\ \frac{1}{m} \sin(\theta) & \frac{l}{J} \cos(\theta) \end{bmatrix} \begin{bmatrix} F \\ \tau \end{bmatrix}. \quad (58)$$

Sada uvodimo pretpostavku da točkom  $P'$  možemo izravno upravljati tako da

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (59)$$

kombinirajući jednažbe (58) i (59) dobijemo

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -v\omega \sin(\theta) - l\omega^2 \cos(\theta) \\ v\omega \cos(\theta) - l\omega^2 \sin(\theta) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \cos(\theta) & -\frac{l}{J} \sin(\theta) \\ \frac{1}{m} \sin(\theta) & \frac{l}{J} \cos(\theta) \end{bmatrix} \begin{bmatrix} F \\ \tau \end{bmatrix}. \quad (60)$$

Sređivanjem jednažbe (60) dobiju se izrazi za primijenjenu silu i primijenjeni moment:

$$F = mu_1 \cos(\theta) + mu_2 \sin(\theta) + ml\omega^2, \quad (61)$$

$$\tau = -\frac{J}{l} u_1 \sin(\theta) + \frac{J}{l} u_2 \cos(\theta) - \frac{J}{l} v\omega, \quad (62)$$

ili zapisano u matricnom obliku:

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = R^{-1}(\theta) \begin{bmatrix} m & 0 \\ 0 & \frac{J}{l} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} -\frac{J}{l} v\omega \\ l\omega^2 \end{bmatrix}. \quad (63)$$

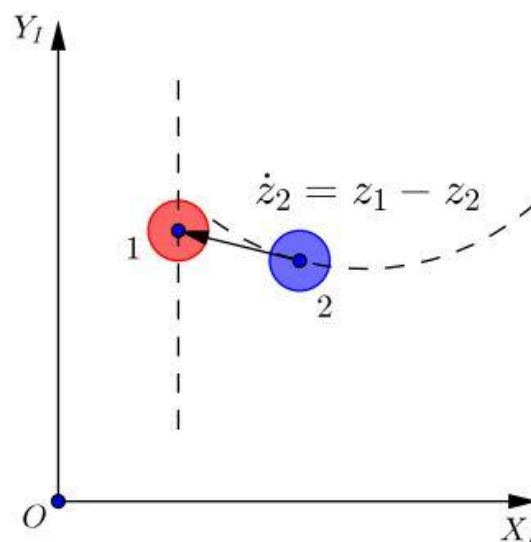
Ova transformacija *unicycle* modela omogućava nam korištenje linearnog modela zapisanog jednažbom (59) kao osnovu za stvaranje upravljačkog sustava, umjesto dinamičkog modela mobilnog robota s diferencijalnim pogonom koji je dan jednažbom (53).

## 5. FORMACIJE MOBILNIH ROBOTA

U ovom poglavlju ćemo pokazati neke tipične probleme gibanja mobilnih robota u formaciji te kako ih riješiti. Sve prikazane formacije te pripadajuće simulacije napravljene su u programu Matlab/Matlab Simulink, a za njihovu izradu koristili smo [7, 14, 17, 18]. Upravljački zakoni su distribuirani, što znači da nema robota koji je vođa za druge robote, nego su svi jednaki.

### 5.1 Praćenje

Početak ćemo s najjednostavnijim primjerom, a to je praćenje. Cilj praćenja je kako mu i ime kaže, kontinuirano pratiti gibajuću metu. Pretpostavimo da u ovom slučaju imamo dva robota, robota 1 koji se giba određenom brzinom po ravnoj liniji i robota 2 koji ga prati (Slika 16.). Kod praćenja vektor brzine robota 2 je uvijek usmjeren prema robotu 1. Trajektorija robota 2 se naziva trajektorija praćenja.



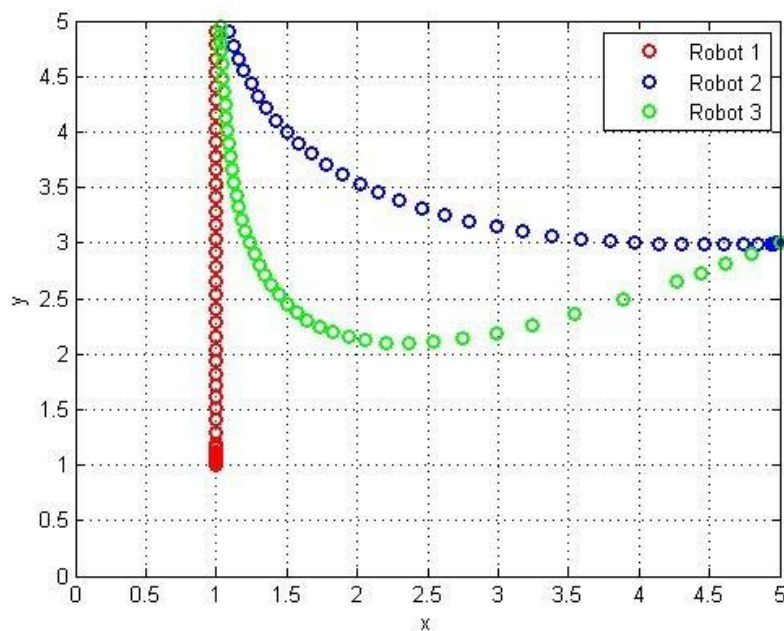
Slika 16. Praćenje [14]

Vektor  $z$  definira položaj robota  $i$  u ravnini

$$z_i = [x_i \quad y_i]^T. \quad (64)$$

### 5.1.1 Simulacija praćenja

Na Slici 17. prikazana je usporedba dvije simulacije. Robot 1 (na simulaciji označen crvenom bojom) je glavni robot, a njega roboti 2 (plavi) i 3 (zeleni) prate. Simulacijama je zajedničko to da roboti koji prate crvenog robota imaju informaciju o njegovoj poziciji, te konstantno ažuriraju tu poziciju. Robot 3 predstavljen je modelom s jednim integratorom, to jest opisan je kinematikom točkastog robota. Robot 2 predstavljen je modelom s dva integratora, odnosno opisan je dinamikom točkastog robota. Kako smo prije objasnili, kod točkastog robota se zanemaruje orijentacija, što nam omogućuje jednostavniju primjenu i lakšu implementaciju. Uz to, omogućeno nam je simuliranje dinamike bez poznavanja mase i inercije.



Slika 17. Praćenje usporedba – model točkastog robota

Upravljački zakon za zelenog robota glasi

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \quad (65)$$

ili napisano kompaktnije

$$\dot{z}_2 = z_1 - z_2. \quad (66)$$

Gibanje robota 1 definirano je u oba slučaja kao

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (67)$$

Upravljački zakon za plavog robota je nešto složeniji zato što su uz pozicije potrebne i brzine robota, pa on u općem slučaju glasi

$$\dot{z}_i = -a_{ij} [(z_i - z_j) + k(v_i - v_j)], \quad (68)$$

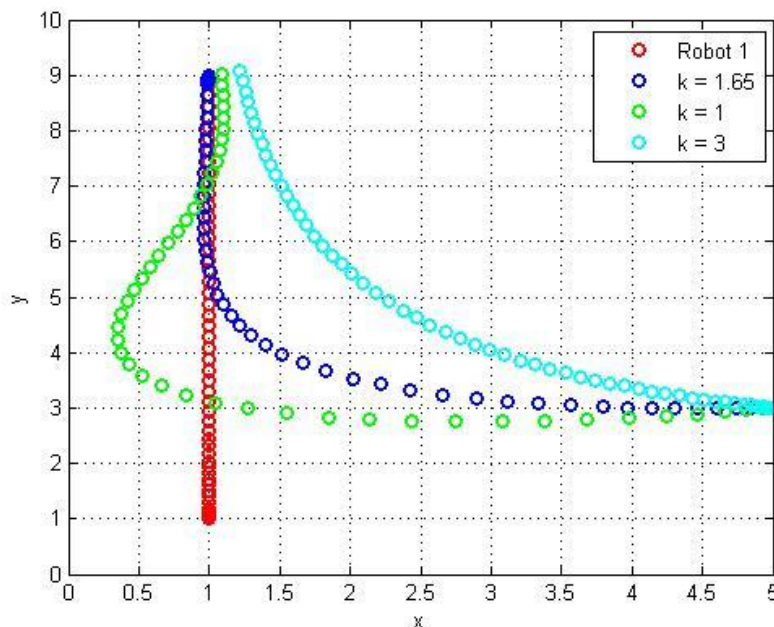
gdje  $a_{ij}$  predstavlja komponentu iz matrice susjedstva (u našem slučaju  $a_{ij}$  je postavljen na vrijednost 1), dok  $k$  predstavlja prigušenje sustava. Za naš slučaj jednačba (68) glasi

$$\dot{z}_2 = -[(z_2 - z_1) + k(v_2 - v_1)]. \quad (69)$$

Vektor  $v$  je definiran kao

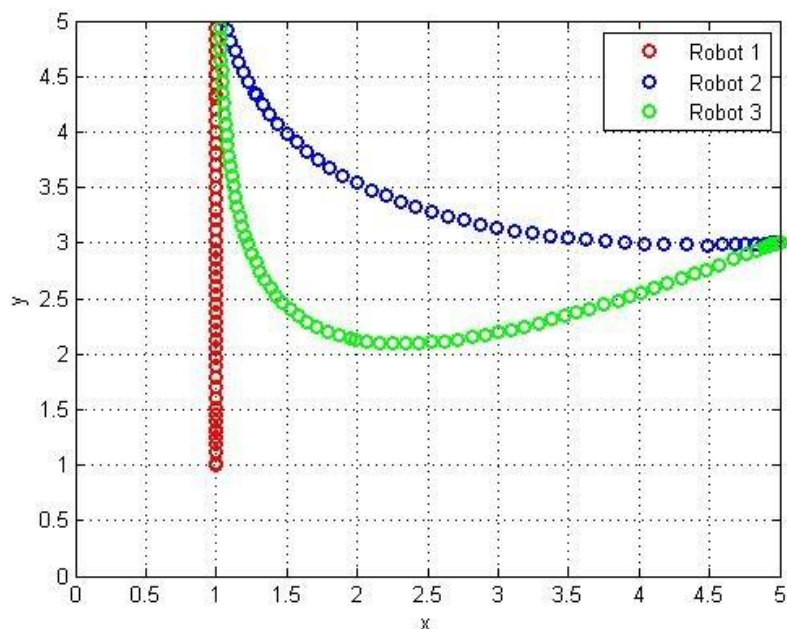
$$v_i = [\dot{x}_i \quad \dot{y}_i]^T. \quad (70)$$

Faktor  $k$  je važan faktor, jer on direktno utječe na ponašanje sustava. U našem slučaju njegova vrijednost je postavljena na 1.65. Slika 18. prikazuje trajektorije 3 robota s identičnim upravljačkim zakonom, uz jedinu razliku da im je faktor  $k$  različit.



Slika 18. Usporedba različitih vrijednosti faktora  $k$

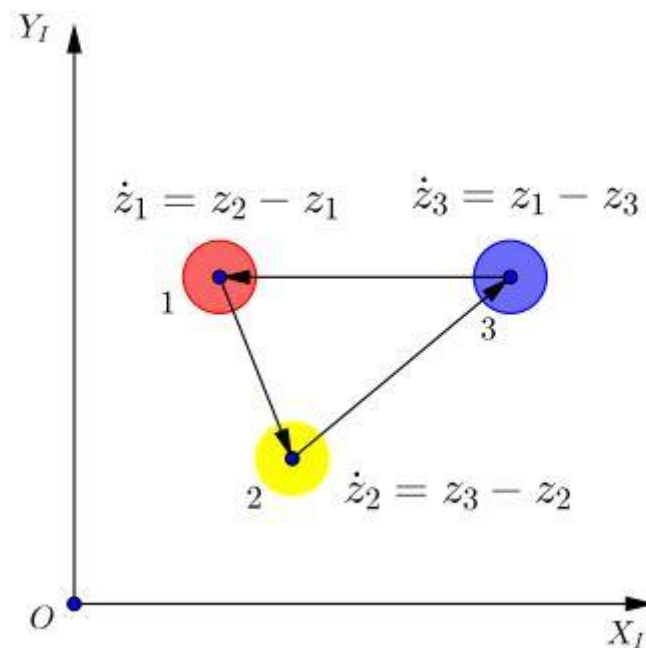
Na Slici 19. prikazana je usporedba rezultata dvije simulacije s modelom transformiranog *unicycle*-a. Kao i u prethodnom slučaju, robot 1 (crveni robot) je glavni, te ga roboti 2 (plavi robot) i 3 (zeleni robot) prate. I u ovom slučaju robot 3 predstavlja kinematiku mobilnog robota, a robot 2 njegovu dinamiku. Za razliku od simulacije sa Slike 17. gdje su korišteni modeli točkastog robota, ovdje je korišten model transformiranog *unicycle*-a, koji omogućava realniji prikaz robota. Model sadrži orijentacije, mase, inercije, sile i momente, te je njegova implementacija zbog toga nešto složenija. Upravljački zakoni po kojima se roboti gibaju su identični kao i za simulaciju sa Slike 17. Svi početni uvjeti i svi faktori i parametri su jednaki kao i u prošloj simulaciji uz dodatak da je masa robota 1 kg, dok je inercija  $0.25 \text{ kgm}^2$ .



Slika 19. Usporedba rezultata praćenja za *unicycle* model

## 5.2 Kružno praćenje

U ovom dijelu ćemo proširiti praćenje iz dijela 5.1. Kružno praćenje, kako mu ime kaže, znači da svaki robot prati jednog od susjednih robota, te na kraju zadnji robot prati prvoga, čime je zatvoren krug. Kružno praćenje završava kada se svi roboti susretnu u jednoj točki (u praksi je to nemoguće zbog toga što bi došlo do sudara, ali je u simulacijama moguće). Na Slici 20. prikazan je primjer s 3 robota koji se kružno prate, a to praćenje je postignuto kad robot 1 prati robota 2, robot 2 prati robota 3, a robot 3 prati robota 1.



Slika 20. Kružno praćenje [14]

U kružnom praćenju, točka u kojoj se svi roboti susretnu naziva se centroidom i ona je nepomična, a svi roboti joj se asimptotski približavaju. Ta točka je stvar međusobnog dogovora između svih robota koji sudjeluju u formaciji.

Trajektorije se mobilnih robota u kružnom praćenju mogu presjecati ili ne, a to ovisi o početnim pozicijama robota i o centroidi prema kojoj konvergiraju. Upravljački zakon koji općenito vrijedi za kinematiku kružnog praćenja glasi

$$\dot{z} = Mz, \quad (71)$$

gdje je  $z$  definiran kao

$$z_i = [x_i \quad y_i]^T. \quad (72)$$



U našem slučaju s 3 robota upravljački zakon za kinematiku robota glasi

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}. \quad (73)$$

Upravljački zakon za dinamiku kružnog praćenja je naravno malo složeniji, zato što robotu pored pozicije robota kojeg prati, treba i brzina tog robota. Općeniti upravljački zakon za dinamiku glasi

$$\dot{z}_i = -a_{ij} \left[ (z_i - z_j) + k(v_i - v_j) \right], \quad (68)$$

te je on, kao što se vidi, identičan onom za praćenje. Parametar  $a_{ij}$  predstavlja komponentu iz matrice susjedstva (u našem slučaju  $a_{ij}$  je postavljen na vrijednost 1), dok  $k$  predstavlja prigušenje sustava (za ovu simulaciju je postavljen na vrijednost 2), dok je  $v$  definiran kao

$$v_i = [\dot{x}_i \quad \dot{y}_i]^T. \quad (74)$$

### 5.2.1 Simulacija kružnog praćenja

Na Slici 21. prikazana je simulacija kinematike kružnog praćenja, dok je na Slici 22. prikazana simulacija dinamike kružnog praćenja koristeći model transformiranog *unicycle*-a. Zbog jednostavnosti su prikazane samo te dvije simulacije. Simulacije kinematike i dinamike modela točkastog robota nisu prikazane jer, kako smo vidjeli u poglavlju 5.1, one ne odstupaju značajno od simulacija kinematike i dinamike modela transformiranog *unicycle*-a. Simulacijama je zajedničko to da roboti imaju informaciju o poziciji robota kojeg prate, te konstantno ažuriraju tu poziciju. Roboti se gibaju na način kako je opisano u poglavlju 4. Za simulaciju sa Slike 21. upravljački zakon glasi:

$$\dot{z}_1 = z_2 - z_1, \quad (75)$$

$$\dot{z}_2 = z_3 - z_2, \quad (76)$$

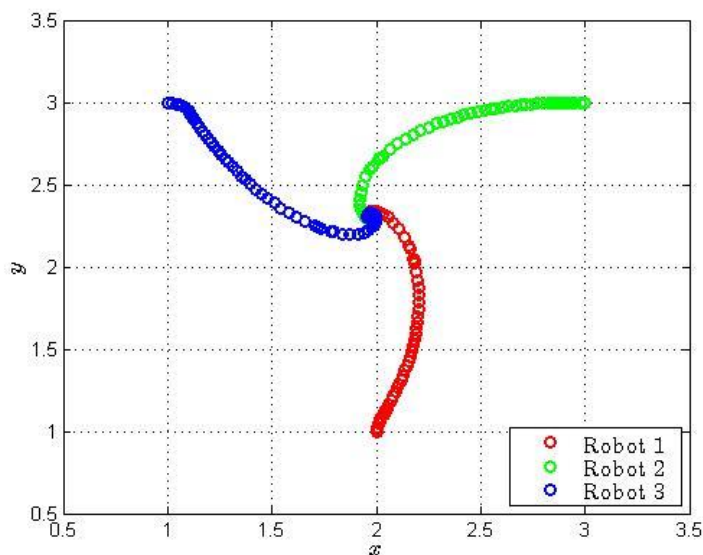
$$\dot{z}_3 = z_1 - z_3. \quad (77)$$

Za simulaciju sa Slike 22. upravljački zakon glasi:

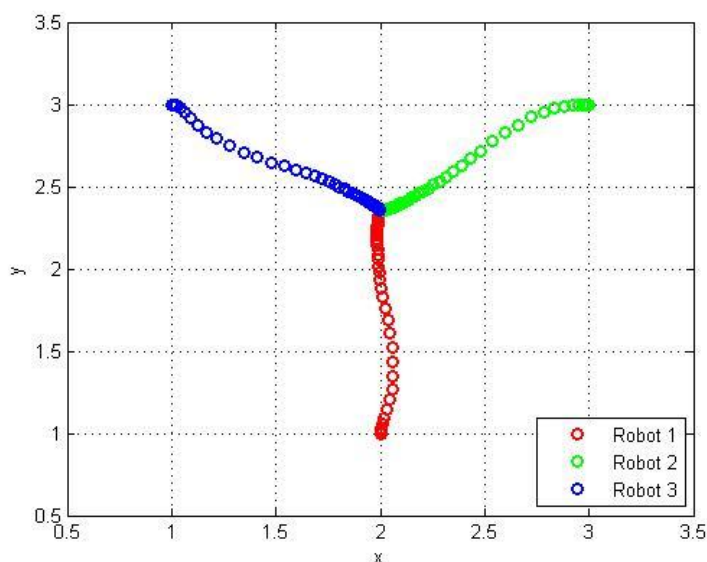
$$\dot{z}_1 = -[(z_1 - z_2) + k(v_1 - v_2)], \quad (78)$$

$$\dot{z}_2 = -[(z_2 - z_3) + k(v_2 - v_3)], \quad (79)$$

$$\dot{z}_3 = -[(z_3 - z_1) + k(v_3 - v_1)]. \quad (80)$$



Slika 21. Simulacija kinematike kružnog praćenja



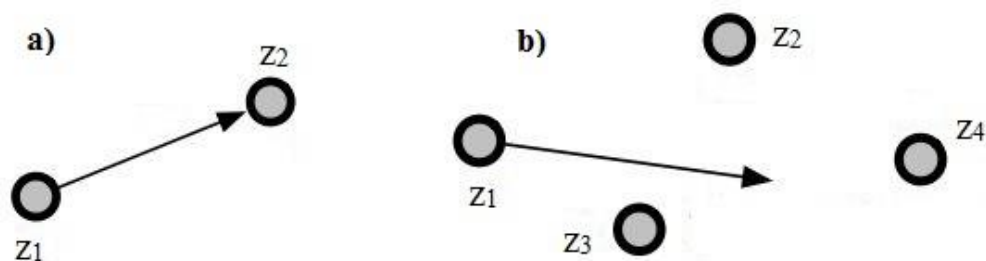
Slika 22. Simulacija dinamike kružnog praćenja

### 5.3 Susretanje (*rendezvous*)

Jednostavno rečeno, zadatak *rendezvous*-a je dovesti sve robote na zajedničku lokaciju. Da bi problem bio veći, robotima je onemogućeno komuniciranje, pa nema dogovora gdje će se susresti, nego se o točnoj lokaciji odlučuje dok se roboti gibaju.

Za simulaciju kinematike jedina informacija koju roboti imaju je relativna pozicija susjednih robota. To znači da robot 1 na lokaciji  $z_1$  ne može odrediti lokaciju robota 2 ( $z_2$ ), nego samo udaljenost među njima  $z_2 - z_1$  [17]. Kod simulacije dinamike nužno je uz relativne pozicije znati i relativne brzine susjednih robota. Na početku simulacije roboti se nalaze na svojim početnim pozicijama, dok se na kraju svi susretnu u jednoj točki. Točka susretnja ovisi o početnim pozicijama robota.

Razlika između kružnog praćenja i susretanja je u tome da robot kod kružnog praćenja poznaje samo poziciju (i brzinu kod simulacije dinamike) robota kojeg prati, dok lokacije ostalih robota ne poznaje, a kod susretanja robot poznaje relativne pozicije (i brzine kod simulacije dinamike) svih robota.



Slika 23. a) robot 1 cilja prema robotu 2; b) robot 1 cilja u svoje susjedstvo [17]

Ukoliko imamo dva robota (Slika 23. a)) i želimo da se oni susretnu, onda oni jednostavno idu jedan prema drugome. Ukoliko imamo više robota, tada je cilj da svaki robot cilja prema centru svoga susjedstva (Slika 23. b)) .

Općeniti upravljački zakon za kinematiku susretanja glasi

$$\dot{z} = \frac{1}{N_i} \sum_{j \in N_i} (z_j - z_i), \quad (81)$$

gdje  $N_i$  je broj susjeda robotu  $i$  [17].

Kod dinamike je upravljački zakon složeniji, te glasi

$$\dot{z}_i = -\frac{1}{N_i} \sum_{j \in N_i} a_{ij} [(z_i - z_j) + k(v_i - v_j)], \quad (82)$$

gdje su parametri  $a_{ij}$ ,  $k$  i  $v$  definirani u prijašnjim simulacijama (vrijednost parametra  $k$  je 1.65), dok je  $N_i$  broj susjeda robotu  $i$ .

### 5.3.1 Simulacija susretanja

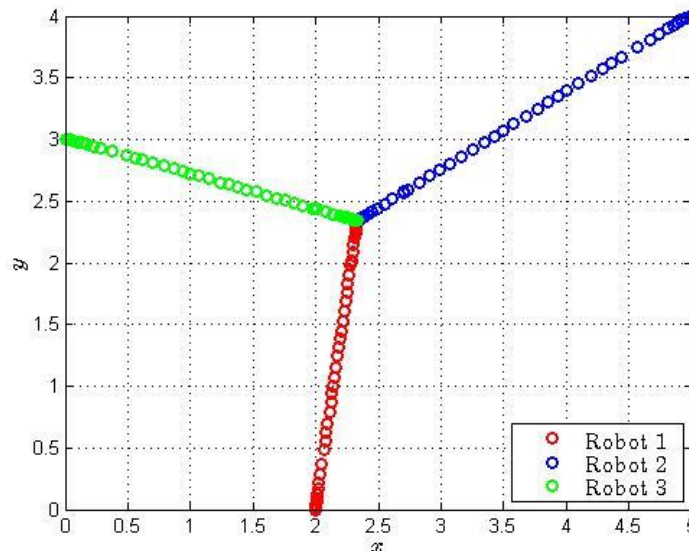
Zbog jednostavnosti prikaza, kao i u slučaju kružnog praćenja, ovdje ćemo prikazati samo simulacije kinematike i dinamike susretanja koje su napravljene koristeći model transformiranog *unicycle*-a. Na Slici 24. prikazan je rezultat simulacije kinematike susretanja, dok Slika 25. prikazuje rezultat simulacije dinamike susretanja.

