2. Mozgó objektum követése
Feladat specifikáció:
Adottak:

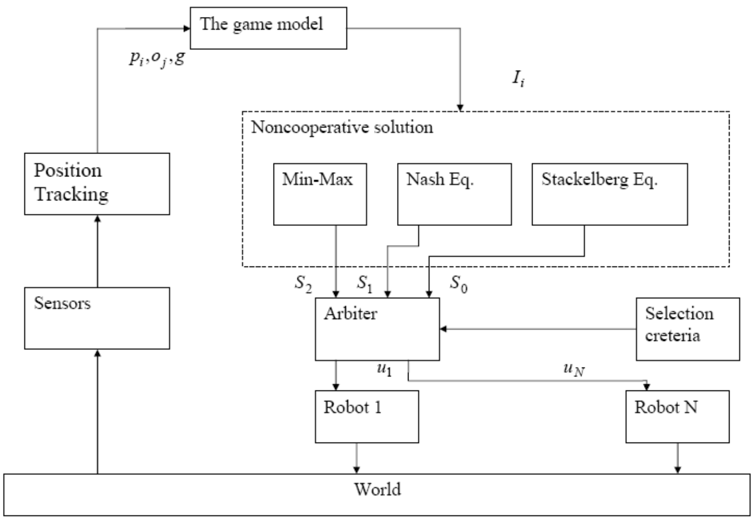
1. Az N darab Differenciális meghajtású anholonom mobil robotból álló csapat tagjainak pozíciója, orientációja a t_n diszkrét időpontban
2. Statikus (álló) akadályok pozíciója a terepen.
3. Dinamikus (mozgó) célpont pozíciója a t_n diszkrét időpontban.

Beavatkozás (döntések): Az N csapattag mozgását előíró sebességek:
Az i. robot szögsebessége
Az i. robot iránymenti sebessége

Cél: Az N csapattag mozgását illetően olyan döntések kialakítása, amely biztosítja, hogy

1. a csapattagok „tömegközéppontja ” kövesse a g célpontot.
2. A csapattagok a tömegközépponttól egy R sugarú körön belül legyenek.
3. Az ütközést elkerülendő bármely két csapattag ne legyen közelebb egymáshoz, mint egy előírt *Dro*.
4. A csapattagok előírt formációban legyenek.
5. A csapattagok ne ütközzenek akadályokba.

Az irányítás blokkdiagramja:



A feladat diszkretizálása:
Cél:
A problématér dimenziójának szűkítése. Játékelméleti szempontból csak így lesz kezelhető a megoldás számítása.
Diszkretizálandók: A robotok döntései (a “beavatkozó jelek”)
A robot szögsebességének diszkretizálása:

$$\boxed{\Omega_i = \left[\omega_i^1, \omega_i^2, \cdots \omega_i^{K_i}\right] \quad i = 1, 2, \cdots N}$$

Praktikus: Ha *Ki=3*, azaz , engedünk egy jobbrafordulást (negatív előjel), egy 0 szögsebességet és egy balrafordulást (pozitív előjel).
Az robot iránysebessége nagyon durva felbontást eredményezne → Nem diszkretizáljuk. .
Ehelyett: egzaktul számítunk egy célpontkövető sebeséget, amely
• célpont irányába mutat
• nagysága a célponttól való távolságtól függ

$$\boxed{v_{i,g} = v_{g,est} + \frac{d_{i,g}(v_{g,est})}{\Delta t} \frac{1}{1 + e^{-\alpha(d_{i,g}(v_{g,est}) - R_0)}}}$$

Az iránymenti sebesség komponensei:

- $v_{g,est}$ A a célpont becsült sebessége, amelyet a célpont előző F mintavételben felvett sebességeinek átlaga adja:

$$v_{g,est} = \frac{1}{F} \sum_{k=1}^{F} v_g(t_{n-k})$$

- $d_{i,g}(v_{g,est})$ Az i. robot és a célpont közötti jósolt távolság a t_{n+1} mintavételben (ha feltesszük, hogy a célpont $v_{g,est}$ sebességgel mozog tovább)
- α: a célpontkövető együttható. Ő határozza meg, hogy a $d_{i,g}(v_{g,est}) / \Delta t$ célpontkövető sebességkomponens milyen mértékben befolyásolja a robot sebességét. Meghatározza, hogy a célpontkövetés mennyire legyen "sima".

