

Indirekt és direkt kommunikációt használó valós NXT robotok gyülekezési eljárásai

Pásztor Attila¹, Czuprák Zsolt²

^{1,2} Informatika Szakcsoport/GAMF Kar/ Kecskeméti Főiskola

Összefoglalás: A mesterséges intelligencia egy intenzíven vizsgált területe a robotcsoportok és robotrajok gyülekezési eljárásainak vizsgálata. A direkt kommunikációt használó robotok valamilyen rádió rendszert, fényjelzéseket vagy egyéb kommunikációs protokollt használva gyűlhetnek össze egy előre adott pont vagy tárgy körül vagy a mesterséges tér egy előre meg nem adott körében. Az indirekt kommunikációt használó rendszerek esetében a robotraj minden egyes egyede önálló entitásként dolgozik, a robotok csak a helyzetükön és a mozgásaikon keresztül nyújtanak társaiknak információt és így próbálnak minél szűkebb területre összegyűlni.

Számos kutató készített a fenti problémák megoldására elméleti algoritmusokat, de munkájukat főként csak képernyő szimulátorokkal tesztelték.

Az írásban valós NXT és Dual NXT robotok segítségével mutatunk be megoldásokat a fenti problémákra. Azt akartuk bizonyítani, hogy a gyenge kommunikációs és navigációs képességekkel ellátott robotok képességeit növelve, képessé lehet tenni őket, összetett gyülekezési eljárások szimulálására.

Abstract: An intensely investigated area of artificial intelligence is studying the gathering methods of robot swarms and robot teams. The robots can use the direct communication e.g. some kind of radio system, light signals or another communicational protocol and they may gather around an object or in a special circle of artificial area. In the case of those systems which use indirect communication all members of the robot swarm work as independent entity, the robots can provide information for partners by their positions and movements.

Several researchers made theoretical algorithms onto the solution of the above mentioned problems, but they used only mathematical or robot simulators to test their ideas.

Our article deals with some solution of gathering problems using NXT and Dual NXT robots. We wanted to demonstrate that we can increase the capabilities of those robots which have poor communication and navigation abilities so they will be suitable to simulate complex gathering methods.

Kulcsszavak: NXT és Dual NXT robotok, gyülekezési eljárások szimulációi, szenzorok kalibrálása.

Keywords: NXT and Dual NXT robots, simulations of gathering methods, calibration of sensors.

1. Bevezetés

Az utóbbi évtizedekben számtalan tudományos kutató vizsgálja a természetben szabadon élő állatok viselkedésmintáit, szokásait és megfigyeléseik eredményeit megpróbálják adaptálni valós robotokra. Kialakult a mesterséges intelligencia egy viszonylag

új kutatási területe a „raj intelligencia”. Az olyan decentralizált, önszerveződő rendszereteket, ahol az egyes egyszerű ágensek helyi szabályokat követve, az egyes ágensek között kölcsönhatások alakulhatnak ki, melyek a rendszer intelligens, globális viselkedéséhez vezethetnek, swarm intelligenciának szokás nevezni. A kifejezést Gerardo Beni és Jing Wang használta először a celluláris robotrendszerrel kapcsolatban [1]. A Swarm Intelligencia természetben megfigyelhető példái a hangyakolóniák viselkedése, a madárrajok csoportos repülése, a halak csoportos mozgása, néhány emlősállat csordaszelleme, a baktériumok növekedés mintái.

A „robot raj” és a „robot csoport” két különböző fogalommá vált. Az előbb említett robot raj Clough [2] meghatározásában a független egyedek érzékeléseinek és észleléseinek gyűjteménye, melyek reaktív kölcsönhatásaiból egy globális viselkedés alakul ki. A robot csoport tagjainak Kou-Chi Lin szerint [3] ezzel ellentétben tudatos magatartása van, ismerik saját szerepüket, ismerik a többi egyed szerepét és helyét a csoportban és tudják, hogyan viszonyuljanak egymáshoz, hogy a kitűzött feladatukat teljesítsék. Ha a robotok azonos felépítésűek, akkor homogén, különben heterogén robotrendszerrel beszélünk.

Tudományos vizsgálataikhoz és kutatásaihoz számtalan kutató az intelligens ágensek számítógépes szimulációját alkalmazza. Sokan használják erre a célra a Webots mobil robot szimulátort, a Repast általános ágens szimulátort vagy a Matlab programot. A szimulált eljárások legtöbb esetben költségkímélőek és segítséget nyújthatnak a probléma előzetes vizsgálatához, a hipotézisek felállításához. A szimulátorok azonban számtalan előnyük ellenére sem helyettesíthetik, csak kiegészíthetik, a valós robotokkal való kísérletezést. A valós kísérletek közben számtalan olyan „zavaró” tényező merülhet fel, amit a szimulátorokkal való feladatmegoldás során könnyen figyelmen kívül hagyhatunk.