Upravljački zakon za simulaciju kinematike modeliran je prema jednadžbi (81) te glasi

$$\dot{z}_1 = \frac{1}{2} ((z_2 - z_1) + (z_3 - z_1)), \quad (83)$$

$$\dot{z}_2 = \frac{1}{2} ((z_1 - z_2) + (z_3 - z_2)), \quad (84)$$

$$\dot{z}_3 = \frac{1}{2} ((z_1 - z_3) + (z_2 - z_3)). \quad (85)$$



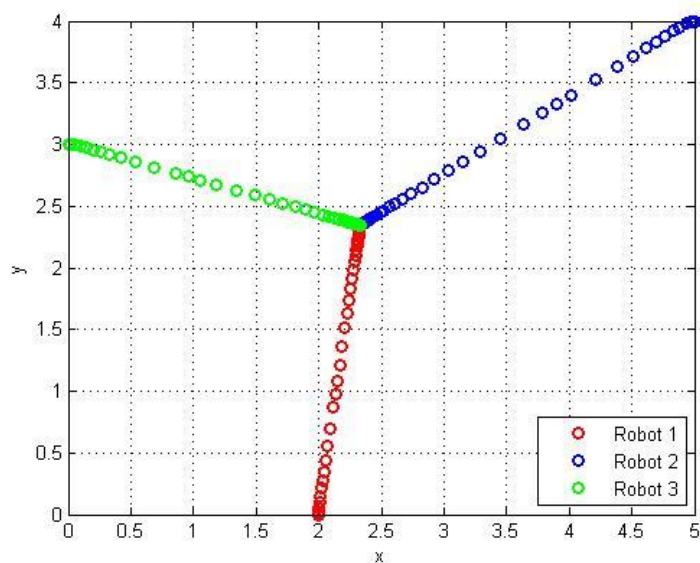
Slika 24. Susretanje kinematika

Upravljački zakon za simulaciju dinamike susretanja glasi

$$\dot{z}_1 = -\frac{1}{2} \left\{ [(z_1 - z_2) + k(v_1 - v_2)] + [(z_1 - z_3) + k(v_1 - v_3)] \right\}, \quad (86)$$

$$\dot{z}_2 = -\frac{1}{2} \left\{ [(z_2 - z_1) + k(v_2 - v_1)] + [(z_2 - z_3) + k(v_2 - v_3)] \right\}, \quad (87)$$

$$\dot{z}_3 = -\frac{1}{2} \left\{ [(z_3 - z_1) + k(v_3 - v_1)] + [(z_3 - z_2) + k(v_3 - v_2)] \right\}. \quad (88)$$



Slika 25. Susretanje dinamika

## 5.4 Uspostavljanje statičke formacije

U ovom odjeljku ćemo se baviti simulacijom čiji je rezultat da roboti uspostave formaciju koju smo im zadali. Ovo će u našem slučaju biti formacija trokuta. U odjeljku 5.2 vidjeli smo da kod kružnog praćenja svi roboti konvergiraju prema jednoj točki, zvanom centroida. Uspostavljanje formacije je ustvari samo proširenje kružnog praćenja u kojem roboti prate virtualne pozicije drugih robota. Na taj način možemo odrediti željene udaljenosti među robotima, a time i definirati formaciju koju želimo da oni uspostave. Također je važno napomenuti da ne možemo odrediti gdje će se formacija u ravnini uspostaviti, kao što nismo u kružnom praćenju mogli odrediti gdje će biti centroida prema kojoj svi roboti konvergiraju. Mjesto u kojem će formacija biti uspostavljena ovisi o početnim uvjetima. Simulacije uspostavljanja formacije su izrađene koristeći se rezultatima iz [18].

Općeniti upravljački zakon za kinematiku uspostavljanja formacije je

$$\dot{z}_i = - \sum_{j \in N_i} (z_i - z_j) - (\xi_i - \xi_j), \quad (89)$$

gdje su  $\xi_i$  i  $\xi_j$  željene pozicije robota,  $N_i$  je broj susjeda, dok je  $\dot{z}_i$  definirano isto kao i u prijašnjim simulacijama.

Općeniti upravljački zakon za dinamiku uspostavljanja formacije je složeniji, budući da su potrebne i brzine robota. Koristeći se s [18] i [7] upravljački zakon glasi

$$\dot{z}_i = - \sum_{j \in N_i} a_{ij} \left\{ [(z_i - z_j) - (\xi_i - \xi_j)] + k(v_i - v_j) \right\}, \quad (90)$$

gdje su  $\xi_i$  i  $\xi_j$  željene pozicije robota,  $N_i$  je broj susjeda,  $\dot{z}_i$  i  $v_i$  su definirani isto kao i u prijašnjim simulacijama, dok je  $k$  faktor prigušenja, koji je ovdje postavljen na vrijednost 1.65. Parametar  $a_{ij}$  predstavlja komponentu iz matrice susjedstva (u našem slučaju  $a_{ij}$  je postavljen na vrijednost 1).

### 5.4.1 Simulacije uspostavljanja formacije

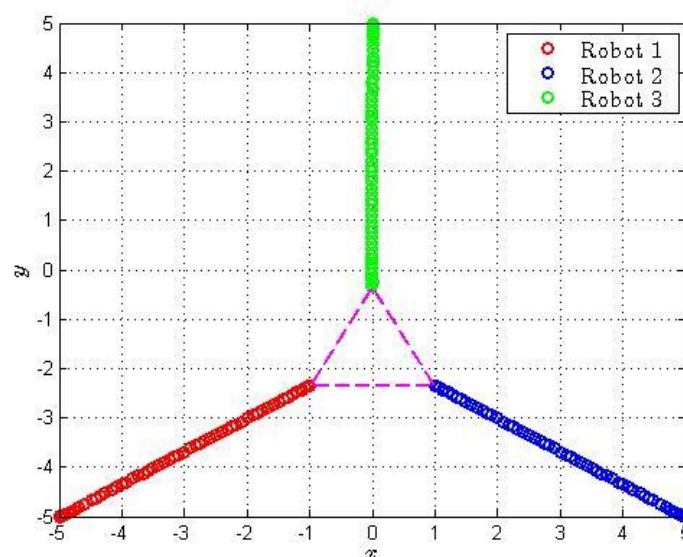
Zbog jednostavnosti prikaza, kao i u slučaju kružnog praćenja, ovdje ćemo prikazati samo simulacije kinematike i dinamike susretanja koje su napravljene koristeći model transformiranog *unicycle*-a. Na Slici 26. prikazan je rezultat simulacije kinematike susretanja, dok Slika 27. prikazuje rezultat simulacije dinamike susretanja.

Upravljački zakon za simulaciju sa Slike 26. (kinematika) glasi

$$\dot{z}_1 = -[(z_1 - z_2) - (\xi_1 - \xi_2) + (z_1 - z_3) - (\xi_1 - \xi_3)], \quad (91)$$

$$\dot{z}_2 = -[(z_2 - z_1) - (\xi_2 - \xi_1) + (z_2 - z_3) - (\xi_2 - \xi_3)], \quad (92)$$

$$\dot{z}_3 = -[(z_3 - z_1) - (\xi_3 - \xi_1) + (z_3 - z_2) - (\xi_3 - \xi_2)]. \quad (93)$$



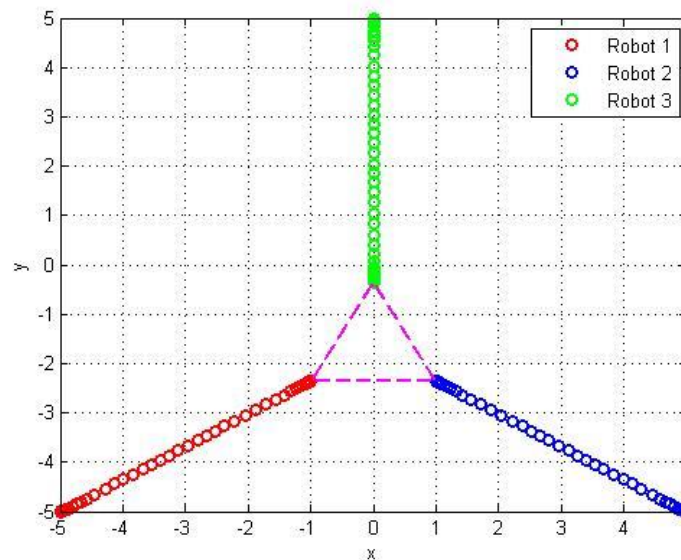
Slika 26. Simulacija uspostavljanja formacije (kinematika)

Upravljački zakon za simulaciju sa Slike 27. (dinamika) glasi

$$\dot{z}_1 = -\left\{[(z_1 - z_2) - (\xi_1 - \xi_2) + (z_1 - z_3) - (\xi_1 - \xi_3)] + k[(v_1 - v_2) + (v_1 - v_3)]\right\}, \quad (94)$$

$$\dot{z}_2 = -\left\{[(z_2 - z_1) - (\xi_2 - \xi_1) + (z_2 - z_3) - (\xi_2 - \xi_3)] + k[(v_2 - v_1) + (v_2 - v_3)]\right\}, \quad (95)$$

$$\dot{z}_3 = -\left\{[(z_3 - z_1) - (\xi_3 - \xi_1) + (z_3 - z_2) - (\xi_3 - \xi_2)] + k[(v_3 - v_1) + (v_3 - v_2)]\right\}. \quad (96)$$



Slika 27. Simulacija uspostavljanja formacije (dinamika)

## 5.5 Zaključci

Usporedbom simulacija sa Slike 17. i Slike 19. vidimo da su ostvarene gotovo identične trajektorije robota te da je pogreška između modela točkastog robota i modela transformiranog *unicycle* modela relativno mala. Zbog toga imamo slobodu odabira modela robota kojeg želimo koristiti. Model točkastog robota je jednostavnije implementirati, dok model transformiranog *unicycle*-a prikazuje stvarnije stanje stvari. Iako smo ovdje prikazali simulacije kružnog praćenja, susretanja i uspostavljanja formacije samo za model transformiranog *unicycle*-a, napominjemo da je to bila samo stvar osobnog odabira, budući da sve simulacije prikazuju vrlo slične rezultate, pa nije bilo potrebe prikazivanja svih izvedenih simulacija.



## 6. GIBANJE U FORMACIJI

U ovom ćemo poglavlju predstaviti dva upravljačka zakona koji omogućuju gibanje u formaciji. Na temelju ta dva zakona izvest ćemo prilagođene upravljačke zakone koji će odgovarati određenim grafovima komunikacije. Jedan skup upravljačkih zakona primijenit ćemo uzimajući u obzir dinamiku robota, dok ćemo u drugom skupu promatrati samo kinematiku gibanja u formaciji. Na temelju izvedenih upravljačkih zakona napraviti ćemo prikladne simulacije koje će uključivati ekstremne situacije, kad svi roboti komuniciraju sa svim robotima te kad većina robota prima informacije samo od nadređenog robota. Na kraju svakog odjeljka komentirat ćemo predstavljene upravljačke zakone te donijeti određene zaključke.

### 6.1 Linijsko gibanje u formaciji

Prvi upravljački zakon koji ćemo koristiti objašnjen je u [7, 19] gdje je predstavljen za tri robota. Mi ćemo taj upravljački zakon ovdje najprije ukratko predstaviti, a zatim ćemo objasniti kako ćemo njega promijeniti i zašto. Nakon toga ćemo prikazati niz simulacija izvedenih u programu Matlab, te ćemo komentirati prikazane simulacije.

#### 6.1.1 Upravljački zakon

Neka  $n$  predstavlja broj robota u formaciji. Formacijski uzorak (izgled formacije) je definiran kao skup

$$F = \{z_1^d, \dots, z_n^d\}, \quad (97)$$

gdje je  $z_i^d$  željena konstantna lokacija  $i$ -tog robota. U ovom slučaju postoje dva zadatka koja se međusobno suprotstavljaju. Prvi zadatak je odvesti sve robote u njihove željene krajnje pozicije, koje su definirane izgledom formacije (jednadžba (97)). Drugi zadatak je održati željenu formaciju za vrijeme putovanja od početnih do krajnjih pozicija.

Neka je  $E_g$  mjera pogreške između trenutne pozicije robota i željene krajnje pozicije definirana kao

$$E_g = \sum_{i=1}^n \tilde{z}_i^T K_g \tilde{z}_i, \quad (98)$$

gdje je  $K_g$  dijagonalna simetrična pozitivno-definitna matrica i  $\tilde{z}_i \triangleq z_i - z_i^d$ . Slično jednadžbi (98), definiramo  $E_f$  mjeru pogreške formacije kao

$$E_f = \sum_{i=1}^n (\tilde{z}_i - \tilde{z}_j)^T K_f (\tilde{z}_i - \tilde{z}_j), \quad (99)$$

gdje je  $K_f$  dijagonalna simetrična pozitivno-semidefinitna matrica. Održavajući  $E_f$  malim za vrijeme gibanja, roboti će izjednačiti svoje trenutne udaljenosti prema željenim udaljenostima koje su definirane izgledom formacije, što znači da će se roboti gibati u formaciji.

Ukupna mjera pogreške ostvarenog gibanja je suma  $E_g$  i  $E_f$  te je definirana kao

$$\begin{aligned} E(t) &= E_f(t) + E_g(t) \\ &= \sum_{i=1}^n \left[ \tilde{z}_i^T K_g \tilde{z}_i + (\tilde{z}_i - \tilde{z}_j)^T K_f (\tilde{z}_i - \tilde{z}_j) \right], \end{aligned} \quad (100)$$

gdje  $K_f$  i  $K_g$  mjere važnost održavanja formacije odnosno gibanja prema željenim krajnjim pozicijama. Cilj upravljanja je da  $E(t) \rightarrow 0$ , kako  $t \rightarrow \infty$ .

Upravljački zakon koji omogućuje postizanje ovog cilja glasi

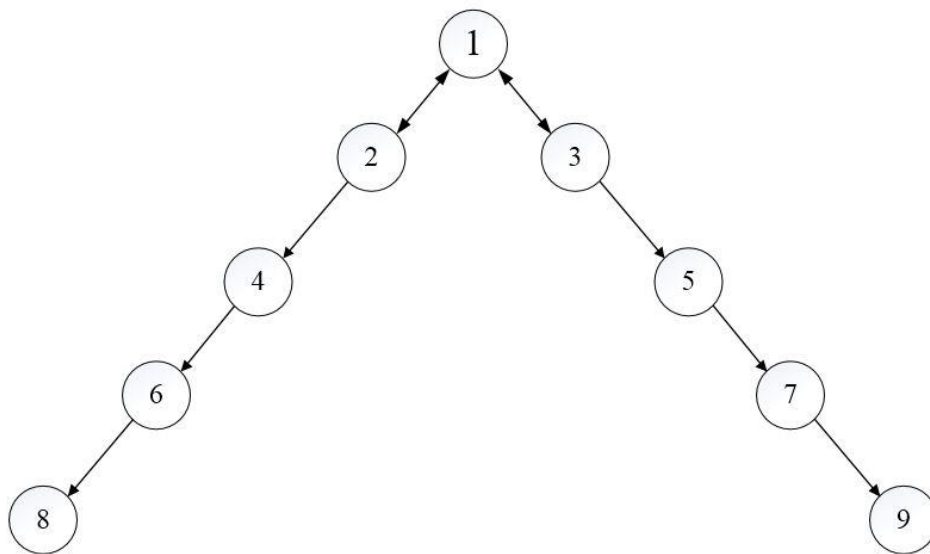
$$\begin{aligned} u_i &= -K_g \tilde{z}_i - D_g v_i \\ &\quad - \sum_{i=1}^n K_f (\tilde{z}_i - \tilde{z}_j) - D_f (v_i - v_j), \end{aligned} \quad (101)$$

gdje prva dva člana jednadžbe (101) vode robote prema željenim krajnjim pozicijama, dok treći član održava formaciju sa susjednim robotima.

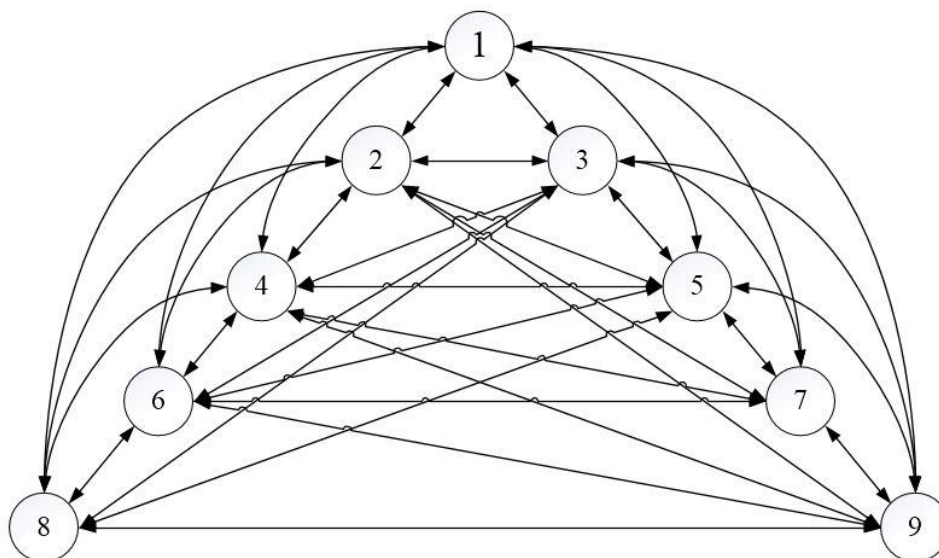
Iz jednadžbe (101) se može vidjeti da je ovo model s dva integratora, koji predstavlja dinamiku gibanja u formaciji (jednadžba (59)). U ovom upravljačkom zakonu potrebno je da robot poznaje relativne pozicije i relativne brzine svojih susjednih robota, a uz to i njihove željene krajnje pozicije.

### 6.1.2 Kinematika linijskog gibanja u formaciji

Prvo ćemo analizirati ponašanje robota u jednostavnijem slučaju, kada je korišten model s jednim integratorom. Promotrit ćemo dva slučaja, prvi kada je komunikacija između svih robota omogućena, što prikazuje Slika 29., a drugi kada roboti dobivaju informacije samo od svojih nadređenih kao što prikazuje Slika 28. Simulacije će biti napravljene za slučaj s 9 robota i za V formaciju, radi jednostavnosti prikaza.



Slika 28. Graf komunikacije za slučaj s 9 robota i najmanje komunikacije



Slika 29. Graf komunikacije za slučaj s 9 robota i potpunom komunikacijom

Koristeći jednadžbu (101) izvodimo prilagođeni upravljački zakon za kinematiku koji vrijedi u općenitom slučaju

$$u_i = -K_g \tilde{z}_i - K_f \sum_{j=1}^n (\tilde{z}_i - \tilde{z}_j), \quad (102)$$

gdje nam  $K_g$  i  $K_f$  mjere važnost dolaska do krajnjih pozicija i važnost održavanja formacije.

U našem slučaju su vrijednosti  $K_g$  i  $K_f$  odabrane na 0.5 i na 5, što znači da nam je važnije da se formacija uspostavi i održi, nego što nam je važno da formacija dođe do željenih točaka. Na Slici 30. prikazana je simulacija s 9 robota u kojoj je graf komunikacije jednak onom sa Slike 28., dok je na Slici 31. prikazana simulacija s 9 robota u kojoj je graf komunikacije jednak onom sa Slike 29. U obje simulacije postavljeni su identični početni uvjeti, identična pojačanja te je trajanje simulacije bilo identično.

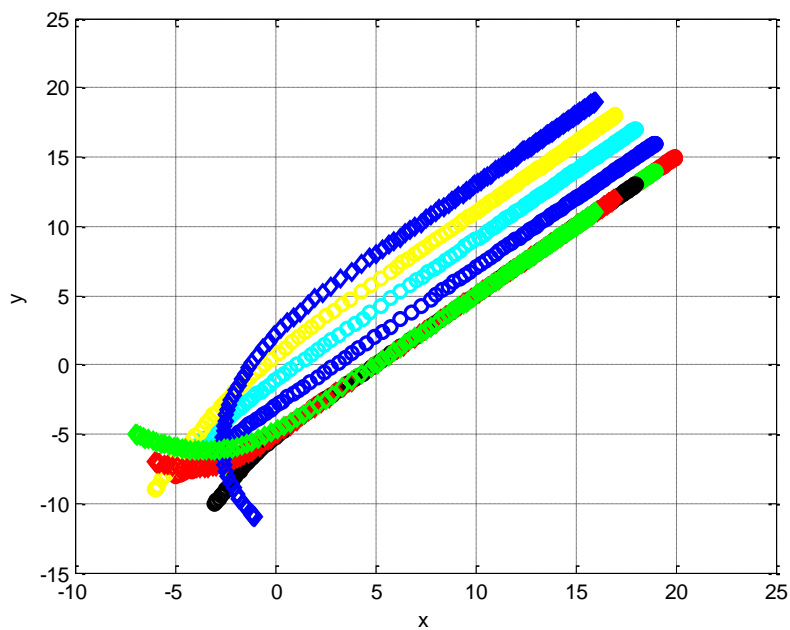
Za prvu simulaciju upravljački zakon za prvog robota jednak je onom na jednadžbi (102), dok za ostale robote on glasi

$$u_i = -K_g \tilde{z}_i - K_f (\tilde{z}_i - \tilde{z}_j). \quad (103)$$

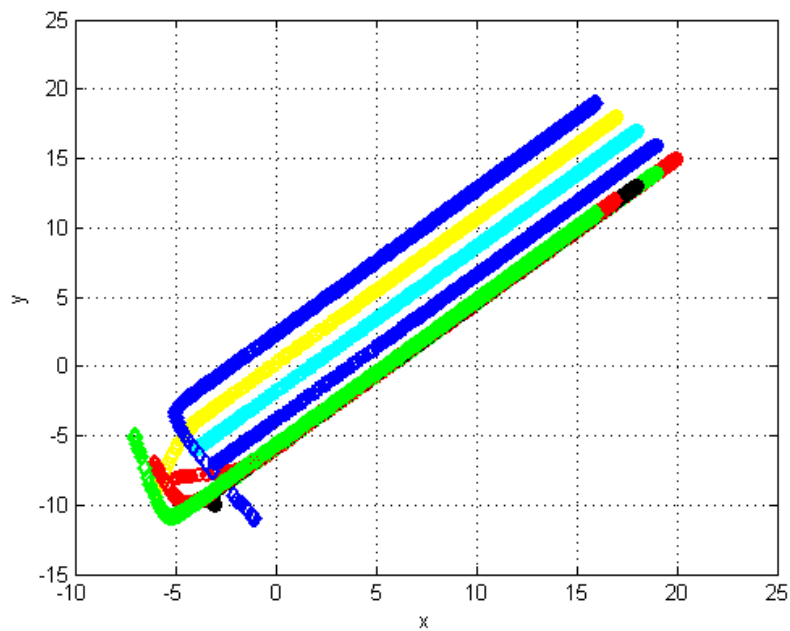
Za drugu simulaciju upravljački zakon je identičan onome prikazanom na jednadžbi (102).

Iz simulacija se jasno vidi da roboti konvergiraju prema krajnjim pozicijama te da se precizno pozicioniraju u njima. Također se vidi da je razlika između pokazane dvije simulacije relativno malena. Roboti na Slici 31. malo brže stanu u formaciju zbog puno više komunikacije među robotima. Budući da obje prikazane simulacije postignu formaciju i dođu do cilja, možemo zaključiti da nije nužno imati veliku količinu komunikacije među robotima da bi se zadatak (kinematsko gibanje u formaciji) izvršio.

Ovdje su prikazane samo dvije simulacije, za slučaj s 9 robota. Napravljene su također i simulacije s 5 i s 15 robota, koje potvrđuju ponašanje koje smo ovdje objasnili. Te dodatne simulacije u ovom radu nisu prikazane radi jednostavnosti i da se ne nagomilavaju simulacije koje pokazuju vrlo slične rezultate.



Slika 30. Kinematika V9 formacije s najmanje komunikacije

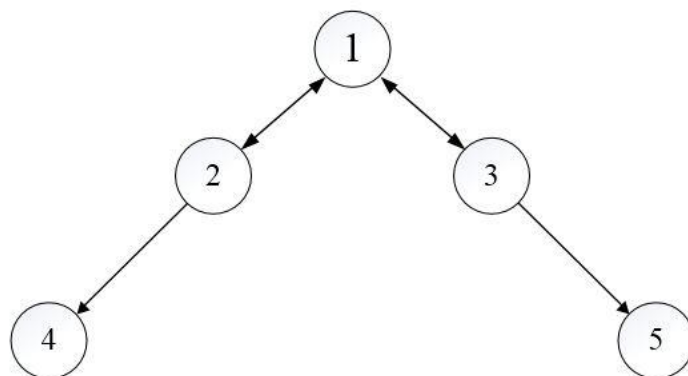


Slika 31. Kinematika V9 formacije s najviše komunikacije

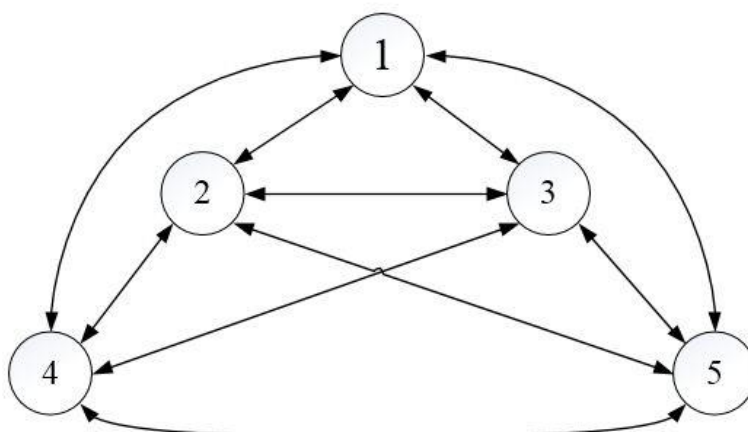
### 6.1.3 Dinamika linijskog gibanja u formaciji

U ovom ćemo odjeljku prikazati simulacije dinamike, dakle koristit ćemo model s dva integratora. Prikazat ćemo 3 skupine simulacija, s 5, s 9 i s 15 robota. Za svaku skupinu ćemo pokazati po dvije simulacije, jednu s najmanje komunikacije među robotima, a drugu s potpunom komunikacijom među svim robotima, uz to komentirat ćemo i razlike u odabiru određenih parametara te rezultate simulacija.