Az i. robot iránysebességének végső alakjához csapattulajdonságot kell megvalósítani:
A lemaradó robotok gyorsuljanak, a célhoz közeli robotok lassuljanak →

Csillapítási tényező:

$$\xi_i = \frac{\rho_{c,i}}{\max_{i=1,2,\cdots N} \ \rho_{c,i}}$$

$$\rho_{c,i} = \begin{cases} d_{c,i}/R_0 & \text{if } \ d_{c,i} > R_0 \\ 1 & \text{otherwise} \end{cases}$$

$d_{c,i}$ az i. robot és a tömegközéppont közötti távolság

Az robot iránysebességének végső alakja:

$$\boxed{v_i = \xi_i v_{i,g} \qquad i = 1, 2, \cdots N}$$

Az robot döntési halmaza (lehetséges beavatkozások halmaza):

$$\boxed{U_i = \{(\omega_i, v_i) : \ \omega_i \in \Omega_i \cup v_i\}}$$

Költségfüggvények

Cél: Kifejezni, hogy egy adott állapot mennyire jó a célkövetés és formáció szempontjából

Az i. robot költségfüggvényének általános alakja:

$$I_i(d_1, d_2, \cdots, d_N) = f_i(d_1, \cdots d_N, o_1, \cdots, o_M, g)$$

Az i. robot költségfüggvényének komponensei:

$$I_i(d_1, \cdots, d_N) = I_i^1 + I_i^2 + I_i^3 + I_i^4$$

I_i^1: az ütközés büntetése más robottal, akadállyal, célponttal

$$I_i^1 = (d_1, \cdots, d_N) = k_1 \frac{1}{\min\limits_{\substack{j=1,\cdots,N \\ k=1,\cdots,M \\ j \neq i}} (\hat{d}_{r,i,j}, \hat{d}_{o,i,k}, \hat{d}_{i,g})}$$

I_i^2 :A robot és robotcsapat célpontja közötti távolságának büntetése

$$I_i^2(d_1, \cdots, d_N) = k_2 \hat{d}_{c,i}$$

I_i^3:A robotcsapat középpontjának célponttól való távolságának büntetése

$$I_i^3(d_1, \cdots, d_N) = k_3 \hat{d}_{c,g}$$

I_i^4 :A robotcsapat előírt formációtól való eltérésének büntetése

A játékelméleti probléma megoldása

Cél: Minden $t=n\Delta t$, időben olyan döntéseket talalni a robotcsapat tagjainak számára, hogy a formációban való követést megvalósítsák. Ehhez a költségfüggvényeiket kell minimalizálni.

Q: Feltétlen létezik olyan döntéscsoport, ami minden robotnak jó?

A: NEM! Könnyen lehet, hogy egy robot döntése, ami önmagának jó, az másik robotnak rossz (mert pl közel kerül hozzá). → A robotok egymással versengenek.

Nash egyensúly

A Nash egyensúlyt valósítanak meg a robotok azon döntései, amelytől egyik robotnak sem érdemes eltérnie, mert rosszabbul jár.

A robotok döntései a *.k* diszkrét időpontban Nash egyensúlyt valósítanak meg, ha

$$\begin{aligned}
I_1\left(d_1^{k_{10}}, d_2^{k_{20}}, \cdots d_N^{k_{N0}}\right) \quad &\leq \quad I_1\left(d_1^{k_1}, d_2^{k_{20}}, \cdots d_N^{k_{N0}}\right) \\
&\vdots \\
I_N\left(d_1^{k_{10}}, d_2^{k_{20}}, \cdots d_N^{k_{N0}}\right) \quad &\leq \quad I_N\left(d_1^{k_{10}}, d_2^{k_{20}}, \cdots d_N^{k_N}\right)
\end{aligned}$$

Stackelberg egyensúly

A Stackelberg egyensúlynál feltehető, hogy a robotok között egyfajta hierarchia van. Ez azt jelenti, hogy a felsőbb szintjén álló robot (vezető) előre bejelentheti a döntését a hierarchiában alatta lévő robotok számára.