Eltérően az előbbi kutatóktól, sokan alacsony költségű valós robotokat használnak kísérleteikben. James McLurkin [4] kísérleteiben is önszerveződő hangyarobotokból indult ki. Több mint 100 olyan robotot fejlesztett ki, melyek különböző érzékelőkkel, rádió-modemmel, audió-rendszerrel, öntöltő-berendezéssel voltak felszerelve és rajként cselekedtek. Stefano Nolfi vezetésével [5] kicsi kerek robotok fejlesztettek, melyek együttesen igyekeztek olyan feladatokat megoldani, amelyekre külön-külön képtelenek lennének. Marco Dorigo és munkatársai [6] a Swarm-bots európai uniós projekt keretében a megszokottól eltérő kinézetű mobil robotokat, „raj-botokat” (s-bots) fejlesztettek. Ez az önszerveződő kolónia, harmincöt, alacsony költségű, egyszerű, rovarformájú egyedből állt össze. Az egyszerű robotok összegyűltek markolóik segítségével egymáshoz tudtak kapcsolódni és így, egy-egy robot által kivitelezhetetlen küldetéseket a környezet felderítését, súlyosabb tárgyak nehéz terepen történő szállítását megoldó rendszerre alakulnak át. Mubarak Shah és Niels da Vitoria Lobo [7] a jelenlegiekénél lényegesen kisebb ember nélküli kémrepülőkön dolgoztak. A légi járművek koordinált mozgását és a rájuk bízott feladatok végrehajtásának módját az állatvilágból ellesett raj-intelligencia mintákat követve tervezik. A kutatók abból a tényből indultak ki, hogy a vándormadarak külön-külön bajosan, csoportban viszont sokkal könnyebben megoldják problémáikat. Carlo Ratti és az MIT kutatói a SeaSwarm projektben [8] olyan autonóm robotokat terveztek, amelyekben a Mexikói-öbölben a tengeren úszó robot-tutajok egymással kommunikálva derítik fel olajszennyeződéseket, összegyűlnek körülötte és együttesen tüntetik el a szennyeződést. A. Kushleyev és társai [9] olyan kisméretű quadcoptereket fejlesztett ki, ahol a quadcopterek együttműködve tudnak összegyűlni, alakzatban repülni, egymással kommunikálva pontosan tisztában vannak a másik helyzetével és mozgásával. A mozgásmintákat a csoportokban repülő madarak viselkedése adta.

2. Dual NXT robotcsoport gyülekezése adott tárgy köré, bluetooth kommunikációt használva

Írásunkban mi is valós robotokat, úgynevezett programozható modell robotokat használtunk robot csoportok és rajok gyülekezési problémájának vizsgálatára. Azt akartuk megmutatni, hogy ezekkel a valós, nem holonomikus, mobil robotokkal, mint az NXT és a Dual NXT, valós fizikai világot kialakító, mesterséges környezetben megvalósíthatóak különféle gyülekezési eljárások szimulációi. Ahhoz, hogy ezeket a szerény navigációs és kommunikációs képességgel rendelkező robotokat felhasználhassuk, komplex feladatok megoldásában, mint az a következő fejezetekben majd látszik, meg kellett oldanunk a rendszerkorlátok ellenére, a kommunikációban résztvevő robotcsoport elemszámának növelését, a pontatlan szenzorok hibáiból adódó navigációs pontatlanság csökkentését.

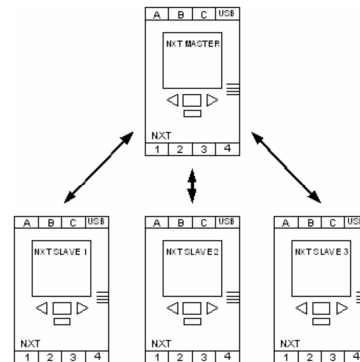
A már kidolgozott elméleti eljárások valós robotokra való alkalmazása számos olyan új problémát vetett fel, ami csak a szimulációs programok használatával nem került volna napvilágra.

A kommunikációs képességek növelése

A szimulációs kísérleteinkben használt NXT robotok (1. ábra) bluetoothon keresztül kommunikálnak egymással. Ez egy olcsó megoldás, és a viszonylag gyors az adatátviteli sebességéhez képest energiatakarékos. A robotok közti kommunikáció master-slave alapú és az eredeti Bluetooth NXT keretrendszer miatt a hálózatban csak egy master és három slave alkalmazható (2. ábra).