Grafovi komunikacije za 5 robota prikazani su na Slici 32. i Slici 33., a za slučaj s 9 robota su prikazani na Slici 28. i Slici 29.. Za slučaj s 15 robota grafovi komunikacije nisu posebno prikazani, ali se oni jasno mogu shvatiti iz Slike 28., 29., 32. i 33.



Slika 32. Graf komunikacije s 5 robota uz najmanje komunikacije



Slika 33. Graf komunikacije s 5 robota uz najviše komunikacije

Koristeći jednadžbu (101) izvodimo prilagođeni upravljački zakon za dinamiku koji vrijedi u općenitom slučaju

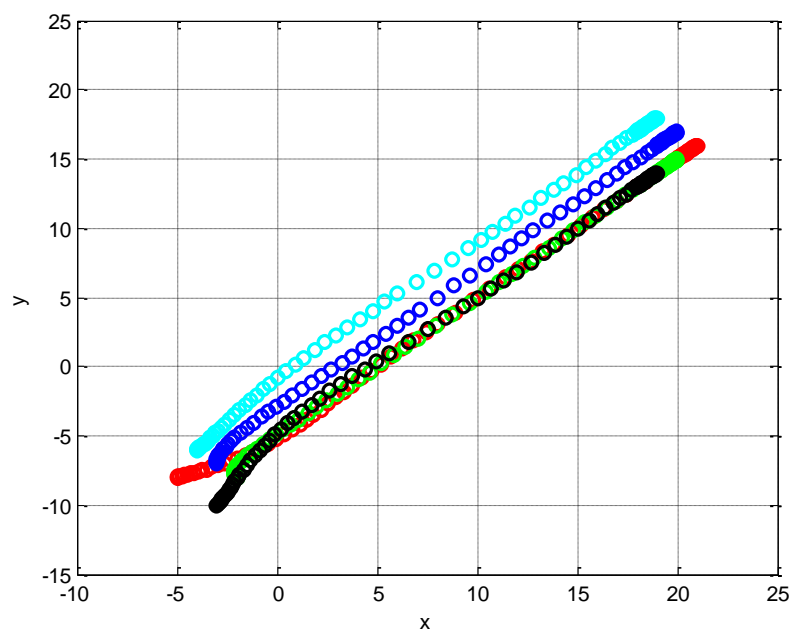
$$u_i = -K_g \tilde{z}_i - D_g v_i - K_f \sum_{j=1}^n (\tilde{z}_i - \tilde{z}_j) - D_f \sum_{j=1}^n (v_i - v_j), \quad (104)$$

gdje nam  $K_g, D_g$  i  $K_f, D_f$  mjere važnost dolaska do krajnjih pozicija i važnost održavanja formacije. U našem slučaju su vrijednosti  $K_g$  i  $K_f$  odabrane na 0.5 i na 5, dok su vrijednosti  $D_g$  i  $D_f$  odabrane na 1, što znači da nam je puno važnije da se formacija uspostavi i održi, nego što nam je važno da formacija dođe do željenih točaka. Za sve tri skupine simulacija postavljeni su identični početni uvjeti, identična pojačanja te je trajanje simulacije identično. Na Slici 34. prikazana je simulacija s 5 robota u kojoj je graf komunikacije jednak onom sa Slike 32., dok je na Slici 35. prikazana simulacija s 5 robota u kojoj je graf komunikacije jednak onom sa Slike 33.

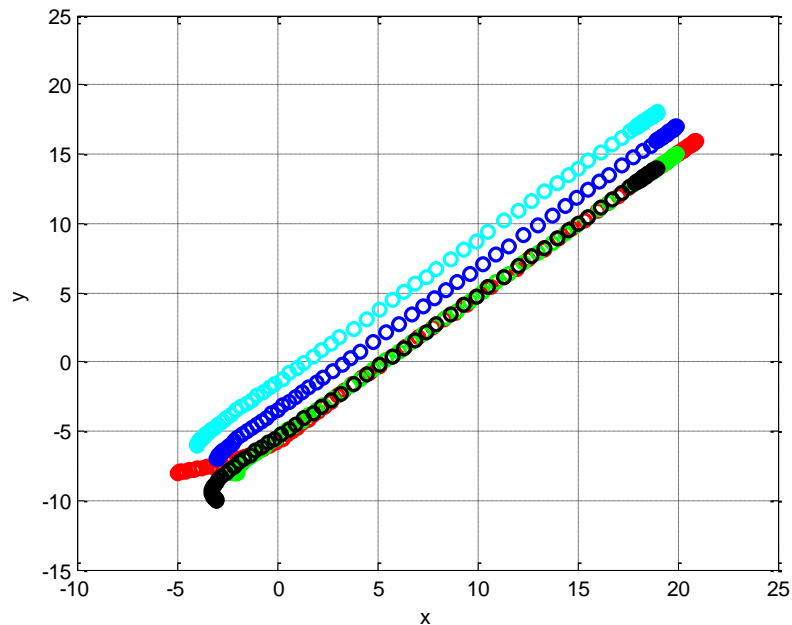
Za simulaciju s 5 robota u kojoj je graf komunikacije jednak onom sa Slike 32. upravljački zakon za prvog robota jednak je onom na jednadžbi (104) zato što robot 1 ima informaciju o relativnoj poziciji, brzini i željenim krajnjim točkama robota 2 i 3, tako da se robot 1 može smatrati na neki način vođom. Upravljački zakon za ostale robote glasi

$$u_i = -K_g \tilde{z}_i - D_g v_i - K_f (\tilde{z}_i - \tilde{z}_j) - D_f (v_i - v_j), \quad (105)$$

Za simulaciju s 5 robota u kojoj je graf komunikacije jednak onom sa Slike 33. upravljački zakon je identičan onome prikazanom na jednadžbi (104).



**Slika 34. Simulacija dinamike s 5 robota s najmanje komunikacije**



**Slika 35. Simulacija dinamike s 5 robota i najviše komunikacije**

Ove dvije simulacije pokazuju međusobno vrlo slične trajektorije gibanja robota, te slične onima koje smo pokazali u odjeljku 6.1.2. Razlika između ove dvije prikazane simulacije je u tome da se u simulaciji sa Slike 35. formacija uspostavi malo prije nego u slučaju simulacije sa Slike 34. Ove simulacije pokazuju da za manji broj robota i prikazane načine komunikacije nema velike razlike između ponašanja robota.

Simulacije s 9 i s 15 robota ćemo obraditi zajedno, zato što pokazuju slične rezultate. Za simulaciju s 9 robota u kojoj je graf komunikacije jednak onom sa Slike 28. upravljački zakon za prvog robota jednak je onom u jednadžbi (104) zato što prvi robot ima informaciju o relativnoj poziciji, brzini i željenim krajnjim točkama robota 2 i 3, tako da se robot 1 može smatrati na neki način vođom. Upravljački zakon za ostale robote je jednak onome definiranom jednadžbom (105) i glasi

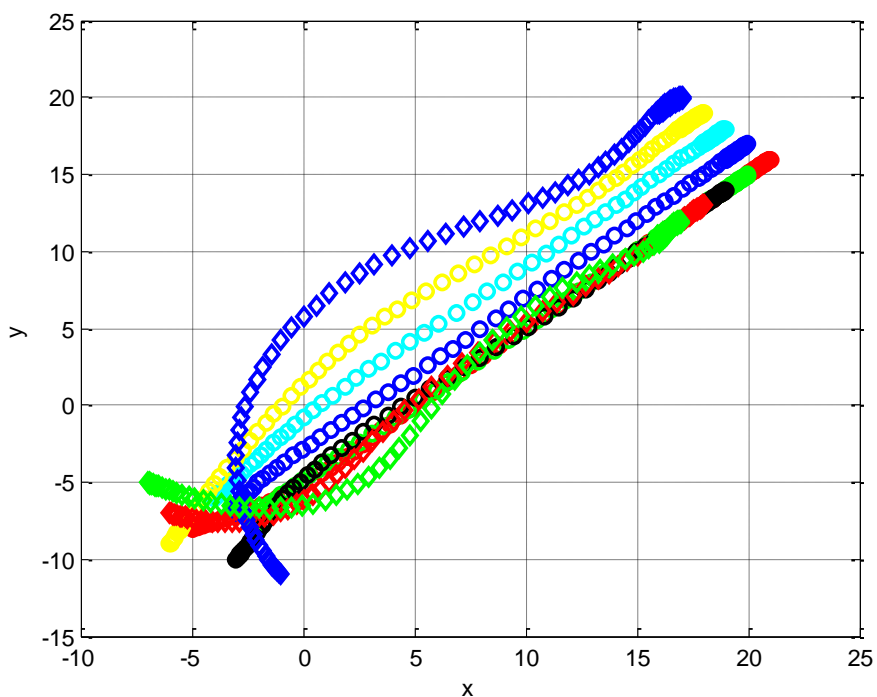
$$u_i = -K_g \tilde{z}_i - D_g v_i - K_f (\tilde{z}_i - \tilde{z}_j) - D_f (v_i - v_j), \quad (105)$$

Za simulaciju s 9 robota u kojoj je graf komunikacije jednak onom sa Slike 29. upravljački zakon je identičan onome prikazanom u jednadžbi (104).

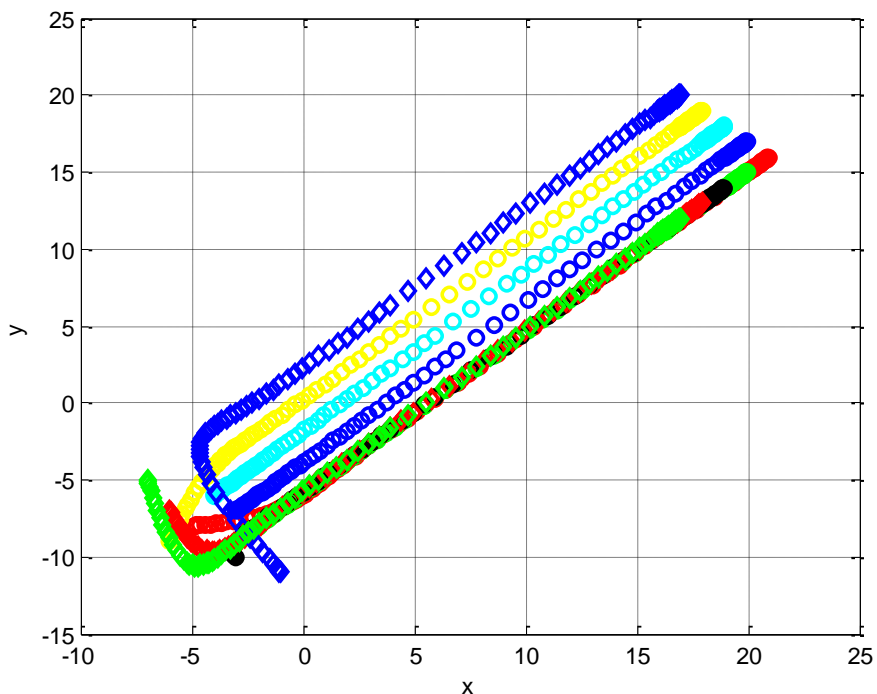
Za simulaciju s 15 robota i najmanje komunikacije, upravljački zakon za prvog robota jednak je upravljačkom zakonu iz jednadžbe (104) dok je za ostale on jednak onom prikazanom



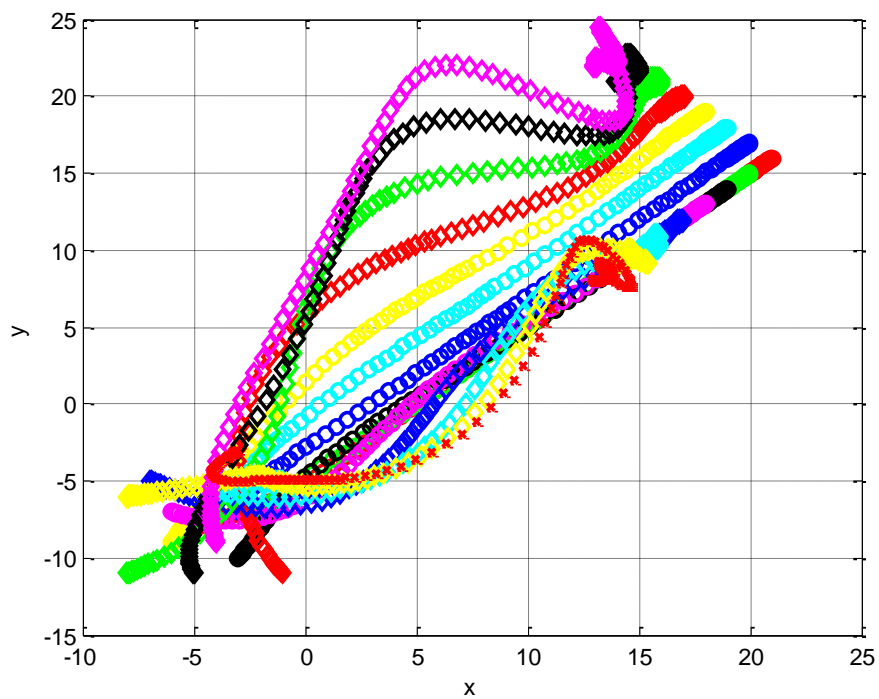
jednadžbom (105). Za simulaciju s 15 robota i najviše komunikacije, upravljački zakon za sve robote jednak je onom definiranom jednadžbom (104).



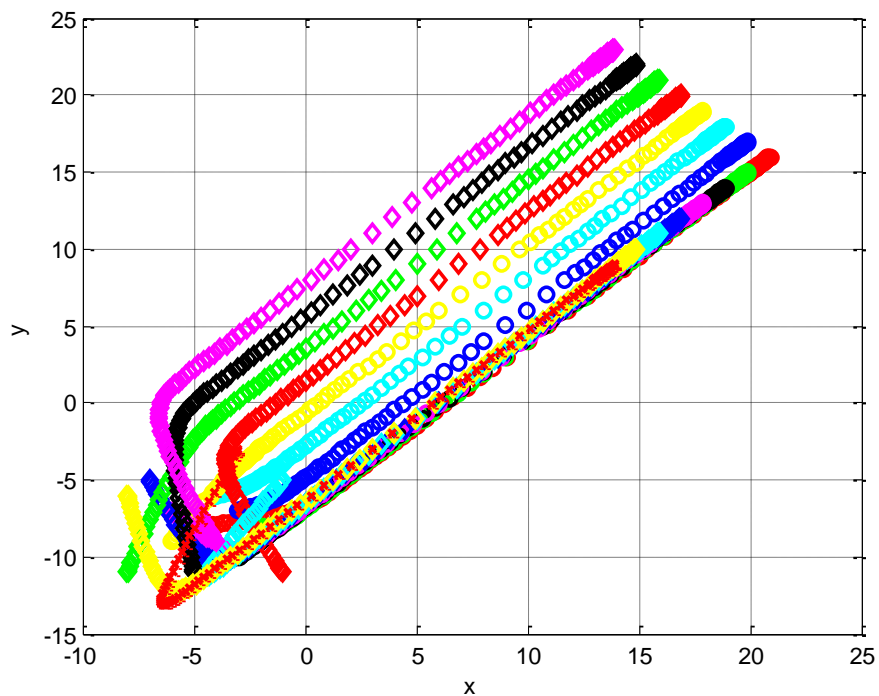
Slika 36. Simulacija dinamike s 9 robota i najmanje komunikacije



Slika 37. Simulacija dinamike s 9 robota i najviše komunikacije



Slika 38. Simulacija dinamike s 15 robota i najmanje komunikacije



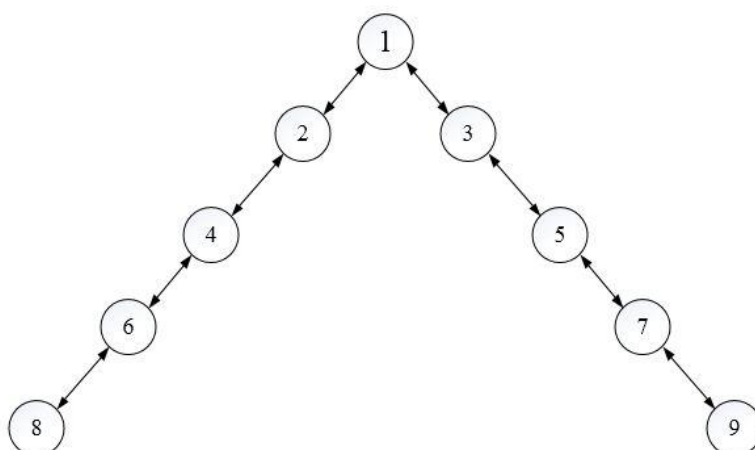
Slika 39. Simulacija dinamike s 15 robota i najviše komunikacije

Iz simulacija prikazanih Slikom 36. i 38., kod kojih je komunikacija između robota održana na minimumu, vidljivo je da povećanjem broja robota formacija nije uspostavljena za vrijeme gibanja. Takvi rezultati simulacije nam pokazuju da ovakav upravljački zakon i ovakav graf komunikacije ne zadovoljavaju, tako da primjena tog grafa komunikacije i upravljačkog sustava nije preporučljiva. Simulacije prikazane Slikama 37. i 39., kod kojih je komunikacija među robotima na najvišoj razini, pokazuju da ovakav način upravljanja zadovoljava, jer omogućuje brzo uspostavljanje formacije, njeno održavanje i konvergenciju željenim krajnjim pozicijama.

### 6.1.4 Komentar

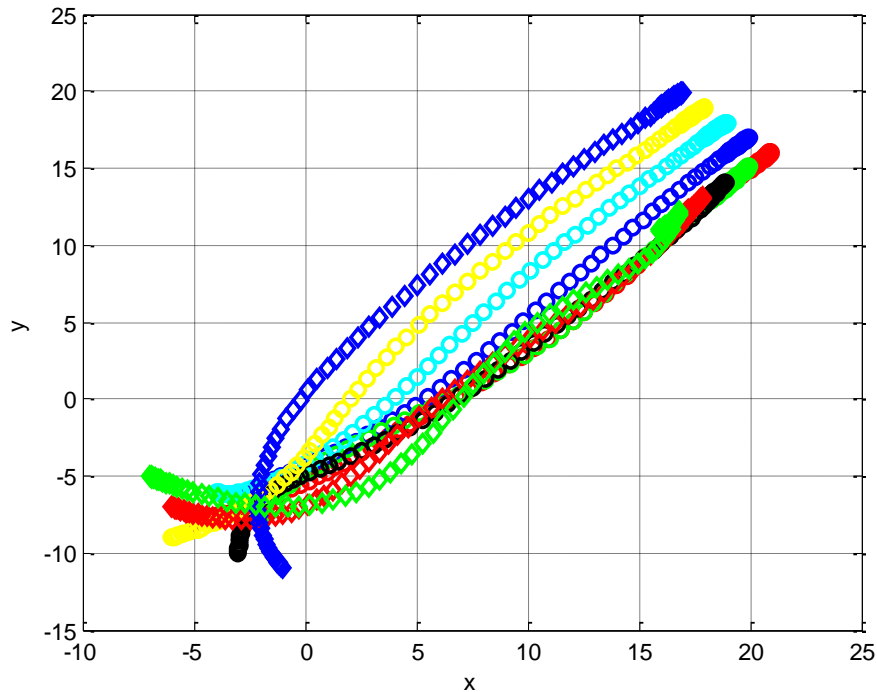
Iako se na prvi pogled upravljački sustav u simulacijama s potpunom komunikacijom čini poželjnijim, to ponekad nije tako. Takav način komunikacije je problematičan kod većeg broja robota, zato što u jednom trenutku svi roboti šalju svim drugim robotima svoja stanja te od svih drugih robota primaju njihova stanja, a tolika razina komunikacije izaziva komunikacijska zagušenja. Također je ograničavajuće to što svi roboti moraju biti u komunikacijskom dosegu svih drugih robota, što nije praktično, a ponekad (kod velikog broja robota) može biti i neizvedivo.

Prihvatljivije rješenje je djelomično povećanje razine komunikacije, kao na primjer na Slici 40. koja je prikazana za slučaj s 9 robota.



**Slika 40.** Graf komunikacije s 9 robota i malo većom razinom komunikacije

Ovakvim načinom povećanja komunikacije, kada svi roboti vide svoja dva najbliža susjeda, osim robota 8 i 9, poboljšava se konvergencija željenim krajnjim točkama i bolja je uspostava i gibanje u formaciji (Slika 41.).

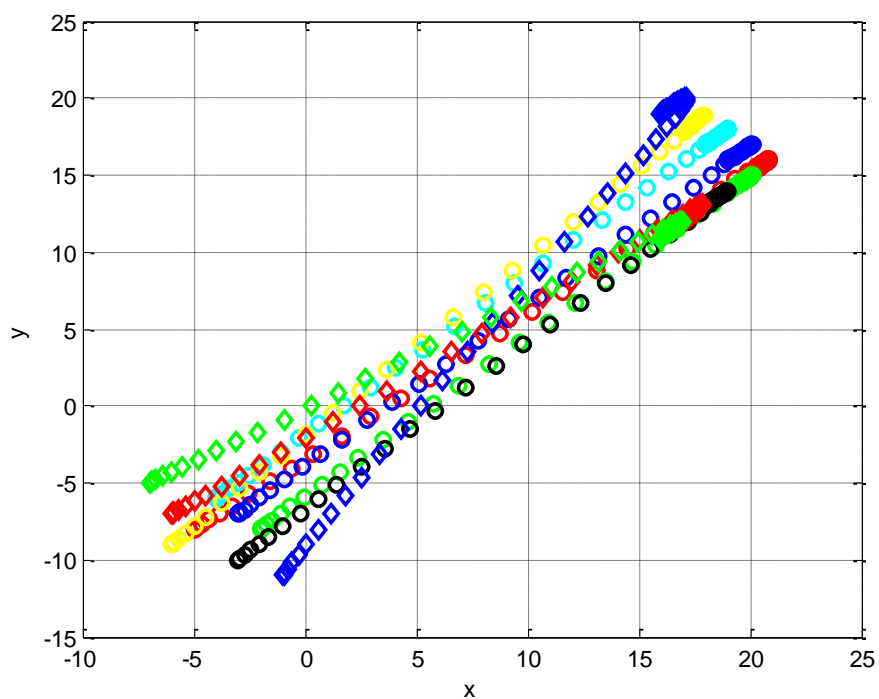


**Slika 41. Poboljšanje očuvanosti formacije i konvergencije željenim krajnjim točkama**

U usporedbi sa Slikom 36., kod Slike 41. manje je "rasipanje" robota, iako se ono i dalje javlja. Ovime smo pokazali da se povećanjem razine komunikacije među robotima povećava konvergencija željenim krajnjim pozicijama, te je oblik formacije postojaniji.

Za kraj ćemo pokazati ovisnost izgleda formacije o vrijednostima parametara  $K_g, D_g$  i  $K_f, D_f$ . U prethodno prikazanim simulacijama su vrijednosti  $K_g$  i  $K_f$  odabrane na 0.5 i na 5, dok su vrijednosti  $D_g$  i  $D_f$  odabrane na 1, što je značilo da je važnije uspostaviti formaciju, a da je manje važno doći do željenih krajnjih pozicija.

Sada ćemo prikazati simulaciju dinamike s 9 robota u kojoj je važnije da roboti dođu do svojih krajnjih pozicija, dok uspostavljanje formacije nije bitno. Vrijednosti  $K_g$  i  $K_f$  odabrane na 2 i na 0.5, dok su vrijednosti  $D_g$  i  $D_f$  odabrane na 2 i na 1.



**Slika 42. Dinamika s 9 robota u kojoj je važnije doći do željenih krajnjih pozicija**

Budući da je cilj da se roboti gibaju u formaciji, ovakvo je ponašanje neprihvatljivo, zato što se formacija uspostavi samo na kraju, kad roboti dođu u svoje krajnje pozicije, a to odgovara poglavlju 5.4 gdje je zadatak bio da roboti stanu u formaciju.