Vezetőnek mindig a legrosszabb helyzetben található robotot érdemes kinevezni, azaz, akinek az adott szituációban legnagyobb a költsége.

Min-max stratégia

Minden robot a legrosszabbat, ellenséges viselkedést feltételez a többi robotról és ebben a szituációban alakítja ki a legjobb döntését. Ilyen stratégia mindig létezik.

Akkor alkalmazható, ha Nash egyensúlynak megfelelő stratégiát szeretnénk megvalósítani, azonban a Nash egyensúly nem létezik.

Döntések több egyensúly létezése esetén

Több, mondjuk Nash-egyensúly létezik. Az egyensúlyok közül az kerül kiválasztásra, amelyikre igaz, hogy a lehető legegyenletesebben osztja szét a költségeket a robotok között.

3. Robotfoci stratégiák. Az UvaTriLearn stratégia. A csapatmozgás tervezésének megvalósítási szintjei, jellemzők. A koordinációs probléma, a játék menete, az ágensek céljai, koordinációs gráfok, Változó Eliminálási Algoritmus, a kifizet_dési (payoff) értékek meghatározása értékszabályok alkalmazásával, szereposztás szabad kommunikáció és annak hiánya esetén. Teljesen megfigyelhet_, nem kommunikáló UvaTriLearn csapat leírása; szerepek, szerepek sorozata, szerepek potenciálfüggvényei, döntések (akciók), felhasznált magas szint_ változók, stratégia értékszabályai (elvek).

Sok esetben három szint különíthető el:

1. Stratégia (magas) szint: Megmondja, hogy egyes csapattagoknak hova kell menni
2. Taktikai (közép) szint: Megmondja, hogy a stratégiai célt milyen pontok mentén érdemes elérni.
3. Mozgástervezési (alacsony) szint: Hogyan kell a robotot mozgatni, hogy a taktikai pontokat optimálisan érjük el (optimális irányítás)

Jellemzők:

- Az ágensek célja közös
- A kooperáció kulcsa: Koordináció
- Koordinációs probléma megoldásának elvi eszköze: Játékelmélet
    - A tiszta játékelméleti megközelítés módja: Következtetés az ágensek teljes keresési tere felett
    - Játékelméleti megközelítés hátránya: A keresési tér dimenziója exponenciálisan nő az ágensek számával → Sok ágens esetén önmagában nem praktikus →

- o A probléma struktúrája sok esetben lehetővé teszi a komplexitás csökkentését →

• A közös döntéstér (más szóval akciótér) méretének csökkentése: Koordinációs Gráf (KG)

- o Koordinácós gráf:
  - Csomópontok: ágensek
  - Élek: A két csomópontot összekötő él kifejezi, hogy a két ágensnek koordinálnia kell a döntéseit.

• Koncepció a közös optimális döntés eléréséhez:

I. A kontextus-specifikus KG frissitése dinamikusan az aktuális állapot alapján

II. Változó elimináció

- Változó elimináció lépései:
  1. Iteratívan megoldani a lokális koordinációs problémát
  2. Az eredményt átadni a gráf más részeinek

Feltevések:

• Egy ágens képes a környezetében lévő másik ágens optimális döntését is számolni (nem kell kommunikáció)

• Környezet: Folytonos, dinamikus, érzékelhető

A koordinációs probléma:

Stratégiai játék: Egy halmazcsoport $\langle n, A_1, ..., A_n, R_1, ..., R_n \rangle$

Ahol

n Az ágensek száma

A_i Az i. ágens döntéseinek (akcióinak) halmaza

R_i Az i. ágens jósági függvénye (payoff), formálisan:

$R_i(A) \rightarrow \mathbb{R}$ ahol $A = A_1 \times, \ldots, \times A_n$ az együttes akciók halmaza

A játék menete: Az ágensek egymástól függetlenül választanak egy akciót a döntési terükből, majd minden ágens az együttes akciók függvényében a jósági függvényének megfelelően "jutalmat" kap.

Az ágensek célja: Olyan egyéni döntéseket hozni, hogy a legjobban profitáló együttes akciót valósítsák meg. n

A stratégiai játék megoldási lehetőségei pl. a Nash, vagy a Pareto egyensúly megtalálása.