## 6.2 Gibanje formacije po krivulji

U ovom poglavlju opisat ćemo i prikazati upravljački zakon koji omogućava gibanje robota u formaciji. Za razliku od odjeljka 6.1, ovdje će se roboti gibati po proizvoljno definiranoj krivulji. U prvom dijelu ćemo prikazati simulacije prema [20] te ukazati na neke nedostatke tamo predloženih rješenja. Nakon toga ćemo prikazati modificiranu verziju upravljačkog zakona za kinematsko gibanje i prikladne simulacije, te upravljački zakon za dinamiku gibanja i prikladne simulacije. Na kraju poglavlja ćemo dati kratki komentar o prikazanim upravljačkim zakonima te prikazanim simulacijama.

### 6.2.1 Gibanje formacije po krivulji

Ovdje predstavljen upravljački zakon predložen je u radu [20]. On garantira da će stanja robota konvergirati nekoj željenoj vrijednosti, to jest  $z_i(t) - z_j(t) \rightarrow \delta_{ij}(t)$ , gdje  $\delta_{ij}(t)$  označava željeni (ovisan o vremenu) razmak između  $z_i(t)$  i  $z_j(t)$ . Predloženi upravljački zakon glasi

$$u_i = \dot{\delta}_i - \sum_{j=1}^n g_{ij} k_{ij} \left[ (z_i - z_j) - (\delta_i - \delta_j) \right], \quad (106)$$

gdje su  $z_i$  i  $z_j$  pozicije robota  $i$  i  $j$ ,  $\delta_i - \delta_j$  predstavlja razmak između pozicija referentnih krivulja u nekom trenutku  $t$  ( $\delta$  predstavlja referentnu trajektoriju),  $g_{ij} = 1$  ako postoji komunikacija između robota  $i$  i robota  $j$ , a ako ne postoji onda je  $g_{ij} = 0$ , dok je  $k_{ij}$  pojačanje. Iz jednadžbe (106) se može vidjeti da se ovdje radi o upravljačkom zakonu s jednim integratorom, to jest radi se o kinematičkom modelu.

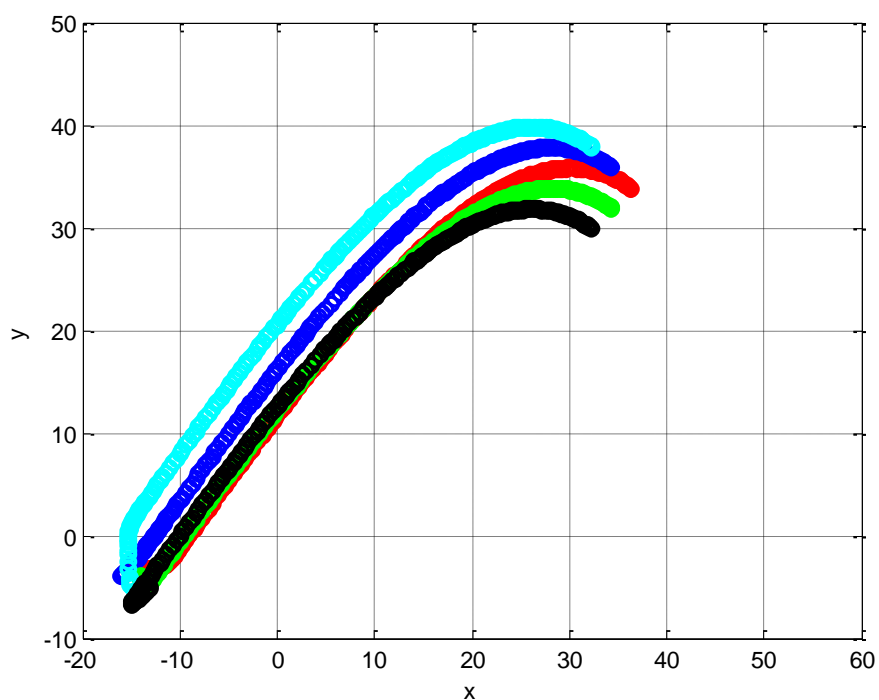
Krivulja (referentna trajektorija) koju ćemo koristiti u svim simulacijama definirana je s

$$\delta(t) = \left[ 60 \sin(\pi t / 100), 40 \sin(\pi t / 50) \right]^T. \quad (107)$$

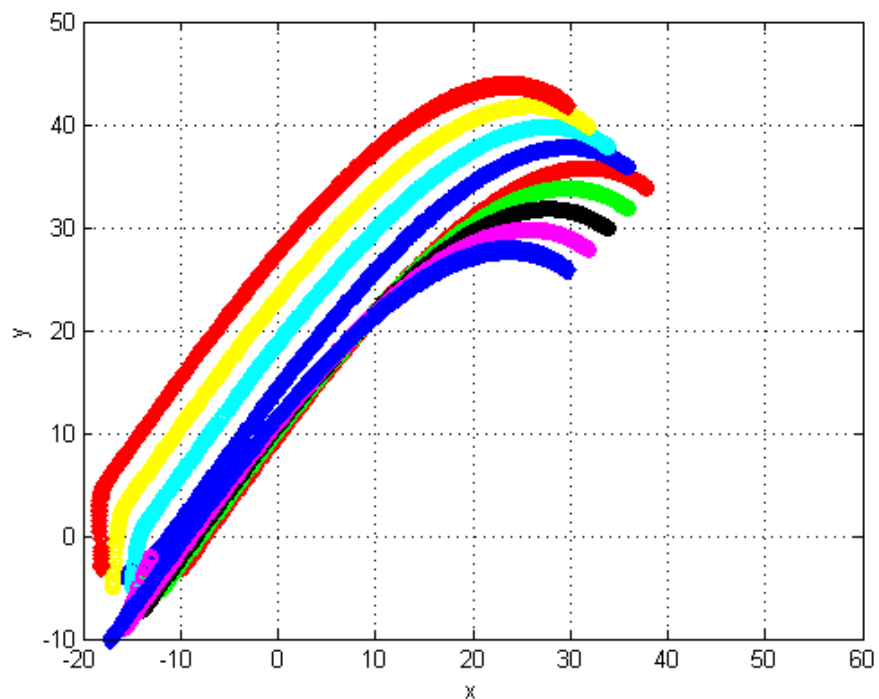
Svi roboti koriste identične krivulje za gibanje, samo su one pomaknute na određene udaljenosti od one od prvog (glavnog) robota, tako da pomoću njih definiramo izgled formacije.

U nastavku ćemo prikazati rezultate simulacija koristeći upravljački zakon definiran jednadžbom (106) za 5 i za 9 robota. Grafovi komunikacije su definirani na Slikama 29. i 33.

Opet ćemo koristiti V formaciju, zbog jasnoće prikaza. Vrijednosti parametara  $k_{ij}$  i  $g_{ij}$  su 1, dok je trajanje simulacije 30 sekundi.

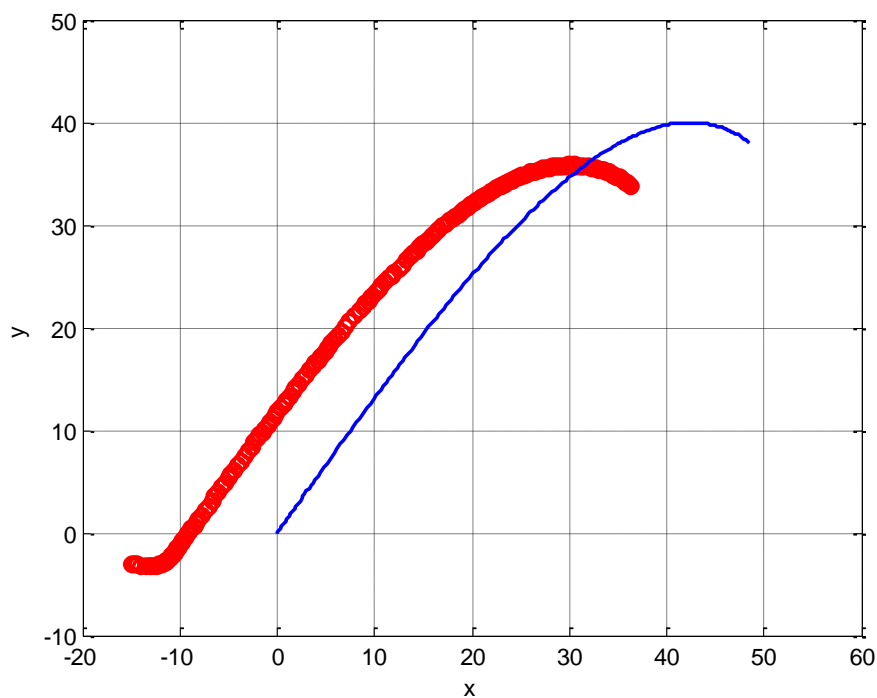


Slika 43. Simulacija s 5 robota i za upravljački zakon definiran jednačbom (106)



Slika 44. Simulacija s 9 robota i za upravljački zakon definiran jednačbom (106)

Iako na prvi pogled obje simulacije pokazuju ispravnost korištenog upravljačkog zakona, budući da je V formacija uspostavljena i održana za vrijeme gibanja, upravljački zakon ipak ima značajne nedostatke. Naime, iako roboti izvrše glavni zadatak, a to je gibanje u formaciji po krivulji, oni ne prate u potpunosti zadanu krivulju. To se najbolje vidi kod glavnog robota (vodećeg) koji ne dolazi do željene udaljenosti (40) na y osi, a isto tako ne dolazi ni do željene udaljenosti (60) na x osi. Odstupanje je prikazano na Slici 45.



**Slika 45. Odstupanje putanje robota od željene trajektorije upravljačkog zakona prikazanog jednačbom (106)**

Ovakvo odstupanje može uzrokovati probleme kod primjena gdje se traži veća točnost gibanja, kao na primjer gibanje kroz uske prolaze ili kroz hodnike zgrada. U takvim situacijama ovaj upravljački zakon nije prikladan.

U sljedećem odjeljku ćemo predložiti modificirani upravljački zakon te ćemo prikazati prikladne simulacije.



## 6.2.2 Kinematika gibanja formacije po krivulji

U ovom ćemo odjeljku ponuditi modificiranu verziju upravljačkog zakona definiranog jednačbom (106), koji će omogućiti precizno praćenje željene trajektorije, a uz to će i formacija biti održana.

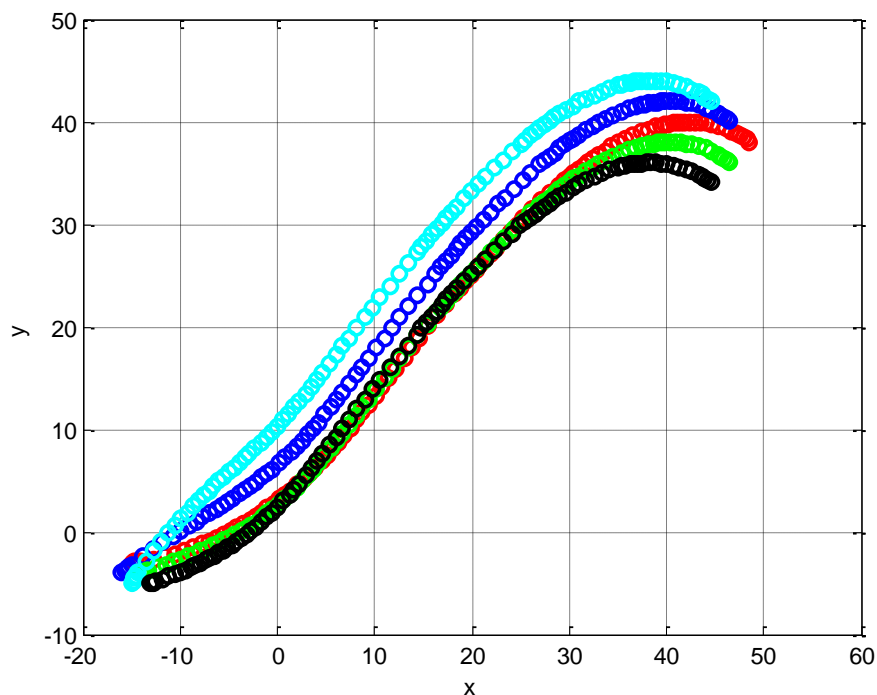
Predloženi unaprijeđeni upravljački zakon glasi

$$u_i = \dot{\delta}_i - \sigma(z_i - \delta_i) - \sum_{j=1}^n g_{ij} k_{ij} \left[ (z_i - z_j) - (\delta_i - \delta_j) \right], \quad (108)$$

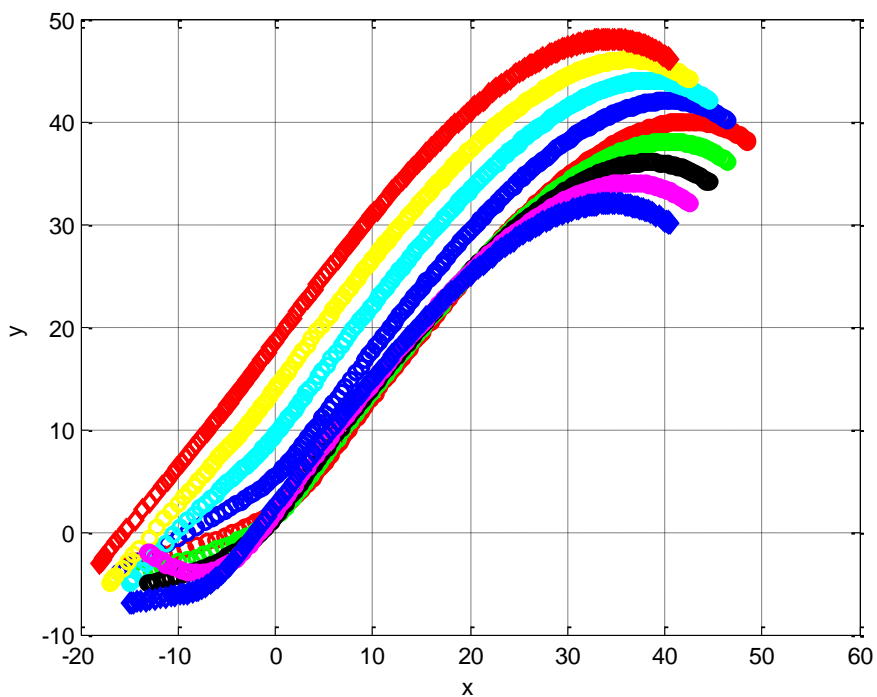
gdje je dodan član  $\sigma(z_i - \delta_i)$  koji povećava važnost praćenja željene trajektorije.  $\sigma$  je pojačanje i ono je postavljeno na vrijednost 2, dok su svi ostali parametri definirani u prethodnom odjeljku.

Prikazat ćemo simulacije s 5 i s 9 robota, za grafove komunikacije koji su definirani na Slikama 28.,29.,32. i 33., što znači da će biti dvije skupine simulacija, jedne za najmanje komunikacije među robotima, a druge uz potpunu komunikaciju između robota. Opet ćemo koristiti V formaciju, radi jednostavnosti prikaza. Svim će simulacijama biti zadani isti početni uvjeti, te će sve simulacije trajati 30 sekundi.

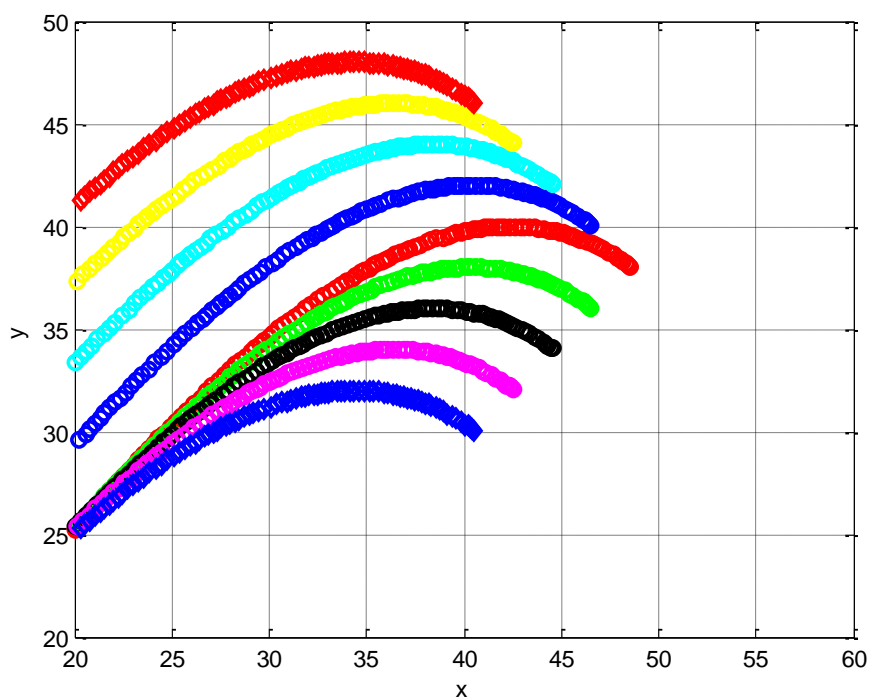
Prvo ćemo prikazati simulacije s najmanje komunikacije između robota.



**Slika 46. Kinematika gibanja formacije po krivulji, 5 robota i najmanje komunikacije**



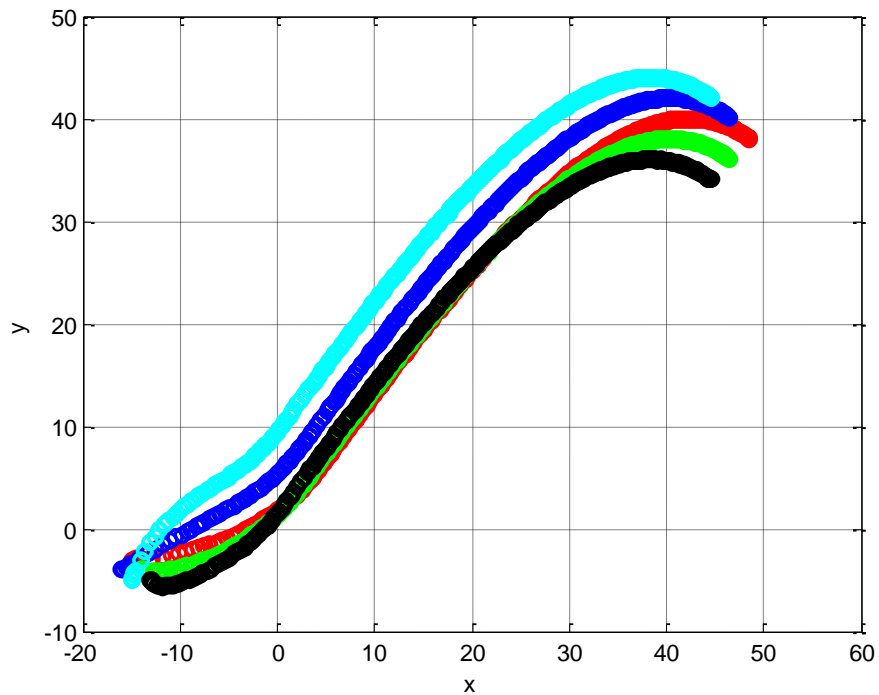
**Slika 47. Kinematika gibanja formacije po krivulji, 9 robota i najmanje komunikacije**



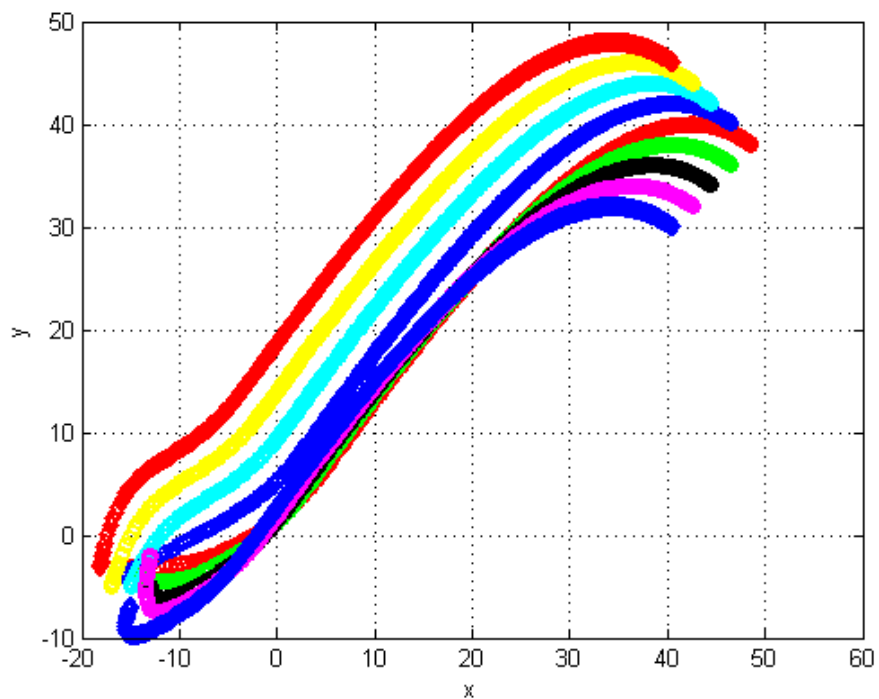
**Slika 48. Kinematika gibanja formacije po krivulji, 9 robota i najmanje komunikacije, detalj prikaza formacije**

Kako se i vidi na Slikama 46.,47. i 48., V formacija je uspostavljena i održana za slučaj s najmanje komunikacije između robota, te roboti s relativno malim odstupanjima prate željene trajektorije.

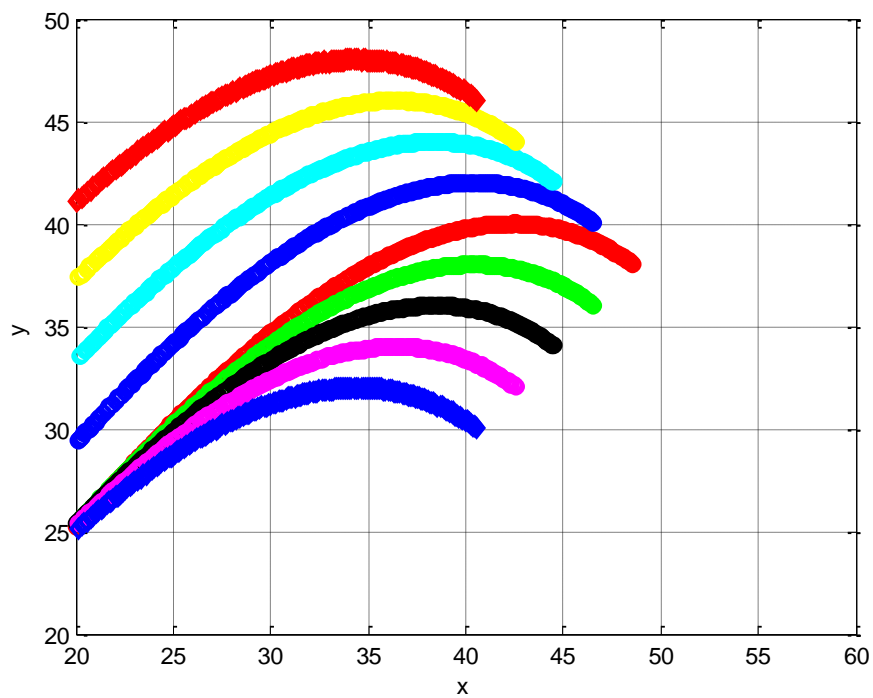
Slijedi prikaz simulacija kinematike s najviše komunikacije između robota.



**Slika 49. Kinematika gibanja formacije po krivulji, 5 robota i najviše komunikacije**

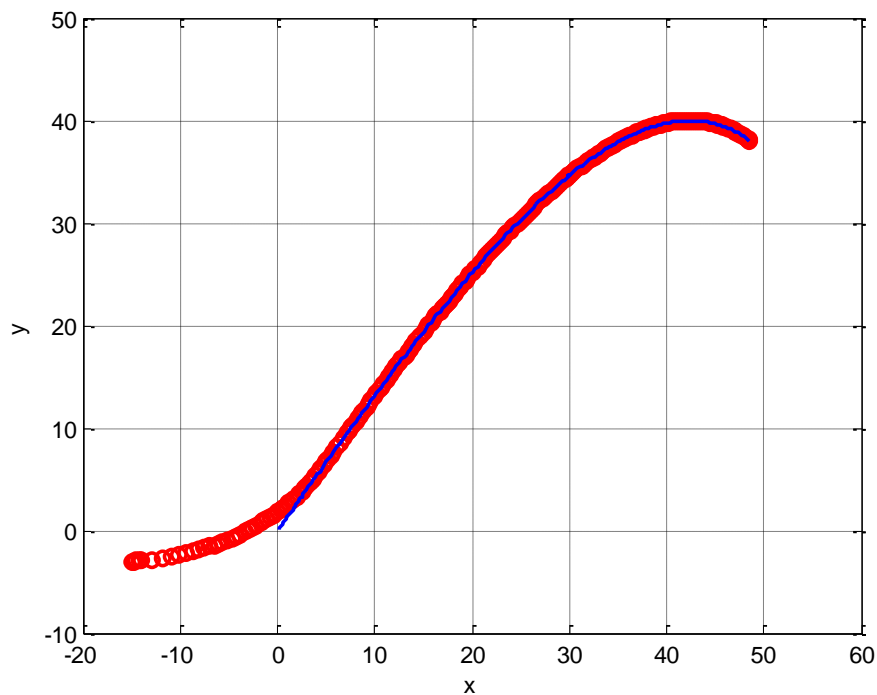


**Slika 50. Kinematika gibanja formacije po krivulji, 9 robota i najviše komunikacije**



**Slika 51. Kinematika gibanja formacije po krivulji, 9 robota i najviše komunikacije, detalj prikaza formacije**

Prikazane simulacije uspostave formaciju, te se gibaju u formaciji, a uz to s vrlo malim odstupanjima prate željene trajektorije, kao što pokazuje Slika 52., za vodećeg robota i njegovu trajektoriju.



**Slika 52. Odstupanje putanje robota od željene trajektorije upravljačkog zakona prikazanog jednačbom (108)**

Gornje simulacije ukazuju na dodanu vrijednost našeg predloženog upravljačkog zakona. Željene trajektorije se prate s vrlo malim odstupanjima, te je formacija uspostavljena i održana. Razlike između simulacija s najmanje i s najviše komunikacije između robota su vrlo male. Samo na početku se vidi značajnija razlika u brzini uspostavljanja formacije, što je bilo i očekivano zbog različite razine komunikacije. U kasnijim dijelovima simulacije, razlika gotovo i nema.

### 7.2.3 Dinamika gibanja formacije po krivulji

U ovom ćemo odjeljku predložiti upravljački zakon za dinamiku gibanja formacije po krivulji. Uz to ćemo prikazati prikladne simulacije kojima ćemo provjeriti učinkovitost predloženog upravljačkog zakona, a na kraju ćemo prikazane simulacije i komentirati.

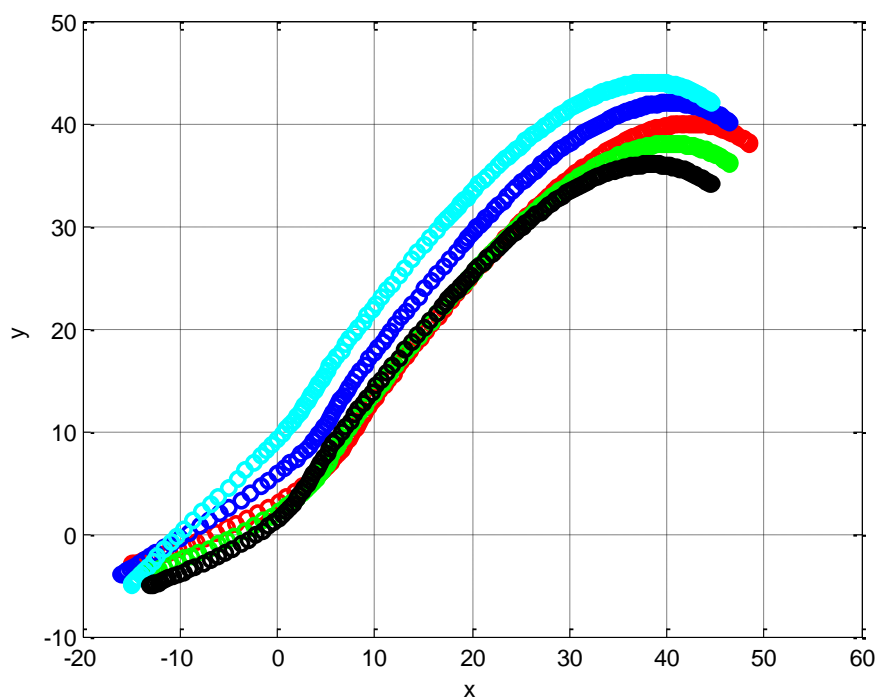
Predloženi upravljački zakon glasi

$$\begin{aligned}
 u_i = & \ddot{\delta}_i - \sigma \left[ (z_i - \delta_i) + \gamma (v_i - \dot{\delta}_i) \right] \\
 & - \sum_{j=1}^n g_{ij} k_{ij} \left[ (z_i - z_j) - (\delta_i - \delta_j) \right] \\
 & - \sum_{j=1}^n \varepsilon (v_i - v_j),
 \end{aligned} \tag{109}$$

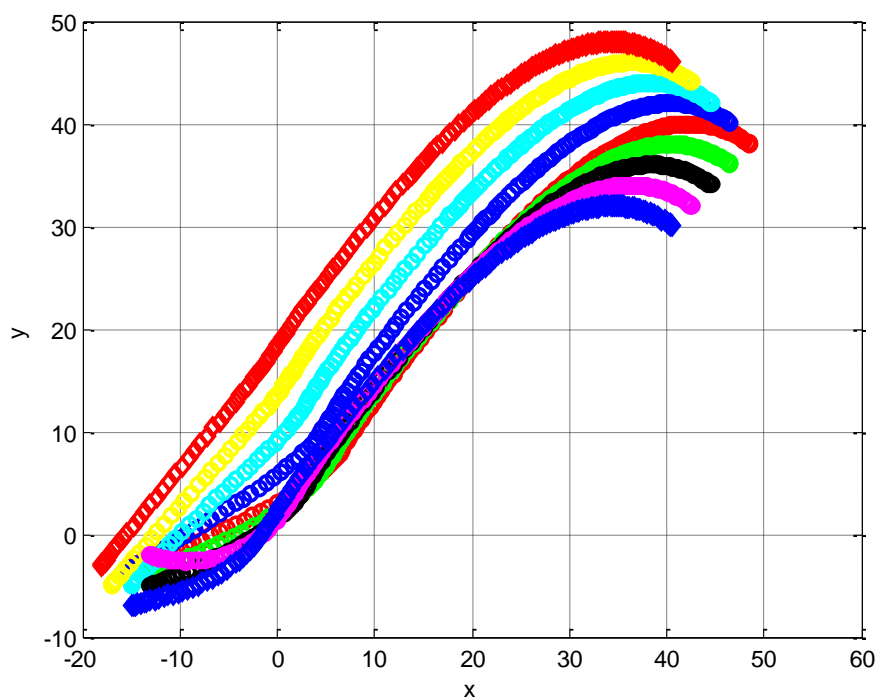
gdje drugi član povećava važnost praćenja željene trajektorije, treći član je zadužen za uspostavljanje formacije, dok četvrti pokušava izjednačiti brzine.  $\sigma$  je pojačanje i ono je postavljeno na vrijednost 2, dok su svi ostali parametri definirani u prethodna dva odjeljka uz pojačanje  $\varepsilon = 1$ .

Opet ćemo prikazati simulacije s 5 i s 9 robota, za grafove komunikacije koji su definirani na Slikama 28., 29., 32. i 33., što znači da će biti dvije skupine simulacija, jedne za najmanje komunikacije među robotima, a druge uz potpunu komunikaciju između robota. Opet ćemo koristiti V formaciju, radi jednostavnosti prikaza. Svim će simulacijama biti zadani isti početni uvjeti, te će sve simulacije trajati 30 sekundi.

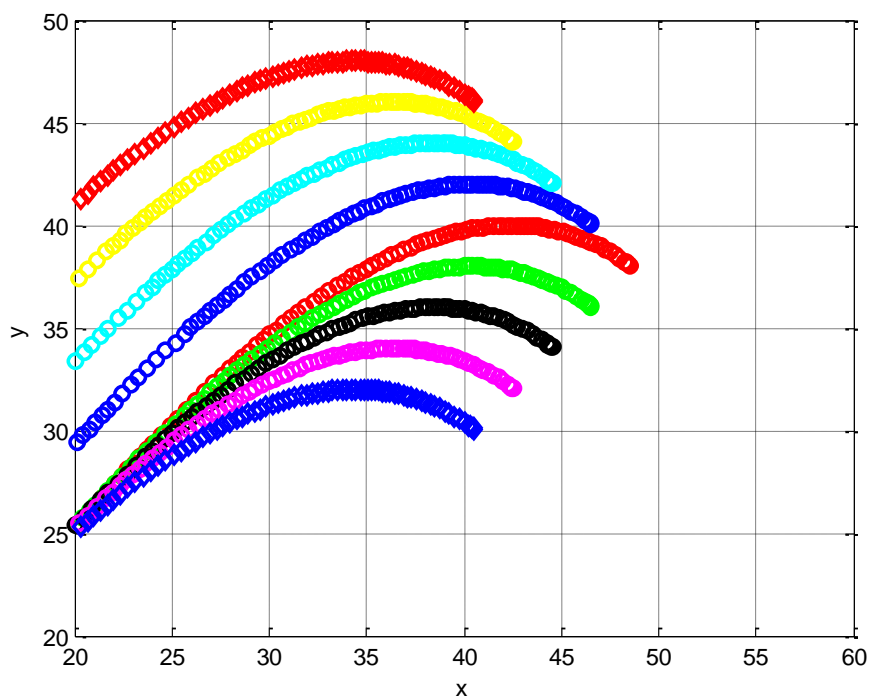
Prvo ćemo prikazati simulacije s najmanje komunikacije između robota.



Slika 53. Dinamika gibanja formacije po krivulji, 5 robota i najmanje komunikacije

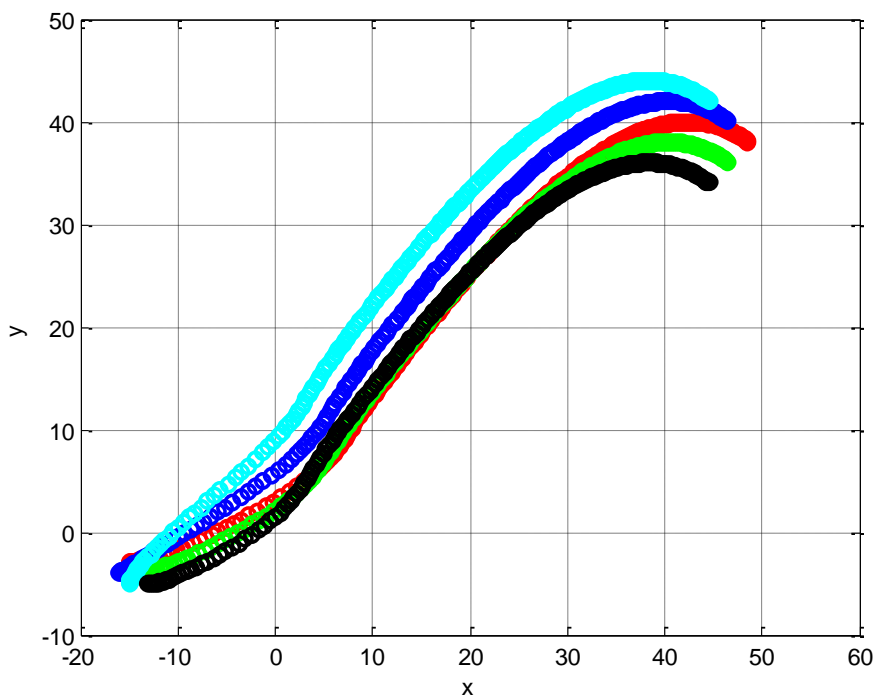


Slika 54. Dinamika gibanja formacije po krivulji, 9 robota i najmanje komunikacije

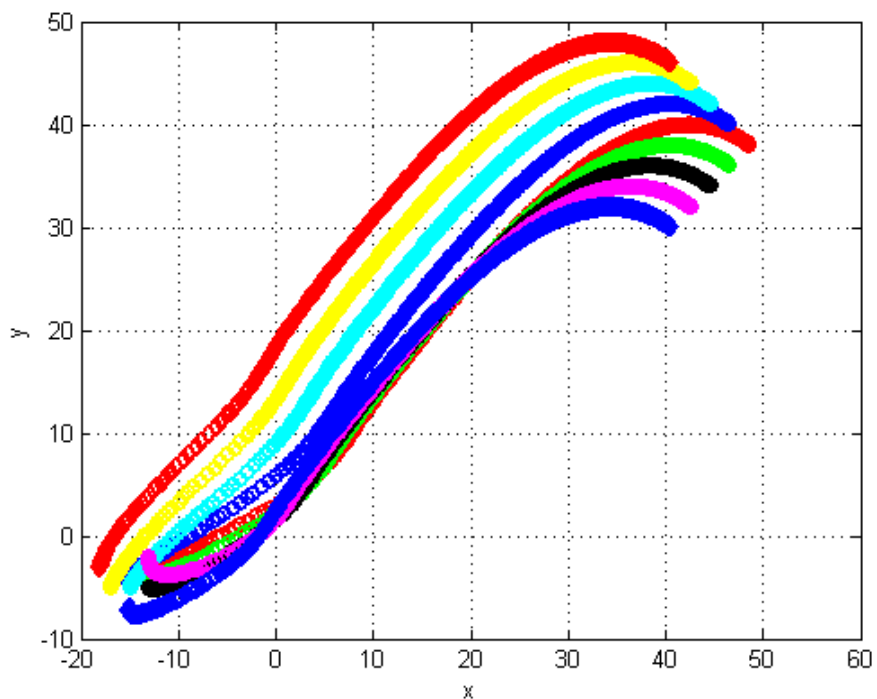


**Slika 55. Dinamika gibanja formacije po krivulji, 9 robota i najmanje komunikacije, detalj prikaza formacije**

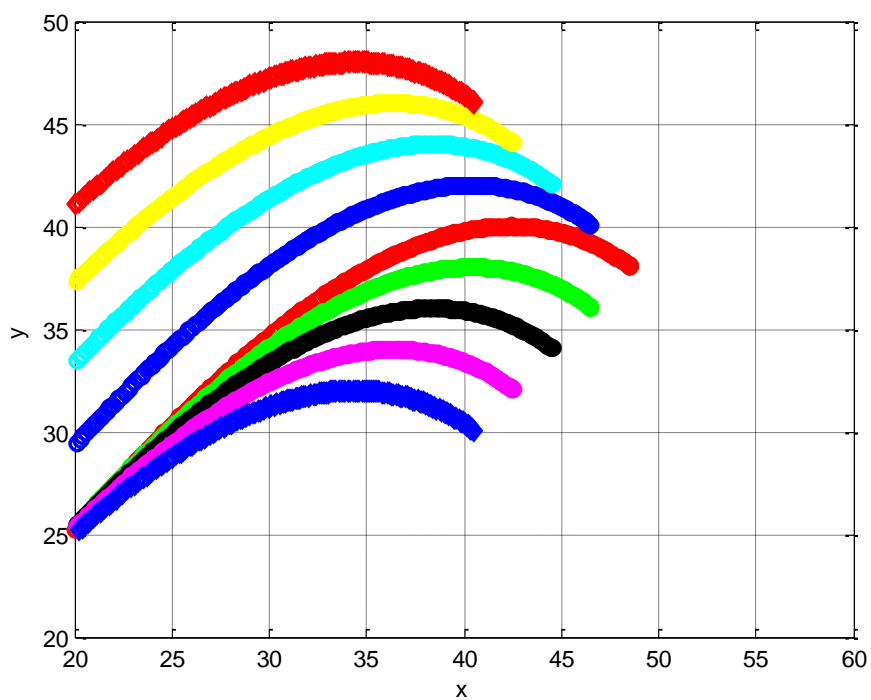
Sada ćemo prikazati simulacije s najvišom razinom komunikacije.



**Slika 56. Dinamika gibanja formacije po krivulji, 5 robota i najviše komunikacije**



Slika 57. Dinamika gibanja formacije po krivulji, 9 robota i najviše komunikacije



Slika 58. Dinamika gibanja formacije po krivulji, 9 robota i najviše komunikacije, detalj prikaza formacije



Sve prikazane simulacije izvršavaju zadatke koje smo im zadali: uspostave formaciju, gibaju se u formaciji te prate željene trajektorije. Razlika između simulacija s najmanjom razinom komunikacije i s najvišom razinom komunikacije je veoma malena i očekivano se javlja na početku simulacije, dok se formacija ne uspostavi, a kasnije razlike gotovo da i nema. Ove simulacije su prikazale efikasnost predloženog upravljačkog sustava te jednostavnu mogućnost njegove primjene.

## 6.2.4 Komentar

Prikazane simulacije kinematike i dinamike gibanja u formaciji ilustriraju učinkovitost predloženih upravljačkih zakona. Roboti i u slučajevima s najmanje komunikacije precizno ispunjavaju svoje zadatke. Za razliku od poglavlja 6.1.3 gdje roboti u dinamici, za najmanju razinu komunikacije, značajno odstupaju te otežano uspostavljaju formaciju, ovdje to nije slučaj. Razlog za to je što roboti nemaju samo informaciju o robotu kojeg prate, nego imaju i trajektoriju koju moraju pratiti. Upravo je to postojanje referentne trajektorije razlog efikasnosti predstavljenih upravljačkih zakona. Uz to što nema rasipanja robota tijekom gibanja, postoji i veća razina kontrole gibanja robota, zato što u svakom trenutku znamo gdje bi se oni trebali nalaziti.

U našem slučaju korištena je samo jedna krivulja, ali predstavljeni upravljački zakoni primjenjivi su za sve vrste krivulja. Treba paziti da se krivulje precizno definiraju, uz to treba izabrati prikladne krivulje, one koje nisu previše brze, tako da ih roboti mogu pratiti.

Sada ćemo prikazati rezultat simulacije kinematike i dinamike s 5 robota i najvišom razinom komunikacije za slučaj kada je referentna trajektorija definirana kao

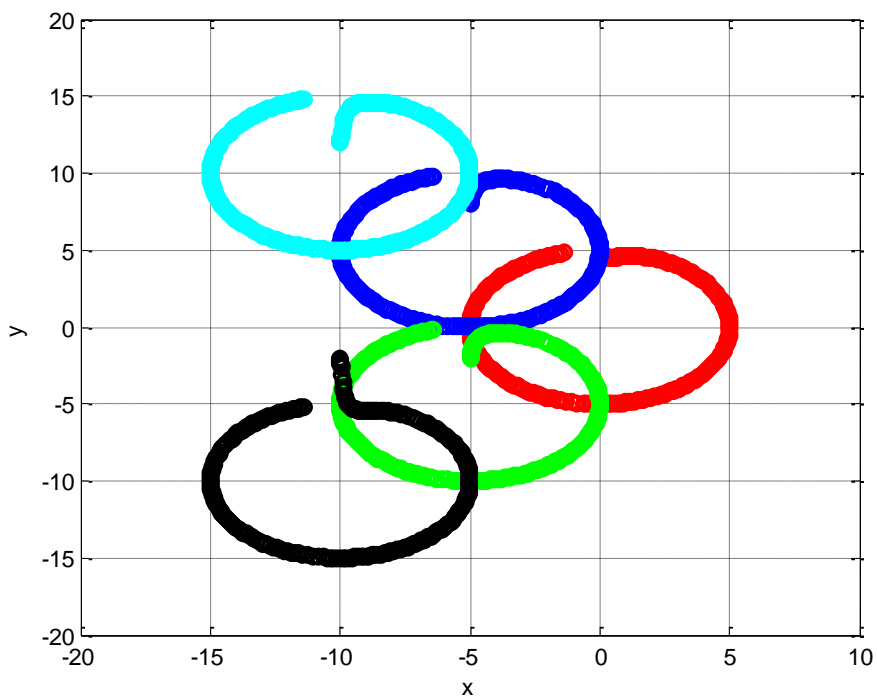
$$\delta(t) = [5 \sin(t/5), 5 \cos(t/5)]^T, \quad (110)$$

i kada je definirana kao

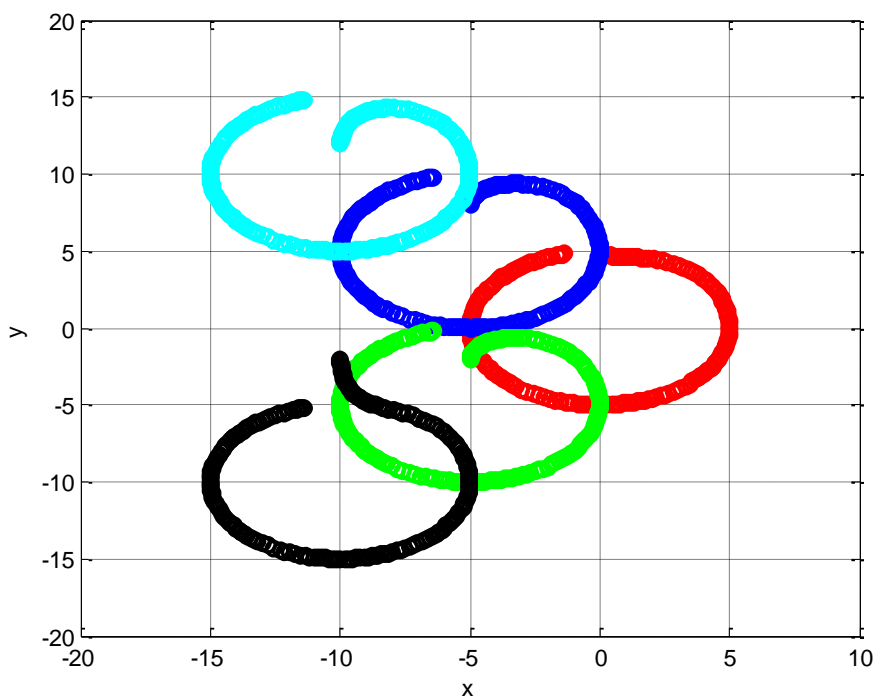
$$\delta(t) = [5(t/2), 5 \sin(t)]^T. \quad (111)$$

Krivulja definirana jednadžbom (110) odgovara kružnici radijusa 5, a Slike 59. i 60. prikazuju rezultate simulacija koristeći navedenu krivulju. Krivulja definirana jednadžbom (111) odgovara sinusoidi, a Slike 61. i 62. prikazuju rezultat simulacija.

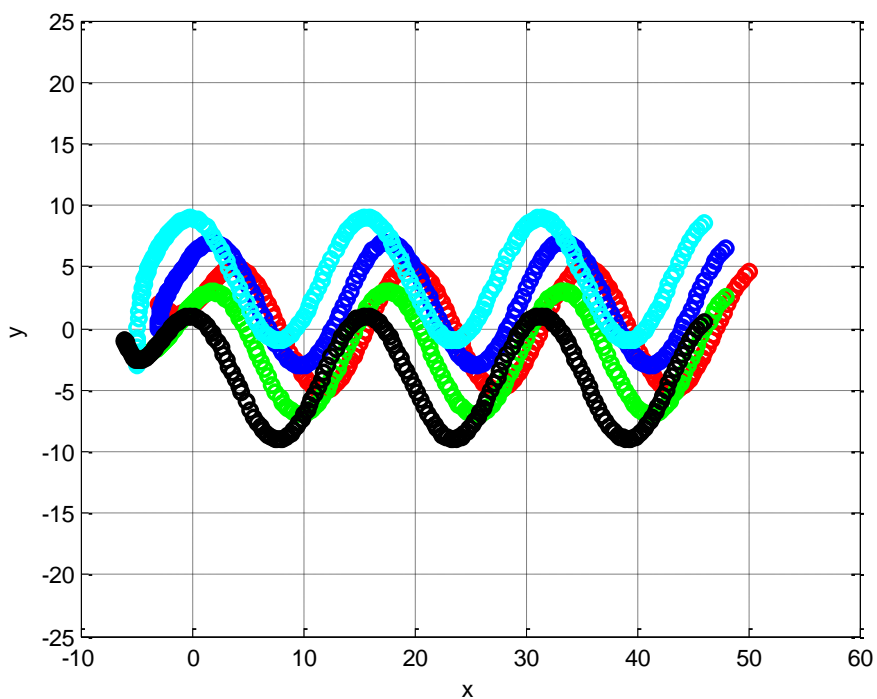
Pomoću ovih simulacija smo pokazali da predstavljeni upravljački zakon pokazuje zadovoljavajuće rezultate za niz različitih tipova krivulja.