A koordinációs gráf:

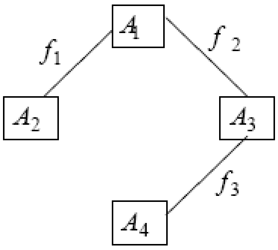
- Csomópontok: ágensek
- Élek: A két csomópontot összekötő él kifejezi, hogy a két ágensnek koordinálnia kell a döntéseit.

Probléma: Multiágens rendszerek koordinálásának számítási ideje az együttes döntések terén az ágensek számával exponenciálisan nő.

Megoldási javaslat:

Csak azokat az ágenseket koordináljuk, amelyek döntéseikkel egymásra hatással vannak →**Koordinációs gráf**

Feltételezés: A globális payoff (jósági) függvény a lokális jósági függvények lineáris kombinációja.



A globális koordinációs feladat = Több lokális koordinációs feladat együttese
A globális playoff fv. Dekompozíciója:

$$R(a) = f_1(a_1, a_2) + f_2(a_1, a_3) + f_3(a_3, a_4)$$

Változó eliminálási algoritmus (Variable Elimination Algorithm - VEA)

Koncepció: Az ágenseket egyesével elimináljuk, miközben végrehajtunk lokális maximalizációt.
Algoritmus:

1. Egy ágens szelektálása eliminációhoz
2. A szelektált ágens begyűjti a szomszédoktól azok payoff függvényeit
3. A szelektált ágens a szomszédok lehetséges döntéseit figyelembe véve feltételesen optimalizálja a döntését.
4. A szelektált ágens a feltételes payoff függvényét tudatja a szomszédaival.
5. A lépések ismételt végrehajtása egy következő ágens szelektálásával mindaddig, amíg az utolsó ágenst is elimináljuk.
6. Az utolsó ágens maximalizálja a döntésével a végső feltételes stratégiát

7. Az elimináció inverz sorrendjében az ágensek megtudják a később eliminált ágensek stratégiáit és kialakítják a saját optimális döntésüket a saját feltételes stratégiájuknak megfelelően.

[RS-UvA].2.2.A payoff értékek meghatározása értékszabályok (value rule) alkalmazásával

A payoff függvény lehet

- **Mátrix alakú**
- **Értékszabály**: (Kisebb számítási költség)

Döntéstér (korábbiaknak megfelelően) : $A = A_1 \times, \ldots, \times A_n$

Az i. **ágens döntése :** $a_i \in A_i$

Együttes döntés: $a \in A$

A diszkrét állapotok halmaza: X Ebből egy állapot: $x \in X$

A c **kontextus:** Az állapotok és döntések összes kombiációjának halmaza: $c \in C \subseteq X \cup A$

Az értékszabály: $\langle p;\ c : v \rangle$

Ez egy függvény : $p : C \to \mathbb{R}$ $c = (x, a)$

$$p(x, a) = v$$

Csak azok az értékszabályok befolyásolják a globális payoff függvényt, amelyek konzisztensek az aktuális kontextussal:

$$R(a) = \sum_{i=1}^{m} p_i(x, a)$$ Ahol m az értékszabályok száma

[RS-UvA].4.1. Szerepkiosztás szabad kommunikáció esetén

Feltételezés: A szerepkiosztási algoritmus ismert minden ágens számára

Algoritmus:

1. A szerepek M' sorozatának definiálása fontossági sorrendben. $|M'| \geq n$ ahol n az ágensek száma.
2. Minden A_i ágens kiszámítja az r_{im} potenciált. Az r_{im} potenciál megmondja, hogy Az A_i ágens mennyire alkalmas az $m \in M'$ szerep betöltésére, ahol m az M' sorozat egy eleme. (Részletek később). Ez $O(|M|n)$ számítás.
3. Minden ágens elküldi minden ágensnek a szerepekhez számított saját potenciáljait. Ez $O(|M|n)$ üzenet küldését jelenti.
4. A sorozat első $m \in M'$ szerepét ahhoz az A_i ágenshez rendeljük, amelyiknek legnagyobb az r_{im}, $i = 1, \ldots, n$ potenciálja. Ezt az ágenst a továbbiakban nem vesszük figyelembe.
5. A kiválasztást megismételjük az M' sorozat következő szerepére, amíg minden ágensre nem jut egy szerep. (általában az ágensek számánál kevesebb aktív szerep van, így a legutoljára szerepet kapó ágensek többnyire 'passive' szereppel passzívak lesznek, ők csak a stratégiai célpontjuk felé mozognak)