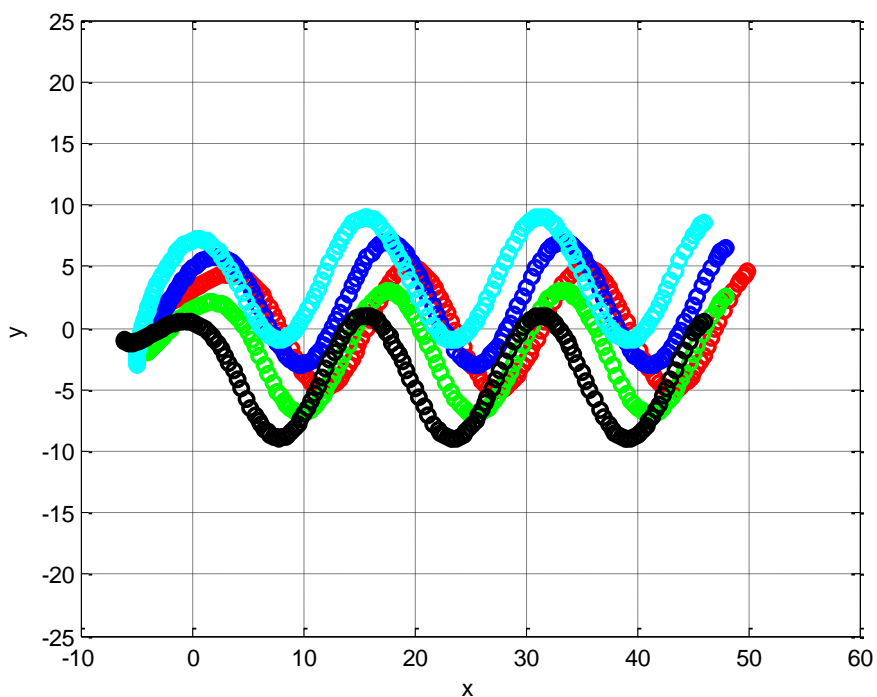
Slika 59. Kinematika gibanja formacije po krivulji, gibanje po kružnici



Slika 60. Dinamika gibanja formacije po krivulji, gibanje po kružnici



**Slika 61. Kinematika gibanja po krivulji, gibanje po sinusoidi**



**Slika 62. Dinamika gibanja po krivulji, gibanje po sinusoidi**

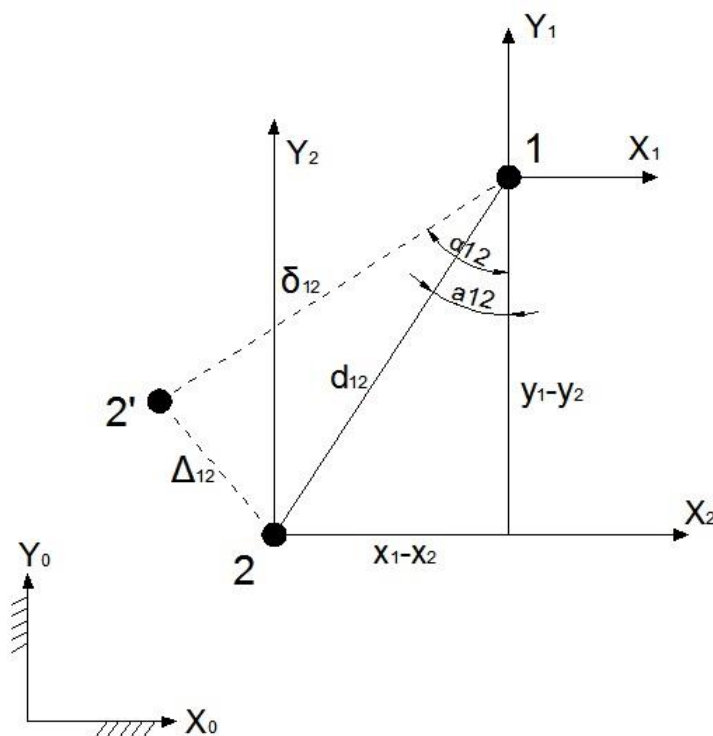
## 6.3 Kombinacija gibanja u formaciji

U ovom ćemo poglavlju predložiti upravljački zakon koji omogućava gibanje formacije i linijski i po nekoj zadanoj krivulji. Prvo ćemo opisati i izvesti upravljački zakon, a kasnije ćemo ga ispitati u simulacijama. Na kraju poglavlja ćemo dati kratak komentar prikazanih simulacija.

### 6.3.1. Upravljački zakon

U ovom ćemo odjeljku izvesti upravljački zakon koji će nam omogućiti gibanje formacije uz novitet da možemo upravljati orijentacijom cijele formacije. Cilj je ostvariti gibanje formacije kao da se radi o krutom tijelu u ravnini, a zadane su trajektorija, pozicije i orijentacija tog tijela (formacije).

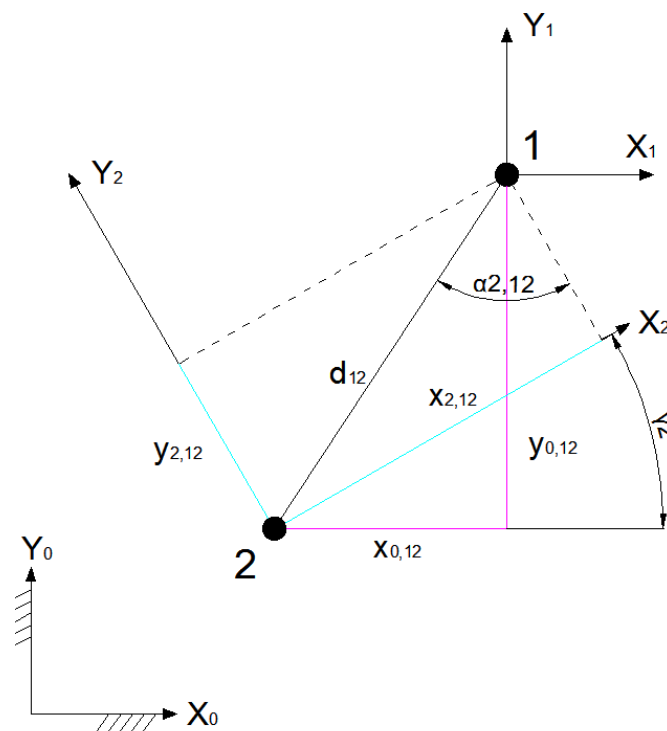
Slika 63. prikazuje osnovu ideje za upravljanje formacijom. Ovdje želimo upravljati s udaljenosti i s kutom između robota 1 i robota 2, a ne direktno s njihovim pozicijama kako smo to radili u prethodnim poglavljima.



Slika 63. Definiranje koordinatnih sustava i parametara

Na Slici 63.  $\{X_0, Y_0\}$  predstavlja globalni nepomični koordinatni sustav,  $\{X_i, Y_i\}$  je lokalni koordinatni sustav robota  $i$ ,  $d_{12}$  je trenutna udaljenost između robota 1 i robota 2,  $a_{12}$  je trenutni kut između robota 1 i robota 2,  $\delta_{12}$  je željena udaljenost između robota 1 i 2,  $\alpha_{12}$  je željeni kut između robota 1 i 2 (definiran kao kut koji dužina koja spaja dva robota zatvara s osi  $Y_0$  nepomičnog koordinatnog sustava),  $\Delta_2$  je razlika između trenutne i željene pozicije robota 2, dok su  $x_1 - x_2$  i  $y_1 - y_2$  razlike između pozicija robota 1 i robota 2 u globalnom koordinatnom sustavu.

Cilj upravljanja je da  $d_{12} \rightarrow \delta_{12}$  i  $a_{12} \rightarrow \alpha_{12}$ . Uz to, želimo i mogućnost upravljanja orijentacijom cijele formacije na jednostavan način. To nam omogućuje kut  $\gamma$ .



Slika 64. Definiranje kuta  $\gamma_2$

Na Slici 64. vidimo da je orijentacija lokalnog koordinatnog sustava robota 2 definirana kutom  $\gamma_2$ .  $x_{0,12}$  predstavlja razliku između pozicija robota 1 i robota 2 u smjeru osi  $x$  u globalnom koordinatnom sustavu, to jest  $x_{0,12} = x_1 - x_2$ , dok je  $y_{0,12}$  razlika između pozicija robota 1 i robota 2 u smjeru osi  $y$  u globalnom koordinatnom sustavu, to jest  $y_{0,12} = y_1 - y_2$ .  $x_{2,12}$  i  $y_{2,12}$  su vrijednosti  $x_{0,12}$  i  $y_{0,12}$  u lokalnom koordinatnom sustavu robota 2, a one se dobiju pomoću matrice rotacije na sljedeći način

$$\begin{bmatrix} x_{2,12} \\ y_{2,12} \end{bmatrix} = \begin{bmatrix} \cos(\gamma_2) & \sin(\gamma_2) \\ -\sin(\gamma_2) & \cos(\gamma_2) \end{bmatrix} \begin{bmatrix} x_{0,12} \\ y_{0,12} \end{bmatrix}, \quad (112)$$

odnosno zapisano jednostavnije

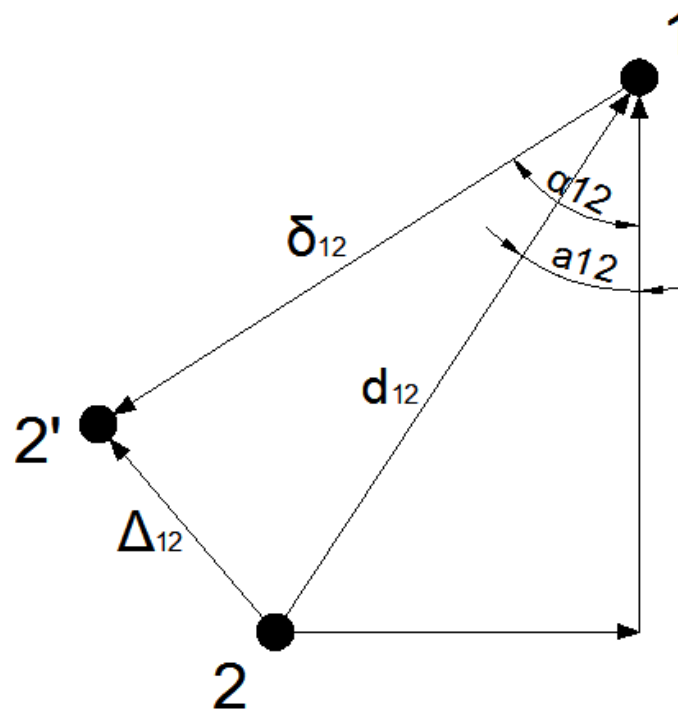
$$\begin{bmatrix} x_{2,12} \\ y_{2,12} \end{bmatrix} = R(\gamma_2) \begin{bmatrix} x_{0,12} \\ y_{0,12} \end{bmatrix}, \quad (113)$$

gdje je  $R(\gamma_2)$  matrica rotacije. Sljedeće definiramo vrijednost trenutnog kuta između robota 1 i robota 2 u lokalnom koordinatnom sustavu robota 2 pomoću Slike 64. kao

$$a_{2,12} = \arctg\left(\frac{x_{2,12}}{y_{2,12}}\right). \quad (114)$$

Udaljenost između robota 1 i robota 2 je jednaka i u globalnom i u lokalnom sustavu, a definirana je kao

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (115)$$



Slika 65. Definiranje udaljenosti između trenutne i željene pozicije robota

Iz Slike 65. je jasno vidljivo da je

$$\vec{\Delta}_{12} = \vec{d}_{12} + \vec{\delta}_{12}. \quad (116)$$

$\vec{d}_{12}$  se definira kao

$$\vec{d}_{12} = \begin{bmatrix} d_{12} \sin(a_{12}) \\ d_{12} \cos(a_{12}) \end{bmatrix}, \quad (117)$$

dok se  $\vec{\delta}_{12}$  definira kao

$$\vec{\delta}_{12} = \begin{bmatrix} -\delta_{12} \sin(\alpha_{12}) \\ -\delta_{12} \cos(\alpha_{12}) \end{bmatrix}. \quad (118)$$

Vrijednosti  $\delta_{12}$  i  $\alpha_{12}$  su jednake u globalnom i u lokalnom koordinatnom sustavu.

Naš cilj upravljanja je da  $d_{12} \rightarrow \delta_{12}$  i  $a_{2,12} \rightarrow \alpha_{12}$ , odnosno  $\Delta_{12} \rightarrow 0$ . Upravljački zakon koji to omogućuje glasi u lokalnom koordinatnom sustavu

$$u_{\text{lok}} = k \Delta_{\text{lok},12}, \quad (119)$$

odnosno

$$u_{\text{lok}} = k \begin{bmatrix} d_{12} \sin(a_{2,12}) - \delta_{12} \sin(\alpha_{12}) \\ d_{12} \cos(a_{2,12}) - \delta_{12} \cos(\alpha_{12}) \end{bmatrix}, \quad (120)$$

gdje je  $k$  pojačanje sustava. Upravljački zakon prikazan jednadžbom (120) u globalnom koordinatnom sustavu glasi

$$u = \begin{bmatrix} \cos(\gamma_2) & -\sin(\gamma_2) \\ \sin(\gamma_2) & \cos(\gamma_2) \end{bmatrix} u_{\text{lok}}, \quad (121)$$

odnosno

$$u = R^{-1}(\gamma_2) u_{\text{lok}}. \quad (122)$$

Ovaj upravljački zakon koristi model s jednim integratorom, odnosno, odnosi se na kinematiku gibanja u formaciji.

U općenitom slučaju upravljački zakon definiran jednadžbom (120) glasi

$$u_{\text{lok}} = k \begin{bmatrix} d_{ij} \sin(a_{j,ij}) - \delta_{ij} \sin(\alpha_{ij}) \\ d_{ij} \cos(a_{j,ij}) - \delta_{ij} \cos(\alpha_{ij}) \end{bmatrix}, \quad (123)$$

Za sve  $i = 1, \dots, n$  i  $j = 2, \dots, n$ , dok upravljački zakon definiran jednadžbom (122) u općenitom slučaju glasi

$$u = R^{-1}(\gamma_j) u_{\text{lok}}, \quad (124)$$

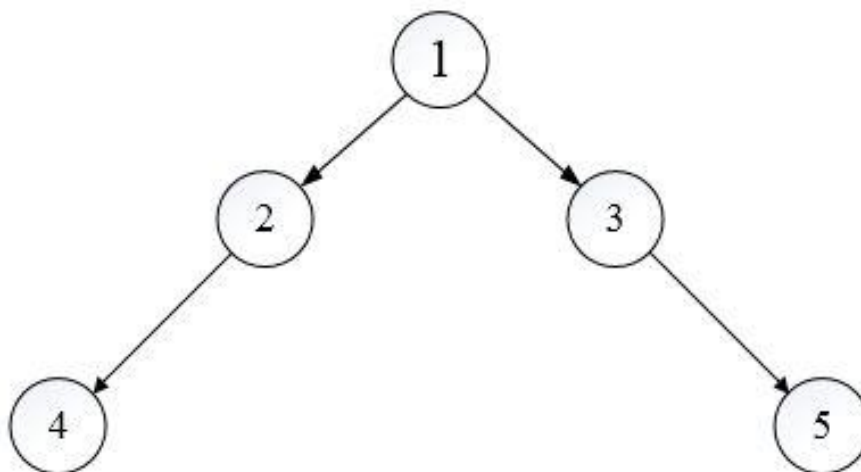
za sve  $j = 2, \dots, n$ .

## 6.3.2 Simulacije

U ovom ćemo odjeljku ispitati predloženi upravljački sustav prikazan jednadžbom (123). Prikazat ćemo 2 skupine simulacija, kada se roboti gibaju po ravnoj liniji i kada se gibaju po nekoj referentnoj krivulji. Za razliku od prijašnjih poglavlja, robot 1 je vodeći robot, a svi ostali roboti njega prate na određenoj udaljenosti i pod određenim kutom. Za svaku skupinu simulacija prikazat ćemo slučaj s 5 i s 9 robota.

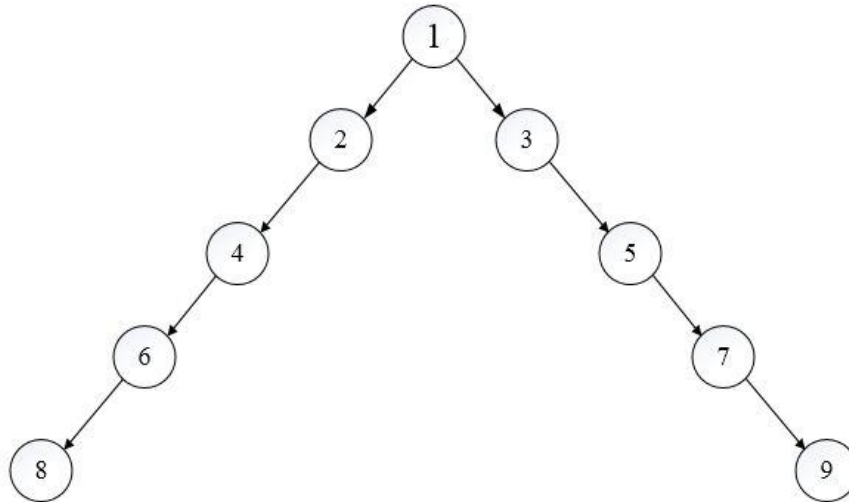
### 6.3.2.1 Gibanje po liniji

Prva skupina simulacija će biti prikazana za gibanje po liniji. Prikazat ćemo slučaj sa 5 i s 9 robota te slučaj s 5 robota kod kojeg mijenjamo orijentaciju cijele formacije promjenom kuta  $\gamma$ . Vodeći robot je robot 1 (u simulacijama označen crvenim kružićima) koji dobiva uputu za gibanje (u našem slučaju dobije naredbu za gibanje translacijskom brzinom konstantne vrijednosti). Graf komunikacije za slučaj s 5 i s 9 robota je sličan prijašnjima uz razliku da robot 1 više ne vidi robote 2 i 3 kako prikazuju Slika 66. i Slika 67.



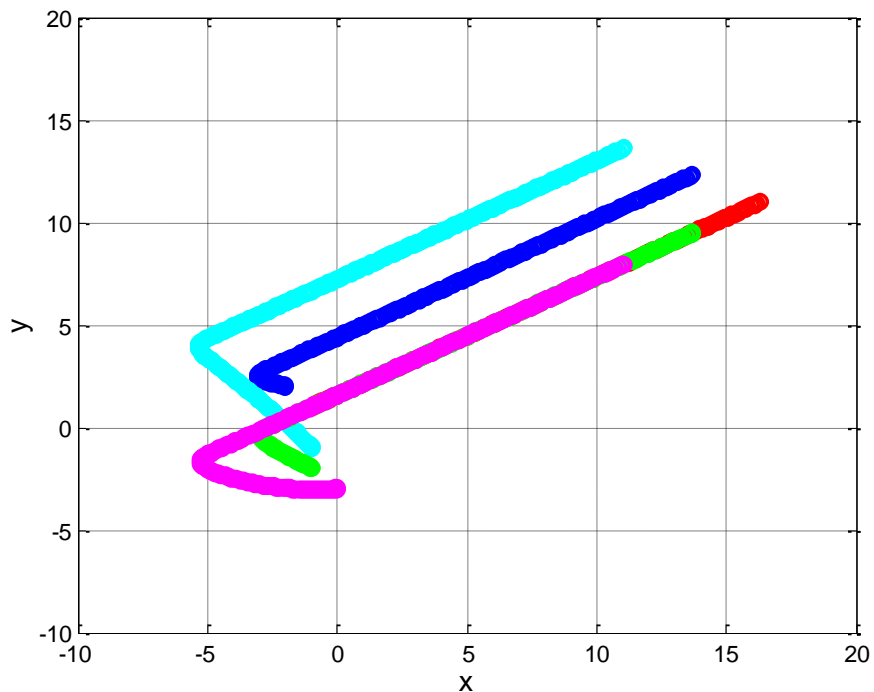
Slika 66. Graf komunikacije s 5 robota gdje je robot 1 vođa



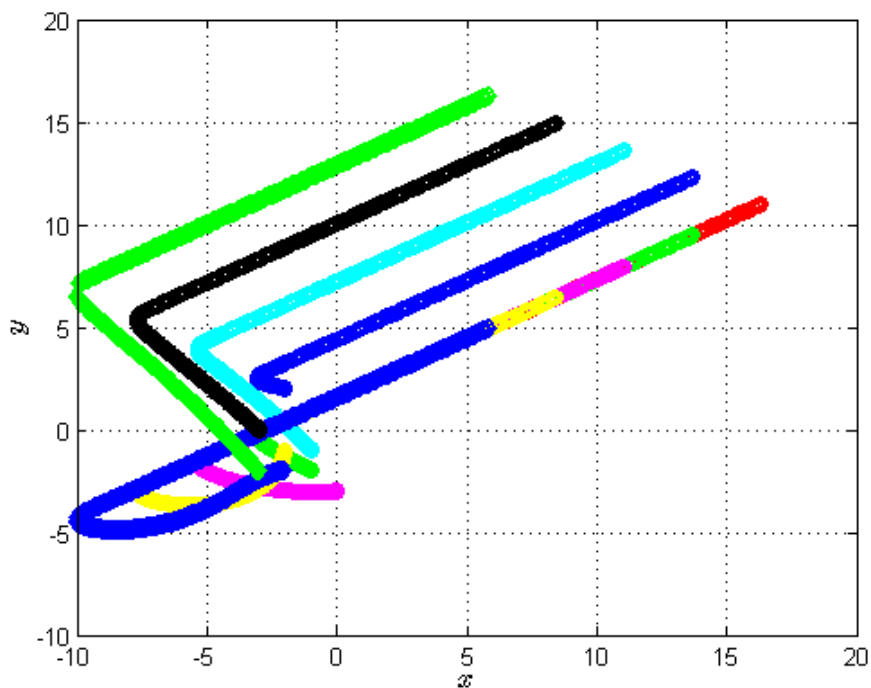


Slika 67. Graf komunikacije s 9 robota gdje je robot 1 vođa

Robot 1 (crveni) koristi isti upravljački zakon kao i vodeći robot iz poglavlja 5.1. Svi roboti osim robota 1 koriste upravljački zakon definiran jednađbom (124). Između svih robota je definirana jednaka željena udaljenost, dok je u svakoj grani V formacije jednak željeni kut između susjednih robota. Vrijednost faktora  $k$  je postavljena na 5, dok je  $\gamma = -\pi/2$ .



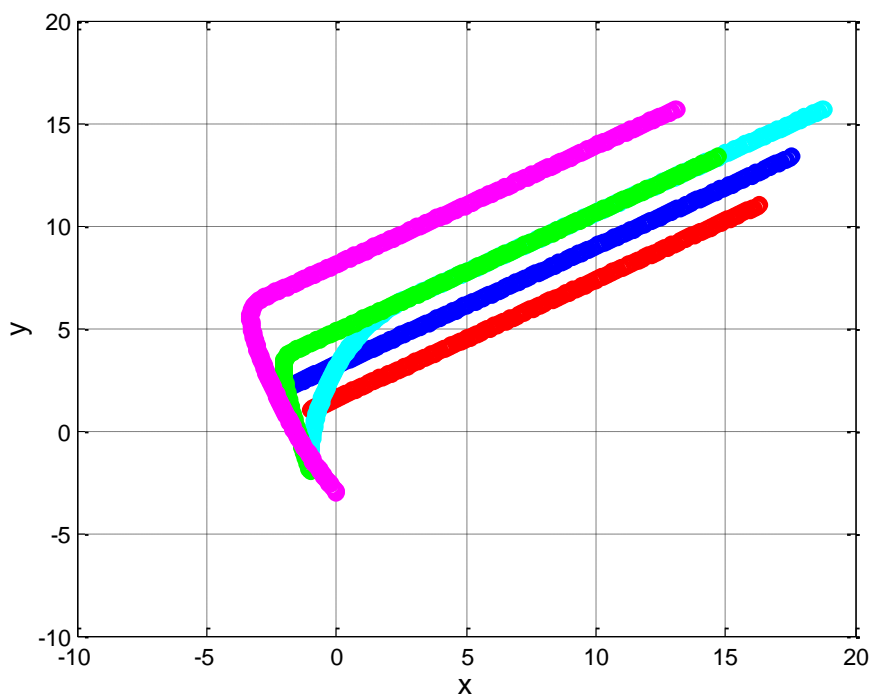
Slika 68. Gibanje formacije s 5 robota po liniji, crveni robot je vodeći robot



**Slika 69.** Gibanje formacije s 9 robota po liniji, crveni robot je vodeći robot

Iz Slika 68. i 69. se jasno vidi da je formacija uspostavljena vrlo brzo te da je ona očuvana za vrijeme gibanja. Ovakvi rezultati simulacije pokazuju da predloženi upravljački zakon na zadovoljavajući način ostvaruje željeni cilj.

Sada ćemo za prikazani slučaj simulacije s 5 robota promijeniti kut  $\gamma$  pratećih robota, što će uzrokovati promjenu orijentacije cijele formacije. Kut će biti postavljen na vrijednost  $\gamma = \pi$ .



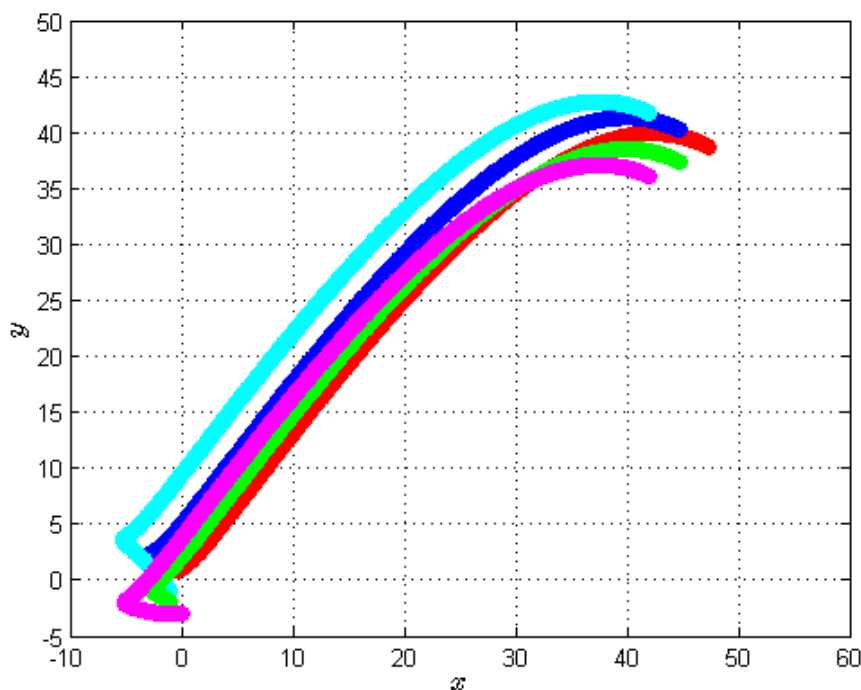
**Slika 70.** Gibanje formacije s 5 robota po liniji, crveni robot je vodeći robot, kut  $\gamma = \pi$

Slika 70. prikazuje da se promjenom kuta  $\gamma$  promijeni orijentacija cijele formacije, ali da to ne utječe na postizanje i održavanje formacije.

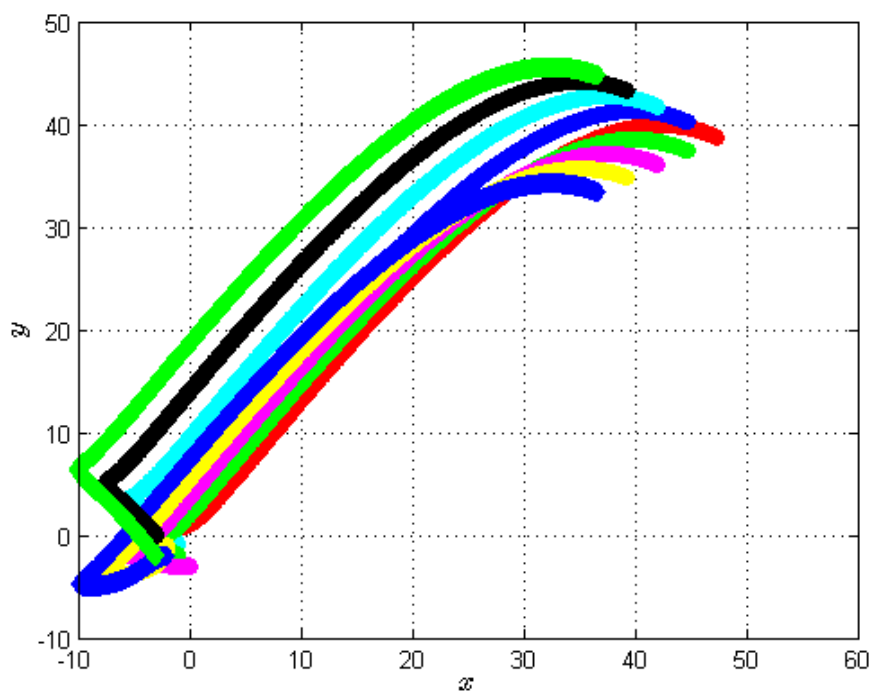
### 6.3.2.2 Gibanje po krivulji

U ovom ćemo dijelu prikazati drugu skupinu simulacija, koje omogućuju gibanje po krivulji. Prikazat ćemo slučaj s 5 i s 9 robota te slučaj s 5 robota kod kojeg mijenjamo orijentaciju promjenom kuta  $\gamma$ . Vodeći robot je robot 1 (u simulacijama označen crvenim kružićima) koji prati referentnu krivulju. Referentna krivulja koju prati robot 1 definirana je jednačbom (107). Za razliku od poglavlja 6.2 gdje su svi roboti pratili svoje referentne krivulje, ovdje samo robot 1 prati referentnu krivulju, dok svi ostali roboti prate samo robota ispred sebe. Graf komunikacije za slučaj s 5 i s 9 robota prikazan je na Slici 66. i Slici 67.

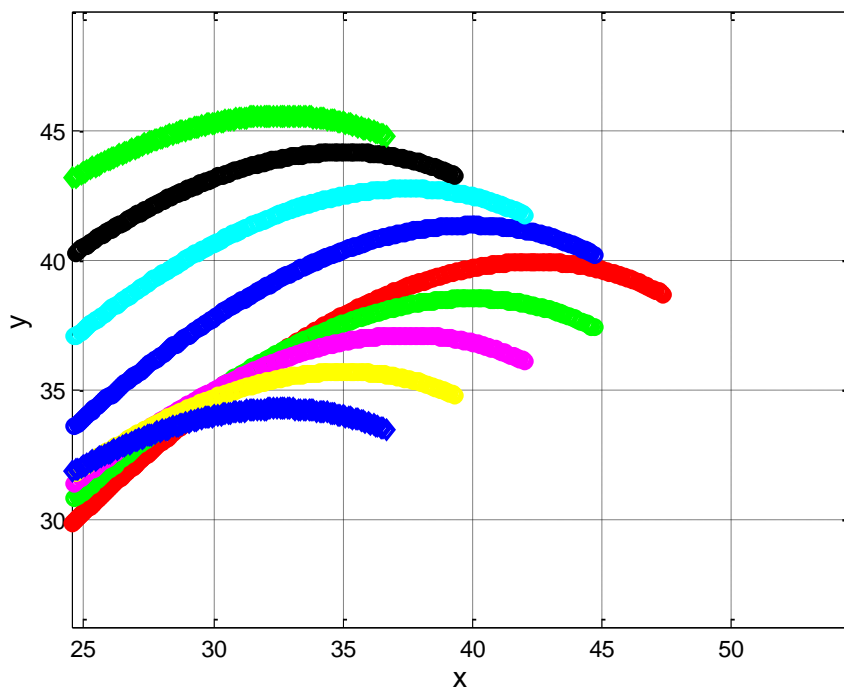
Robot 1 koristi isti upravljački zakon kao i prateći roboti iz poglavlja 5.1, samo što on ovdje prati krivulju. Svi roboti osim robota 1 koriste upravljački zakon definiran jednačbom (124). Između svih robota je definirana jednaka željena udaljenost, dok je u svakoj grani  $V$  formacije jednak željeni kut između susjednih robota. Vrijednost faktora  $k$  je postavljena na 5, dok je  $\gamma = -\pi/2$ .



Slika 71. Gibanje formacije s 5 robota po krivulji, crveni robot je vodeći



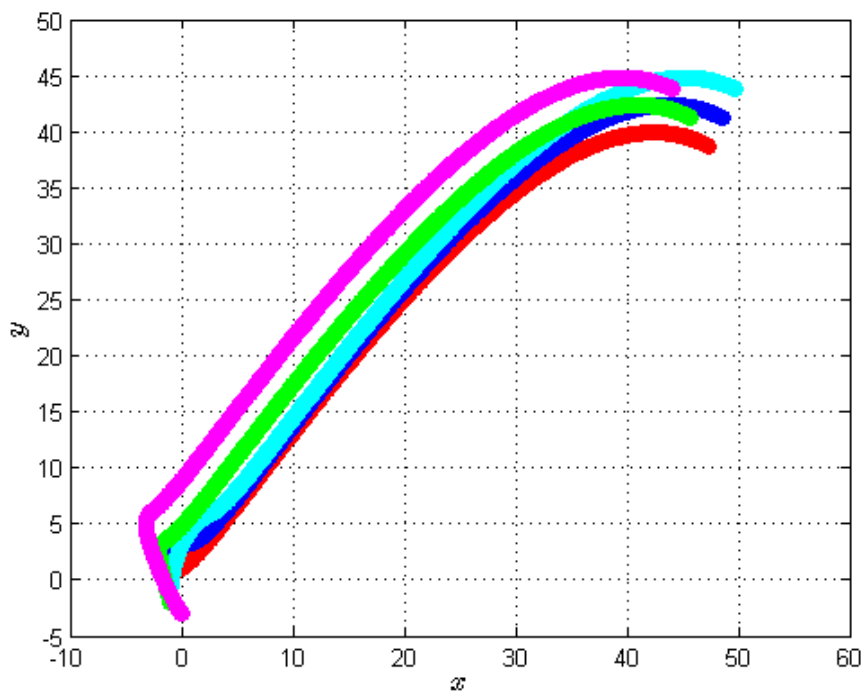
Slika 72. Gibanje formacije s 9 robota po krivulji, crveni robot je vodeći



Slika 73. Gibanje formacije s 9 robota po krivulji, crveni robot je vodeći, detaljni prikaz uspostavljene formacije

Iz Slike 71., 72. i 73. se jasno vidi da je formacija uspostavljena vrlo brzo te da je ona očuvana za vrijeme gibanja. Ovime smo potvrdili da naš predloženi upravljački zakon omogućuje primjenu i za gibanje formacije robota po krivulji.

Sada ćemo za prikazani slučaj simulacije s 5 robota promijeniti kut  $\gamma$  svih robota koji prate robota 1, što će uzrokovati promjenu orijentacije cijele formacije. Kut će biti postavljen na vrijednost  $\gamma = \pi$ .



**Slika 74. Gibanje formacije s 5 robota po krivulji, crveni robot je vodeći robot, kut  $\gamma = \pi$**

Slika 74. prikazuje da se promjenom kuta  $\gamma$  promijeni orijentacija cijele formacije, ali da to ne utječe na postizanje i održavanje formacije. Iz toga zaključujemo da prikazani upravljački zakon zadovoljava i za gibanje formacije po krivulji.

### 6.3.3. Komentar

U ovom poglavlju smo predstavili upravljački zakon koji se može koristiti i za gibanje po liniji i za gibanje po krivulji. Njegova implementacija ima nekoliko prednosti koje su sažete u sljedećoj listi.

1. Upravljački zakon je znatno jednostavniji od upravljačkih zakona prikazanih u poglavljima 6.1 i 6.2, a daje zadovoljavajuće rezultate.
2. Postoji mogućnost direktnog upravljanja operatera s vodećim robotom, što ponekad može biti vrlo korisno.
3. Manja je potrebna razina komunikacije.
4. Postoji mogućnost jednostavne promjene orijentacije cijele formacije.
5. Postoji mogućnost primjene za gibanje po liniji ili po krivulji.

Nažalost, postoje i neki nedostaci predstavljenog upravljačkog zakona. Naime, svi roboti osim robota 1 gledaju svog nadređenog robota, međusobne komunikacije nema, a to može biti jako opasno zato što na primjer gubitkom robota 2 sa Slike 67. odmah gubimo i robote 4, 6 i 8. Tako nešto naravno nije dopustivo pa je potrebno dodati veću razinu komunikacije među robotima da bi se dobilo na robusnosti formacije. Uz to, prikazani upravljački zakon je prikladan samo za kinematiku gibanja u formaciji, za dinamiku gibanja u formaciji on nije primjenjiv.

## 7. ZAKLJUČAK

Područje mobilne robotike je jedno od područja koja se jako brzo razvijaju. Velike mogućnosti primjene mobilnih robota pridonijele su snažnom interesu znanstvenika, inženjera i poduzetnika za ovo područje. U ovom radu smo se bavili temom gibanja mobilnih robota u formaciji, koje ima široko područje primjene, poput pretraživanja terena, nošenja tereta, vožnje na autocestama, kontrole usjeva i brojne druge primjene. Prednosti gibanja mobilnih robota u formaciji su: veća robusnost takvih formacija, fleksibilnost, preciznost, efikasnost, energetska učinkovitost i veća vjerojatnost uspjeha.

U ovom radu predstavili smo tri upravljačka zakona koja omogućuju gibanje robota u formaciji. Svi su upravljački zakoni ispitani nizom simulacija koje su potvrdile da upravljački zakoni postižu zadovoljavajuće rezultate, te da su primjenjivi za izvršenje zadataka za koje su predviđeni. Za razliku od prva dva upravljačka zakona, treći predstavljeni smo u potpunosti samostalno razvili. Ovaj upravljački zakon omogućuje veću slobodu pri projektiranju gibanja, jednostavniji je od prethodna dva, omogućuje direktno upravljanje vodećim robotom, te kao najveću prednost nudi mogućnost jednostavnog mijenjanja orijentacije cijele formacije.

Na kraju smo programirali aplikaciju u programu Matlab. Ona omogućuje jednostavno ispitivanje i mijenjanje simulacija prikazanih u ovom radu. Zamišljena je kao uvod ili kao pomoć svima koji se žele upoznati s područjem mobilne robotike.

Ovaj rad bavi se trenutačnim dostignućima gibanja mobilnih robota u formaciji. Daljnjim razvojem područja mobilne robotike očekuje se sve veća primjena formacija mobilnih robota u različitim djelatnostima. Već sada mobilni roboti preuzimaju poslove koje su nekada radili isključivo ljudi, primjerice poslovi vezani uz skladištenja i transport dobara, a očekuje se da će broj poslova koje će mobilni roboti preuzimati biti sve veći te će u mnogima potpuno zamijeniti ljudsku radnu snagu.

## LITERATURA

- [1] <http://phenomena.nationalgeographic.com/2014/01/15/birds-that-fly-in-a-v-formation-use-an-amazing-trick/>
- [2] <https://annececiledragon.wordpress.com/research/blue-jack-mackerel-school/>
- [3] [https://en.wikipedia.org/wiki/V\\_formation](https://en.wikipedia.org/wiki/V_formation)
- [4] [https://en.wikipedia.org/wiki/Shoaling\\_and\\_schooling](https://en.wikipedia.org/wiki/Shoaling_and_schooling)
- [5] <http://www.princeton.edu/~icouzin/Couzin&Franks.pdf>
- [6] <http://jeb.biologists.org/content/213/14/2357>
- [7] Ren, W., Beard, W.R.: "Distributed consensus in Multi-vehicle cooperative control", Springer-Verlag, London, 2008.
- [8] <https://www.nasa.gov/feature/goddard/nasas-mms-spacecraft-achieve-tightest-flying-formation-ever>
- [9] Casbeer, D.W., Kingston, B.D., Beard, W.R., McLain, W.T., Li, S., Mehra, R.: "Cooperative forest fire surveillance using a team of small unmanned air vehicles", International Journal of Systems Sciences, vol.37, no. 6, pp. 351–360, 2006.
- [10] <https://www.eutruckplatooning.com/home/default.aspx>
- [11] <http://www.navy.mil/index.asp>
- [12] <http://www.kinze.com/article.aspx?id=341>
- [13] Siegwart, R., Nourbakhsh, I., Scaramuzza, D.: "Introduction to Autonomous Mobile Robots" 2nd Edition, Intelligent Robotics and Autonomous Agent series, The MIT Press, Massachusetts Institute of Technology, Massachusetts, 2011.
- [14] Lobrović, M.: eMIR, interni dokument, FSB, Zagreb
- [15] Ivanjko, E., Petrinić, T., Petrović, I.: "Modelling of Mobile Robot Dynamics", Proceedings of the 7th EUROSIM Congress on Modelling and Simulation Vol. 2: Full Papers / Šnorek, Miroslav ; Buk, Zdenek ; Čeppek, Miroslav (ur.). Prague : Czech Technical University in Prague, 2010.
- [16] Egerstedt, M.: "Control of Mobile Robots", Lecture Notes, Georgia Institute of Technology, 2014.
- [17] Egerstedt, M.: "Swarming robots", Snapshot of modern mathematics, Oberwolfach press, No. 1, 2016.



- 
- [18] Egerstedt, M., Mesbahi, M.: " Graph Theoretic Methods in Multiagent Networks", Princeton University Press, 2010.
- [19] Beard, W., Young, B.: "A decentralized approach to formation maneuvers", IEEE Transactions on Robotics and Automation, vol. 19, no. 6, pp. 933-941, 2003.
- [20] Ren, W.: "Multi-vehicle consensus with time-varying reference state", System & Control Letters 56, pp. 474-483, 2007

## PRILOG A - UPUTE ZA KORIŠTENJE APLIKACIJE

U ovom prilogu dat ćemo upute za korištenje aplikacije. Aplikacija je zamišljena kao alat za buduće studente Fakulteta strojarstva i brodogradnje koji se žele upoznati s područjem mobilne robotike. U njoj su programirane, u programu Matlab, gotovo sve simulacije prikazane u ovom radu. Ona omogućuje da korisnici na jednostavan način izrade programirane simulacije te prave njihove usporedbe, mijenjajući parametre ili početne uvjete. Za korištenje aplikacije nije potrebno predznanje iz područja mobilne robotike, potrebno je tek malo predznanje iz korištenja programa Matlab. U aplikaciji je moguće simuliranje proizvoljnog broja robota, ali se preporuča da ih ne bude više od 10, radi jednostavnosti i kvalitete prikaza.

Budući da je simulacija napravljena čisto na programerski način, za bolje shvaćanje događanja u simulaciji preporučljivo je koristiti se i ovim radom.

### 1. Pokretanje aplikacije

Nakon pokretanja aplikacije potrebno je unijeti određeni broj podataka kako bismo pravilno definirali simulaciju koju želimo da nam aplikacija nacrti. Nakon unošenja određenog podatka potrebno je stisnuti tipku Enter. Vrlo je važno da se svi podatci pravilno unesu, tako da se simulacija može izvesti.

Postoje dva glavna pitanja koja se postavljaju kad se aplikacija pokrene. Prvo, aplikacija traži da se odredi želi li korisnik simulaciju kinematike (ako da, onda se unese broj 1) ili simulaciju dinamike (ako da, onda se unese broj 2). Nakon toga slijedi drugo pitanje, koju simulaciju korisnik želi. Opet se unese broj koji označava simulaciju koju korisnik želi, a brojevi su: 1 za praćenje, 2 za kružno praćenje, 3 za susretanje, 4 za uspostavljanje formacije, 5 za linijsko gibanje u formaciji i 6 za gibanje formacije po krivulji.

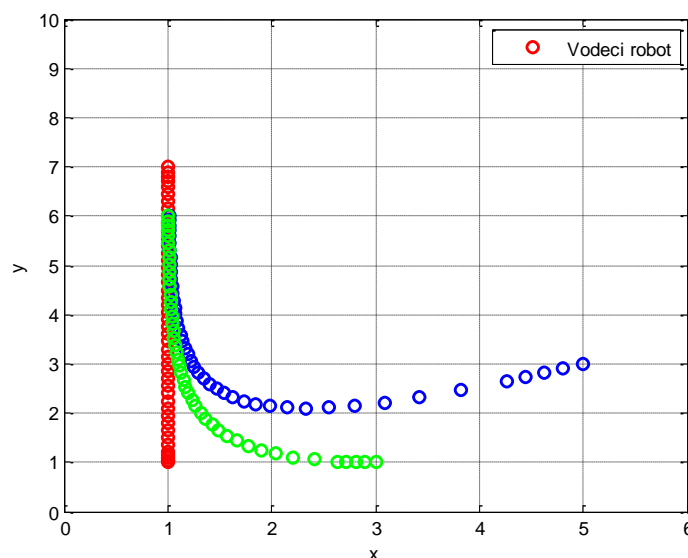
Nakon prva dva pitanja slijede druga pitanja koja pripadaju samim simulacijama. Sada ćemo proći kroz simulacije i objasniti što se u kojem pitanju traži te na koji način treba upisati traženo.

## 2. Kinematika

### Praćenje:

- **Unesite brzinu  $v_x$  i  $v_y$  vodećeg robota:** - ovdje se traži unos brzine u smjeru osi  $x$  i smjeru osi  $y$  vodećeg robota. Na primjer  $[0 ; 1]$  znači da nema brzine u smjeru osi  $x$ , nego samo u smjeru osi  $y$ .
- **Unesite zeljeni broj pratećih robota:** - ovdje se traži broj robota koji prate glavnog robota. Na primjer 2.
- **Unesite početne pozicije robota:** - ovdje se traži da se definiraju početne pozicije pratećih robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[5,3,3,1]$  a to označava da je prvi prateći robot na poziciji  $(5,3)$  a drugi na poziciji  $(3,1)$ . Pozicija vodećeg robota je fiksna radi jednostavnosti.
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,6,0,10]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 6 na  $x$  osi te od 0 do 10 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 6 što znači da će simulacija trajati 6 sekundi.

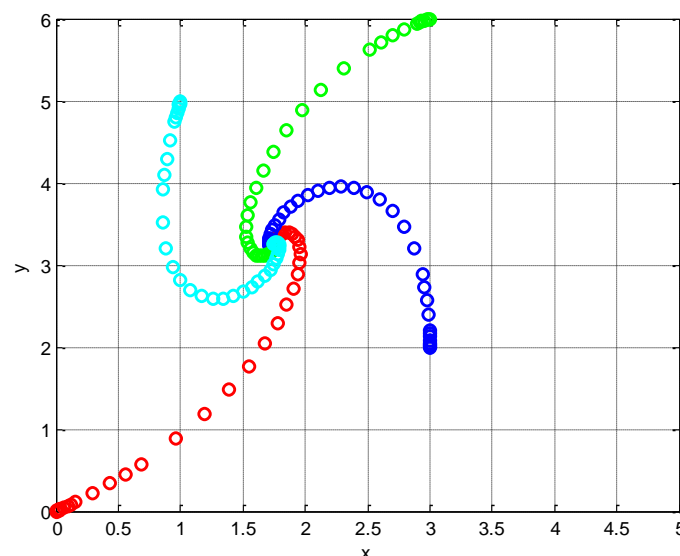
Nakon upisivanja svega traženoga aplikacija će izvesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Kružno praćenje:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. 3 je minimalni broj robota koje simulacija može sadržavati da bi ona imala smisla. Za naš primjer broj robota će biti 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[0,0,3,2,3,6,1,5]$  a to označava da je prvi robot na poziciji  $(0,0)$ , drugi na poziciji  $(3,2)$ , treći na  $(3,6)$  i četvrti na  $(1,5)$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,5,0,6]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 5 na  $x$  osi te od 0 do 6 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 10 što znači da će simulacija trajati 10 sekundi.

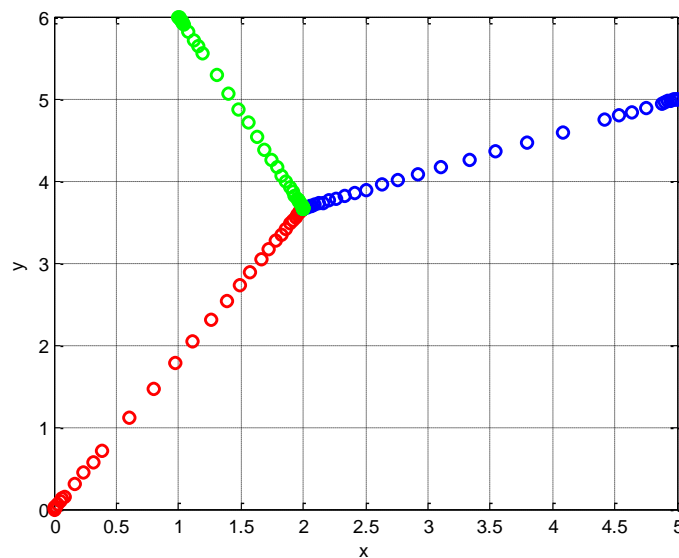
Nakon upisivanja svega traženoga aplikacija će izesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Susretanje:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 3.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[0,0,5,5,1,6]$  a to označava da je prvi robot na poziciji  $(0,0)$ , drugi na poziciji  $(5,5)$ , a treći na  $(1,6)$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,5,0,6]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 5 na  $x$  osi te od 0 do 6 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 10 što znači da će simulacija trajati 10 sekundi.