Megjegyzés: Minden ágensnek csak egy szerep jut, de egy szerep akár több ágensre is juthat. (Az M' sorozat több azonos szerepet is tartalmazhat)

Számítási igény redukáldik, mert:

- A szerepek a folytonos állapot absztrakciójának egy diszkrét kontextusát valósítják meg.
- A szerepek redukálják a döntéstér méretét. (Egy kapusnak nem lesz pl. kapuralövési döntése)→
- A lokális koordinációs gráfok csökkentik a tekintetbe vehető Nash egyensúlyok számát (ez jó).

[RS-UvA].4.2. Szerepkiosztás kommunikáció hiánya esetén

Fontos: A változó eliminálási algoritmus itt is számítható

Különbségek a kommunikációs esethez képest:

- Minden ágens minden ágens r_{im} potenciálját számítja. $O(|M|n)$ számítás
- A kommunikációs esetben minden ágens csak a saját állapotait mérte, a kommunikáció nélküli esetben az összes többi ágensét is kell.
- A számítási költség az egyes ágenseknél megnő. Javítható az algoritmus futásának párhuzamosításával.

Feltételezések a jó működéshez:

- Az A_i ágens payoff értékét minden A_i ágensel kapcsolatban lévő ágens ismeri.
  (Ez erős feltétel. Sok esetben még a kommunikációs esetben sem lehet teljesíteni)
- Minden ágens ki tudja számolni minden ágens r_{im} potenciálját
- A döntési sorrend ismeretes minden ágens számára (Ha több egyenértékű döntés, egyensúly van)
- Az A_i ágens minden, vele kapcsolatban lévő ágens állapotához hozzáfér (meg tudja figyelni).
  (Gyakran nem teljesül, mivel egy ágens csak egy szűk területet érzékel a világból)

[RS-UvA].5. Teljesen megfigyelhető, nem kommunikáló UvA Trilearn csapat

Ágensek száma (egy csapaton belül): 10+1

A kapus szerepe fix, szerepkiosztásnál nem foglalkozunk vele

Szerepek:

- Aktív. Ezen belül:
  - Letámadó (a labda rúgása nem valószínű, mert az ellenfél jobb helyzetben van)
  - Passzoló (a labdát birtokoljuk, meg tudjuk rúgni egy ilyen szerepet játszó ágenssel)
- Fogadó
- Passzív

Lehet még definiálni a kapust

Szerepek sorozata:

M'= {aktív, fogadó, fogadó, passzív, passzív, passzív, passzív, passzív, passzív, passzív}

A szerepek potenciálfüggvényei (Számításuk: Főleg taktikai szinten)

- Az aktív játékos potenciálja:

$$r_{i,active} = \frac{1}{t_i}$$

Ahol t_i az a becsült idő, ami alatt az A_i ágens eléri a labdát

Ha t_i idő alatt az ágens bele is tud rúgni a labdába (előbb, mint az ellenfél), akkor az ágens 'passzoló' szerepben játszik, ha nem tudja elérni, akkor 'letámadó' szerepben.

- A fogadó játékos potenciálja:

$$r_{i,receiver} = \begin{cases} 1/\max(1, d_{i,g}) + 1 & \text{if } d_{i,b} < k \\ 1/\max(1, d_{i,g}) & \text{otherwise} \end{cases}$$

Ahol $d_{i,b}$ az A_i ágens és a labda közötti távolság
$d_{i,g}$ az A_i ágens és az ellenfél kapuja közötti távolság

Azaz inkább annak a játékosnak passzolunk, aki közelebb van a kapuhoz. Külön jutalmazzuk, ha a labdahoz egy k távolságnál közelebb van az ágens.

- A passzív játékos potenciálja konstans:

$$r_{i,passive} = constans$$

Döntések (akciók). (Megvalósításuk taktikai szinten)

- ***Intercept***

  A labda 'elfogása'

- ***PassTo(i,dir):***

  A labda passzolása az A_i ágenstől, *dir* irányba fix távolságra. $dir \in D = \{center, n, nw, w, sw, s, se, e, ne\}$.
  A *dir* irányt a receiverhez képest kell értelmezni

- ***MoveTo(dir):***

  Mozgás *dir irányba*

- ***Dribble(dir):***

  A labda mozgatása *dir* irányba. Cselezés

- ***Score:***

  Kapuralövés az ellenfél kapujának számunkra legkedvezőbb pontjába.