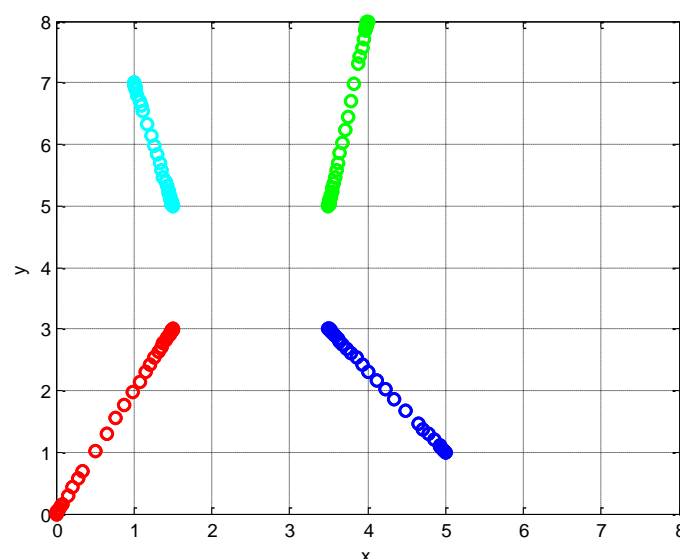
Nakon upisivanja svega traženoga aplikacija će izesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Uspostavljanje formacije:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[0,0,5,1,4,8,1,7]$  a to označava da je prvi robot na poziciji  $(0,0)$ , drugi na poziciji  $(5,1)$ , treći na  $(4,8)$ , a četvrti na  $(1,7)$ .
- **Unesite željni izgled formacije (željene koordinate):** - ovdje se unose željene krajnje koordinate koje definiraju izgled formacije. Na primjer za formaciju kvadrat možemo odabrati  $[2,2,4,2,4,2,4]$  što znači da su željene koordinate točke  $(2,2)$ ,  $(4,2)$ ,  $(4,4)$  i  $(2,4)$ , a to odgovara kvadratu s duljinom stranice 2.
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,8,0,8]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 8 na  $x$  osi te od 0 do 8 na  $y$  osi.
- **Unesite željeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 10 što znači da će simulacija trajati 10 sekundi.

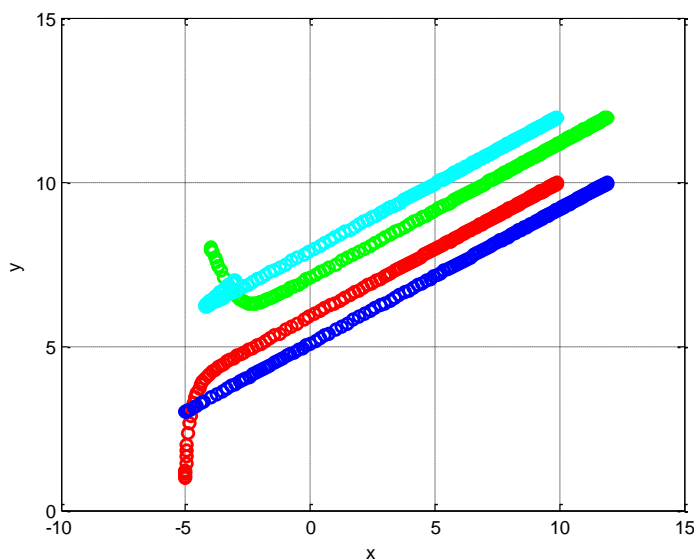
Nakon upisivanja svega traženoga aplikacija će izvesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



### Linijsko gibanje u formaciji:

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[-5,1,-5,3,-4,8,-3,7]$  a to označava da je prvi robot na poziciji  $(-5,1)$ , drugi na poziciji  $(-5,3)$ , treći na  $(-4,8)$ , a četvrti na  $(-3,7)$ .
- **Unesite željni izgled formacije (željene koordinate):** - ovdje se unose željene krajnje koordinate koje definiraju izgled formacije. Na primjer za formaciju kvadrat možemo odabrati  $[10,10,12,10,12,12,10,12]$ .
- **Unesite pojačanja sustava  $K_g$  i  $K_f$ :** - ovdje definiramo što nam je važnije, doći do cilja ili uspostaviti formaciju. Veći  $K_g$  znači da je važnije doći do cilja, dok veći  $K_f$  znači da je važnije uspostaviti formaciju. Na primjer  $[0.5, 5]$ , što znači da je  $K_g = 0.5$ , a  $K_f = 5$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Za naš primjer odaberemo:  $[-10,15,0,15]$  znači da će simulacija prikazati zbivanja u tom prostoru, od  $-10$  do  $15$  na  $x$  osi te od  $0$  do  $15$  na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 10 sekundi.

Nakon upisivanja svega traženoga aplikacija će izesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:

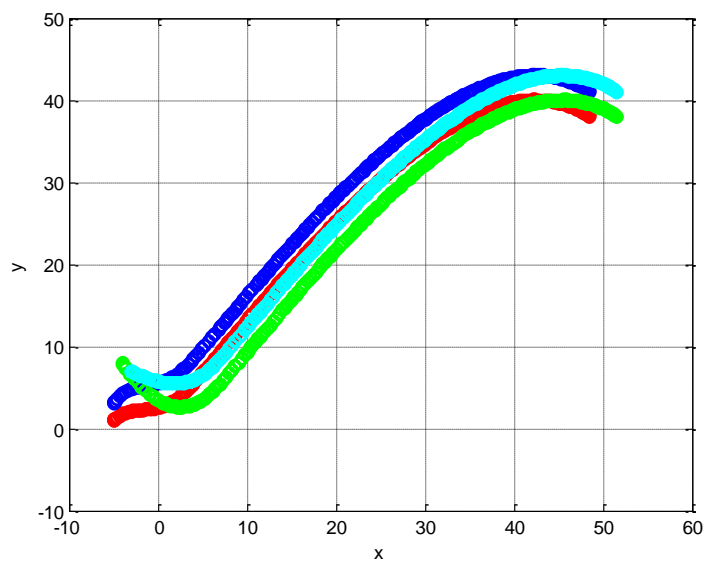


**Gibanje formacije po krivulji:**

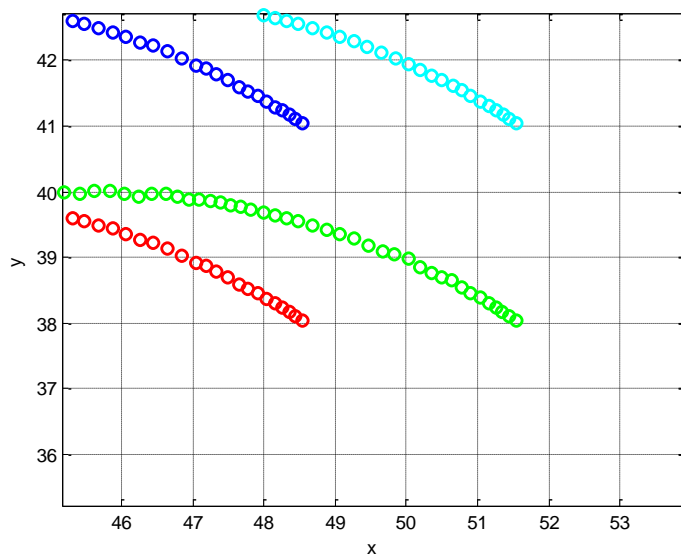
- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[-5,1,-5,3,-4,8,-3,7]$  a to označava da je prvi robot na poziciji  $(-5,1)$ , drugi na poziciji  $(-5,3)$ , treći na  $(-4,-2)$ , a četvrti na  $(-3,-4)$ .
- **Unesite početne točke krivulja (izgled formacije):** - ovdje se unose željene početne koordinate referentnih trajektorija koje definiraju izgled formacije. Na primjer ako želimo da se roboti gibaju u formaciji kvadrat možemo odabrati  $[0,0,0,3,3,0,3,3]$ .
- **Unesite željene parametre krivulje (X i Y):** - ovdje se traži unos parametara referentne krivulje, X označava maksimalni put u smjeru osi  $x$ , Y maksimalni put u smjeru osi  $y$ . U našem će primjeru biti  $[60,40]$ , što znači da je  $X=60$ ,  $Y=40$ .
- **Unesite pojačanja sustava B i K:** - ovdje definiramo sa B važnost praćenja referentne trajektorije, dok je K važnost uspostavljanja formacije. Važno je napomenuti da se povećanjem B povećava i održivost formacije, tako da ne trebamo povećavati oba parametra. Na primjer, sasvim je dovoljno da parametri iznose:  $B = 2$ , a  $K = 1$ . U našem primjeru će oni biti  $[2,1]$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Za naš primjer odaberemo:  $[-10,60,-10,50]$  znači da će simulacija prikazati zbivanja u tom prostoru, od  $-10$  do  $60$  na  $x$  osi te od  $-10$  do  $50$  na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 30 sekundi.



Nakon upisivanja svega traženoga aplikacija će izvesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



Detalj formacije izgleda ovako:

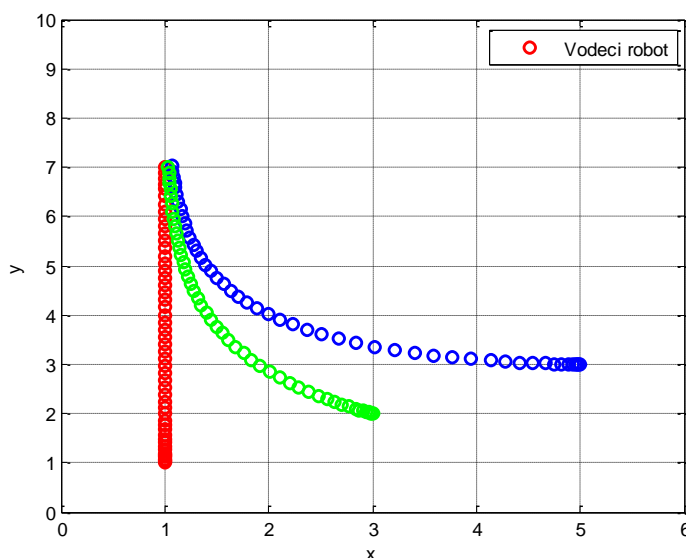


### 3. Dinamika

#### Praćenje:

- **Unesite brzinu  $v_x$  i  $v_y$  vodećeg robota:** - ovdje se traži unos brzine u smjeru osi  $x$  i smjeru osi  $y$  vodećeg robota. Na primjer  $[0, 1]$  znači da nema brzine u smjeru osi  $x$ , nego samo u smjeru osi  $y$ .
- **Unesite željeni broj pratećih robota:** - ovdje se traži broj robota koji prate glavnog robota. Na primjer 2.
- **Unesite početne pozicije robota:** - ovdje se traži da se definiraju početne pozicije pratećih robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[5,3,3,2]$  a to označava da je prvi prateći robot na poziciji  $(5,3)$  a drugi na poziciji  $(3,2)$ . Pozicija vodećeg robota je fiksna radi jednostavnosti.
- **Unesite prigušenje sustava  $K$ :** - ovdje treba unijeti željeno prigušenje sustava koje direktno utječe na gibanje robota. Na primjer, uzet ćemo da je vrijednost  $K$  jednaka  $[2]$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,6,0,10]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 6 na  $x$  osi te od 0 do 10 na  $y$  osi.
- **Unesite željeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 6 sekundi.

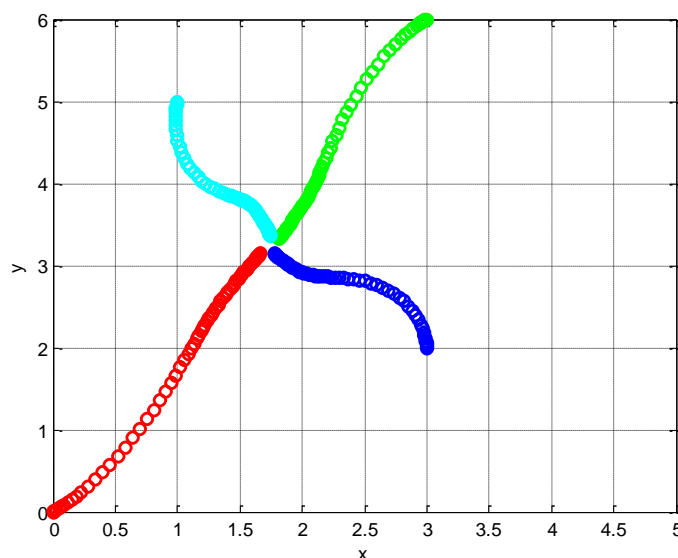
Nakon upisivanja svega traženoga aplikacija će izvesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Kružno praćenje:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. 3 je minimalni broj robota koje simulacija može sadržavati da bi ona imala smisla. Za naš primjer broj robota će biti 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[0,0,3,2,3,6,1,5]$  a to označava da je prvi robot na poziciji  $(0,0)$ , drugi na poziciji  $(3,2)$ , treći na  $(3,6)$  i četvrti na  $(1,5)$ .
- **Unesite prigušenje sustava  $K$ :** - ovdje se traži definiranje prigušenja sustava  $K$ , a on direktno utječe na ponašanje robota. Za naš primjer uzet ćemo da je jednak  $[2]$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,5,0,6]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 5 na  $x$  osi te od 0 do 6 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 6, što znači da će simulacija trajati 6 sekundi.

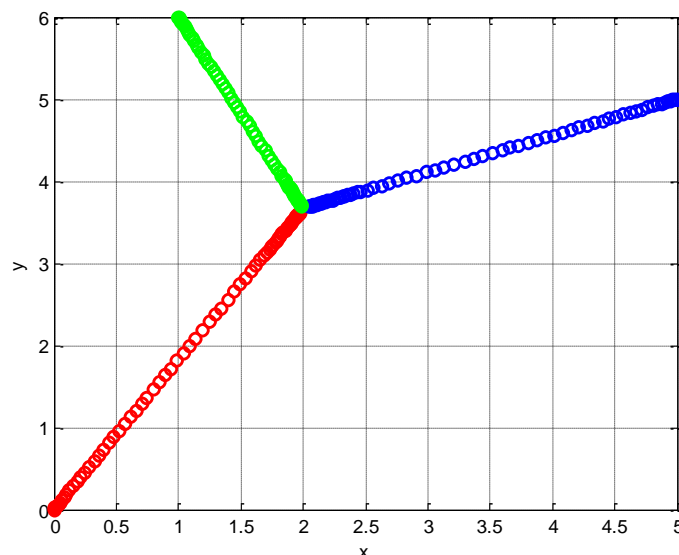
Nakon upisivanja svega traženoga aplikacija će izesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Susretanje:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 3.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[0,0,5,5,1,6]$  a to označava da je prvi robot na poziciji  $(0,0)$ , drugi na poziciji  $(5,5)$ , a treći na  $(1,6)$ .
- **Unesite prigušenje sustava K:** - ovdje se traži definiranje prigušenja sustava  $K$ , a on direktno utječe na ponašanje robota. Za naš primjer uzet ćemo da je jednak  $[2]$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,5,0,6]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 5 na  $x$  osi te od 0 do 6 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 8, što znači da će simulacija trajati 8 sekundi.

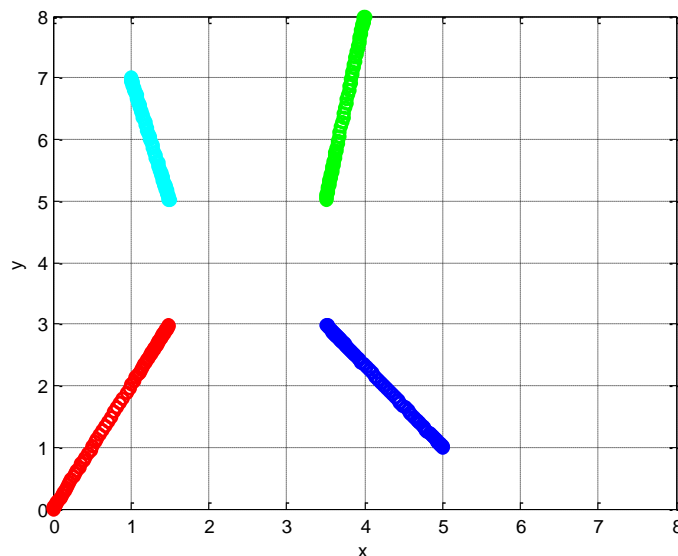
Nakon upisivanja svega traženoga aplikacija će izvesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Uspostavljanje formacije:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[0,0,5,1,4,8,1,7]$  a to označava da je prvi robot na poziciji  $(0,0)$ , drugi na poziciji  $(5,1)$ , treći na  $(4,8)$ , a četvrti na  $(1,7)$ .
- **Unesite željni izgled formacije (željene koordinate):** - ovdje se unose željene krajnje koordinate koje definiraju izgled formacije. Na primjer za formaciju kvadrat možemo odabrati  $[2,2,4,2,4,2,4]$  što znači da su željene koordinate točke  $(2,2)$ ,  $(4,2)$ ,  $(4,4)$  i  $(2,4)$ , a to odgovara kvadratu s duljinom stranice 2.
- **Unesite prigušenje sustava K:** - ovdje se traži definiranje prigušenja sustava  $K$ , a on direktno utječe na ponašanje robota. Za naš primjer uzet ćemo da je jednak  $[2]$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Na primjer:  $[0,8,0,8]$  znači da će simulacija prikazati zbivanja u tom prostoru, od 0 do 8 na  $x$  osi te od 0 do 8 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 10, što znači da će simulacija trajati 10 sekundi.

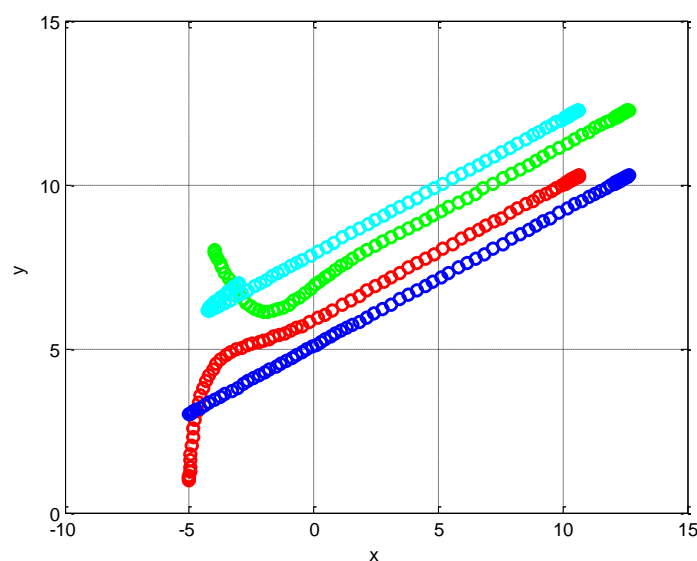
Nakon upisivanja svega traženoga aplikacija će izesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



**Linijsko gibanje u formaciji:**

- **Unesite zeljeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 4.
- **Unesite pocetne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[-5,1,-5,3,-4,8,-3,7]$  a to označava da je prvi robot na poziciji  $(-5,1)$ , drugi na poziciji  $(-5,3)$ , treći na  $(-4,8)$ , a četvrti na  $(-3,7)$ .
- **Unesite željni izgled formacije (željene koordinate):** - ovdje se unose željene krajnje koordinate koje definiraju izgled formacije. Na primjer za formaciju kvadrat možemo odabrati  $[10,10,12,10,12,12,10,12]$ .
- **Unesite pojačanja sustava  $K_g, D_g, K_f$  i  $D_f$ :** - ovdje definiramo što nam je važnije, doći do cilja ili uspostaviti formaciju. Veći  $K_g$  i  $D_g$  znači da je važnije doći do cilja, dok veći  $K_f$  i  $D_f$  znači da je važnije uspostaviti formaciju. Za naše primjere je dovoljno da mijenjamo  $K_g$  i  $K_f$ , dok  $D_g$  i  $D_f$  mogu ostat na 1. Za primjer ćemo uzeti  $[0.5,1,5,1]$ , što znači da je  $K_g = 0.5$ , a  $K_f = 5$ , dok su  $D_g$  i  $D_f$  jednaki 1.
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Za naš primjer odaberemo:  $[-10,15,0,15]$  znači da će simulacija prikazati zbivanja u tom prostoru, od -10 do 15 na  $x$  osi te od 0 do 15 na  $y$  osi.
- **Unesite zeljeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 10 sekundi.

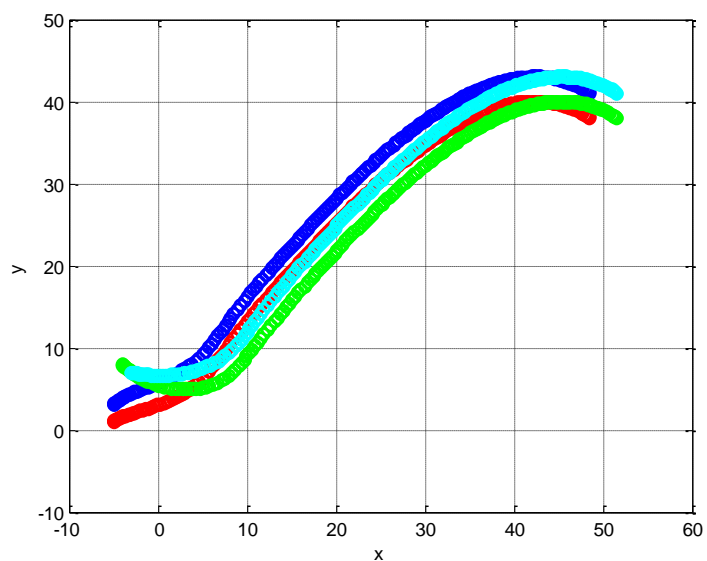
Nakon upisivanja svega traženoga aplikacija će izvesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



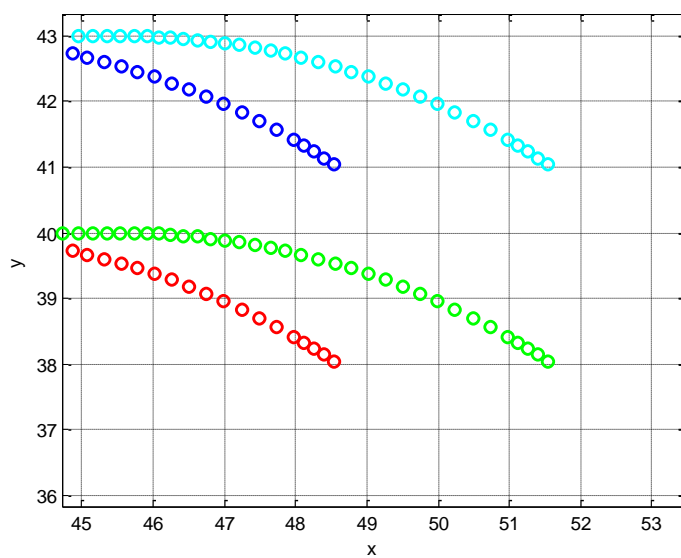
**Gibanje formacije po krivulji:**

- **Unesite željeni broj robota:** - ovdje se traži broj robota koji sudjeluju u simulaciji. Na primjer 4.
- **Unesite početne pozicije robota:** - ovdje se traži da se definiraju početne pozicije robota. Treba definirati poziciju na  $x$  i na  $y$  osi. Dakle, za svaki robot dva broja. Na primjer:  $[-5,1,-5,3,-4,8,-3,7]$  a to označava da je prvi robot na poziciji  $(-5,1)$ , drugi na poziciji  $(-5,3)$ , treći na  $(-4,-2)$ , a četvrti na  $(-3,-4)$ .
- **Unesite početne točke krivulja (izgled formacije):** - ovdje se unose željene početne koordinate referentnih trajektorija koje definiraju izgled formacije. Na primjer ako želimo da se roboti gibaju u formaciji kvadrat možemo odabrati  $[0,0,0,3,3,0,3,3]$ .
- **Unesite željene parametre krivulje (X i Y):** - ovdje se traži unos parametara referentne krivulje, X označava maksimalni put u smjeru osi  $x$ , Y maksimalni put u smjeru osi  $y$ . U našem će primjeru biti  $[60,40]$ , što znači da je  $X=60$ ,  $Y=40$ .
- **Unesite pojačanja sustava B, K i P:** - ovdje definiramo sa B važnost praćenja referentne trajektorije, K definira važnost uspostavljanja formacije, dok sa P definiramo važnost izjednačavanja brzina. Važno je napomenuti da se povećanjem B povećava i održivost formacije, tako da ne trebamo povećavati sve parametre. Na primjer sasvim je dovoljno da parametri iznose:  $B = 2$ ,  $K = 1$  i  $P = 1$ . U našem primjeru će oni biti  $[2,1,1]$ .
- **Unesite dimenzije koordinatnih sustava:** - tu treba unijeti veličinu koordinatnog sustava. Preciznije, definiramo prozor za crtanje. Za naš primjer odaberemo:  $[-10,60,-10,50]$  znači da će simulacija prikazati zbivanja u tom prostoru, od  $-10$  do  $60$  na  $x$  osi te od  $-10$  do  $50$  na  $y$  osi.
- **Unesite željeno trajanje simulacije:** - tu upisujemo željeno trajanje simulacije. Na primjer 30 sekundi.

Nakon upisivanja svega traženoga aplikacija će izesti simulaciju koja će za naš ovdje pretpostavljeni primjer izgledati ovako:



Detalj formacije izgleda ovako:





## **PRILOG B - CD**

1. PROGRAMSKI KOD APLIKACIJE
2. PROGRAMSKI KOD SIMULACIJA