- ***ClearBall:***

  Felszabadítórúgás az ellenfél játékosai (védői) között az ellenfél térfelére

- ***MoveToStratPos:***

  Mozgás stratégiai pozícióba. Ez a játékosok és a labda diszlokációjától függ.

Felhasznált magas szintű (boolean) változók

- ***is-pass-blocked(i,j,dir)***

  Megmondja, hogy egy pass az A_i ágens felől az A_j ágenstől *dir* irányú kis távolságra eső pontjába lehetséges-e, azaz nincs-e blokkolva más (ellenfél) játékosa által. (Kúppal definiálva)

- ***is-empty-space(i,dir)***

  Megmondja, hogy van-e ellenféltől mentes kissugarú szabad körterület az A_i ágenstől *dir* irányba.

- ***is-in-front-of-goal(i)***

  Megmondja, hogy az A_i ágens az ellenfél kapuja előtt van-e

Stratégia értékszabályai:

A legnagyobb payoff értéke a labda elfogásának, és a kapura lövésnek van (10). Ezen kívül 5 és 7 közötti a payoff értéke a passznak, attól függően, hogy a kaputól milyen messze kapja el a labdát a fogadó. Szintén 5 és 7 közötti a payoff értéke annak, hogy a potenciális fogadó a labda irányába mozdul, valamint hogy egy lehetséges fogadó felé mozduljon, hogy később ő kapja a labdát. Ha ezek közül egyiket sem tudja megtenni, akkor kettes payoffal vezet labdát, ha nála van, és szintén kettessel szabadít fel. Egyes payoffal mozog stratégiai pozícióba.

1. Multiágens_ rendszerek kooperációja statikus objektumnak formációban való megközelítésére. A feladat specifikációja, a rendszer leírása, a Hilare robot mozgásegyenlete, az alkalmazott koordinátarendszerek, a formációs (konfigurációs) gráf és a formáció vektor szerepe, a referencia (holonóm) robotrendszer sebessége, a robotkerekek irányítása, a célobjektumot elér_ man_ver feltétele, formáció irányítás.

[SCK].1. Feladat specifikáció

Adottak:

1. Csapattagok, amely n darab Hilare típusú anholonom mobil robotból áll
2. Statikus (álló) akadályok a terepen
3. Statikus (álló) célpont

Cél: Az n Hilare típosú robot (a csapat) mozgásának koordinálása úgy, hogy

A. a csapat formációját előírt állapotba lehessen irányítani

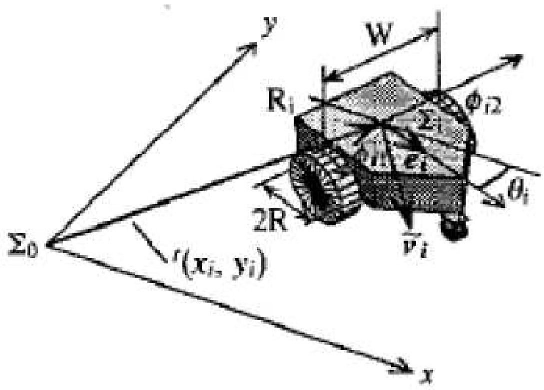
B. A célpont (betolakodó) megfelelő körbekerítésével a csapat védelmezzen meg egy területet

C. A formáció legyen asszimptotikusan (exponenciálisan) stabil.

D. A csapattagok ne ütközzenek akadályokba

[SCK].2. A rendszer leírása

- *Az R_i Hilare robot (azaz az i. csapattag) kinematikája:*



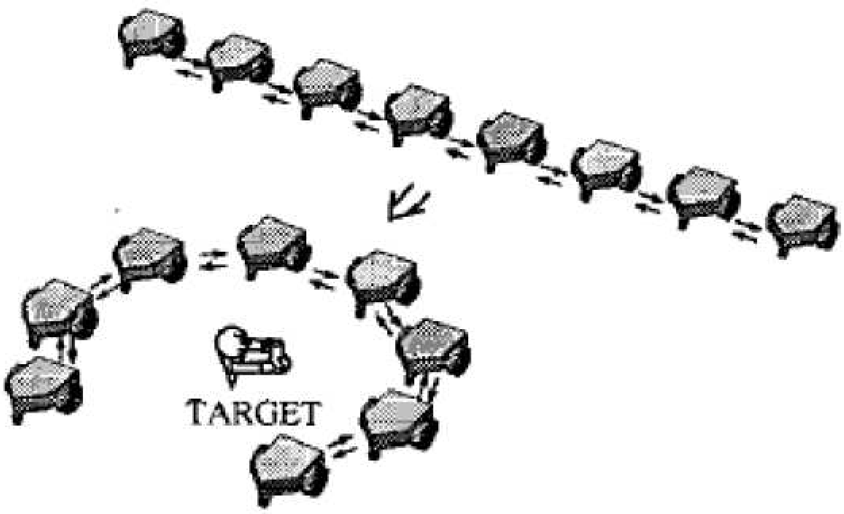
Mozgásegyenlet:

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \frac{R}{2}\cos\theta_i\left(\dot{\phi}_{i1} + \dot{\phi}_{i2}\right) \\ \frac{R}{2}\sin\theta_i\left(\dot{\phi}_{i1} + \dot{\phi}_{i2}\right) \\ \frac{R}{2W}\left(\dot{\phi}_{i1} - \dot{\phi}_{i2}\right) \end{pmatrix}$$

${}^t(x_i, y_i)$	Az R_i robot pozíciója a Σ_0 statikus KR-ben
θ_i	Az R_i robot orientációja a Σ_0 statikus KR-ben
ϕ_{i1}, ϕ_{i2}	A kerekek szöge

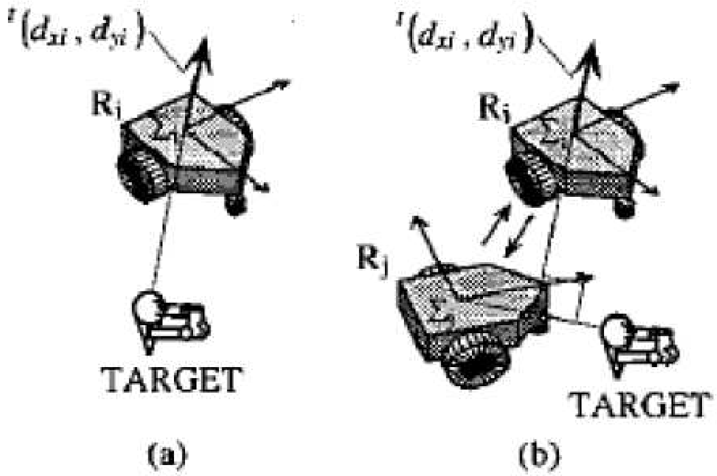
Σ_i	Az R_i robothoz rögzített KR
Σ_0	Statikus (világ) KR
Σ_i^0	KR, melynek pozícióját Σ_0, orientációját a Σ_i adja.

- *Erősen kapcsolt formációs (konfigurációs) gráf*



  - Minden robotnak két szomszédja van, a lánc szélső robotjait kivéve, akiknek csak egy.
  - Feltételezés: a formációs gráf statikus.
  - Az irányított él kezdeti csomópontjában található robotnak előírt relatív helyzetet kell tartania az él végponti csomópontjában található robottal. A relatív helyzetet indirekt módon a formációs vektor adja meg (a TARGET-hez képest).

- *Formáció vektor*



  - Ha egy robotnak két szomszédja van a formáció gráfban, akkor az *i*. robot $^t(d_{xi}, d_{yi})$ formáció vektora TARGET-*i*. robot irányú (a. ábra)
  - Ha egy robotnak csak egy szomszédja van, akkor a formációs vektor szomszédos robot- *i*. robot irányú (b. ábra).

Az R_i robot referencia sebessége (újra):

$$\tilde{v}_i^i = \begin{pmatrix} \tilde{v}_{xi}^i \\ \tilde{v}_{yi}^i \end{pmatrix} = \sum_{j \in L_i} \tau_{ij} \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} + \tau_i \begin{pmatrix} x_t^i \\ y_t^i \end{pmatrix} + \begin{pmatrix} d_{xi}^i \\ d_{yi}^i \end{pmatrix} + \sum_{j \in OBJECTS} \sigma_{ij} \left\{ \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} - D \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} \Big/ \left\| \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} \right\| \right\} \qquad \text{(R1)}$$

Egyes tagok fizikai szerepe:

$\tau_{ij} \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix}$	Az R_i robotot vonzza az R_j robot
$\tau_i \begin{pmatrix} x_t^i \\ y_t^i \end{pmatrix}$	Az R_i robotot vonzza a TARGET célobjektum
$\begin{pmatrix} d_{xi}^i \\ d_{yi}^i \end{pmatrix}$	Az R_i robotot a formációs vektor irányába „húzzuk”
$\sigma_{ij} \left\{ \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} - D \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} \Big/ \left\| \begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} \right\| \right\}$	Az R_i robotot mesterséges potenciáltérrel taszítják a közelébe kerülő (bármely) objektum.

A robotkerekek irányítása

1. **Azaz az R_i robot csak e_i irányban tud mozogni!**
2. **A robot nem tud a $\tilde{v}_i$ referencia sebesség irányában mozogni** (hacsak nem párhuzamos a kettő).

Javaslat:
A robot valódi v_i sebessége legyen a $\tilde{v}_i$ referencia sebesség e_i egységvektor rányába mutató projekciója, azaz:

$$\langle e_i, v_i \rangle = \langle e_i, \tilde{v}_i \rangle \qquad \text{(16-P1)}$$

Probléma: Ha $\tilde{v}_i \perp e_i$ akkor a robot nem mozdul, (hanem megáll). $\Rightarrow$

A θ_i orientációt időben előírjuk pl úgy, hogy egy P, PD, PID szabályozónak megfelelő exponenciális lecsengéssel közeledjen a kívánt orientációhoz (idővariáns ÁV): $\theta_i = \theta_i(t)$

A kerekek mozgatása (a beavatkozó jelek előállítása) Σ_0 KR-ben:

$$\boxed{\begin{aligned} \dot{\phi}_1 &= \left(\langle e_i, \tilde{v}_i \rangle + W\dot{\theta}_i(t)\right) / R \\ \dot{\phi}_2 &= \left(\langle e_i, \tilde{v}_i \rangle - W\dot{\theta}_i(t)\right) / R \end{aligned}}$$

A kerekek mozgatásának (16-C1) és (16-C2) számításához az R_i robot használhatja saját Σ_i KR-ét is :

$$\dot{\phi}_1 = \left(\left\langle e_i^i, \tilde{v}_i^i \right\rangle + W\dot{\theta}_i(t)\right) / R \tag{17-C1}$$

$$\dot{\phi}_2 = \left(\left\langle e_i^i, \tilde{v}_i^i \right\rangle - W\dot{\theta}_i(t)\right) / R \tag{17-C2}$$

Célobjektumot elérő manőver feltétele:

⇒ Az átlagolt rendszerben időben változó θ_i mellett az R_i robot megvalósíthat egy célobjektumot elérő manővert.

Formáció irányítás:

Ha bármely robot

- bármely robothoz viszonyított relatív távolsága ÉS
- bármely akadályhoz viszonyított relatív távolsága ÉS
- a TARGET célobjektumhoz viszonyított relatív távolsága

nagyobb mint a D küszöbérték $\rightarrow \delta_{ij} = 0,\ i = 1,2,\dots,n;\ j \in OBJECTS$ $\rightarrow$ nincs C, Λ, ξ (22-T1)-ben:

$$\begin{pmatrix} \dot{\theta}_x \\ \dot{\theta}_y \end{pmatrix} = \begin{pmatrix} B-\Delta & 0 \\ 0 & B-\Delta \end{pmatrix} \begin{pmatrix} \theta_x \\ \theta_y \end{pmatrix} - \begin{pmatrix} \dot{x}_t v_1 \\ \dot{x}_t v_2 \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{23-T2}$$

Mivel a robot csapat követi a TARGET-et, zárt környezetben a TARGET nem tud elmenekülni és végül csapdába esik. →