1. ábra. NXT robot



2. ábra. Master – slave alapú piconet kommunikációs hálózat

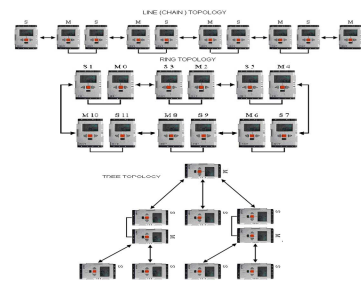
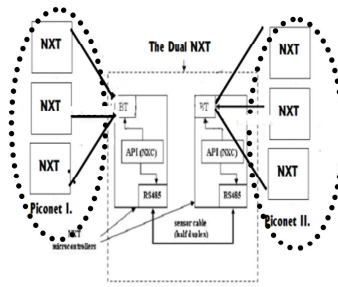
Ahhoz, hogy a szimulációkban résztvevő robotok számát tetszőlegesen megnövelhessük úgy, hogy azok kommunikációs szempontból egy hálózatot alkothassanak új eljárásokat kellett bevezetnünk.

Néhány kutató [10][11] a probléma megoldására egy új és egyszerű megoldást ajánlott, miszerint egy egyszerű autonóm egységet (host) kiegészítettek két bluetooth rádióval, így egy nagyméretű wireless szenzor hálózatot alakítottak ki (dupla-rádió alapú scatternet). Ebben a rendszerben a két összekötött bluetooth rádió két független piconet hálózat része úgy, hogy a host információkat közvetít az ő két rádiója között. Ez az NXT robotok esetén is megoldható mivel két robotnál lehetőség van összekötni I²C buszán RS485 –ös kábel segítségével. Az így kapott két független rádióval ellátott robotot Dual NXT-nek nevezzük (3. ábra). A téglákhoz továbbra is kapcsolhatunk bluetoothon keresztül robotokat, és így olyan megbízható ún. statikus scatternet hálózatot alakítottunk ki, melyben a robotokon belül az összeköttetés kábelen, míg a független robotok között bluetoothon valósul

meg. A módszer továbbfejlesztésével a csoporton belül a robotok száma tetszőlegesen növelhető és számtalan kommunikációs hálózat alakítható ki [12] (4. ábra).



3. ábra. A Dual NXT robot



4. ábra Robotokból kiépített kommunikációs hálózatok

Navigációs pontosság növelése

Azoknál a feladatoknál, ahol a robotokra kényszerítő feltételként hat egy kommunikációs limit távolság betartása, nagyon fontos a robotok maximális távolságának meghatározása, mozgás közben, a robotok precíz vezérlése és a robotok pontos navigációja. Ha két szomszédos robot a feladat megoldása közben a rossz navigáció miatt olyan távol kerül egymástól, hogy a bluetooth kapcsolat megszakad közöttük, akkor a kapcsolatot újraépíteni már nagyon bonyolult lenne, ezért olyan eljárásokat kellett készítenem melyeknek lényege a minél pontosabb navigáció elérése.

Azokban az esetekben, amikor a bejárando területről semmiféle információval sem rendelkezünk és a robotunkhoz sem kapcsolhatunk olyan eszközt, amely a külvilág ingereit érzékelné, nem rendelkezünk globális navigációs eszközökkel, akkor, ha a robot rendelkezik bizonyos képességekkel, lehetőség nyílik az odometriás módszer alkalmazására. A módszerhez a robotnak rendelkeznie kell lépésszámlálóval vagy kerek robot esetén elfordulás számlálóval. A módszer lényege, hogy a robot a kiinduló helyzetétől kezdve számolja a változásokat, a kerékelfordulásokat, ill. a tengely elfordulásokat s így határozza meg a saját helyzetét. Az esetünkben használt NXT robotok rendelkeznek ezekkel a képességekkel. A robot helymeghatározása odometriás eljárás segítségével a kerekek elfordulásának, és a tengelyfordulás számolásának figyelembevételével történik.

A tengelyhossz t , a Δs_b , Δs_j a bal és jobb kerék által megtett út, az x , y az aktuális koordináták, ϕ az irányszöveget jelöli akkor az alábbi (1) képlettel megbecsülhető a robot odometrikus pozíciója.

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_j + \Delta s_b}{2} * \cos(\phi + \frac{\Delta s_j + \Delta s_b}{2t}) \\ \frac{\Delta s_j + \Delta s_b}{2} * \sin(\phi + \frac{\Delta s_j + \Delta s_b}{2t}) \\ \frac{\Delta s_j + \Delta s_b}{t} \end{bmatrix} \quad (1)$$

A Dual NXT robotok vezérléséhez szükséges programokat NXC nyelven írtuk, a Brix Command Center 3.3.17. programozási környezetben, a robotokon a 1.03 firmware futott. A helymeghatározáshoz csak lokális navigációt használtunk, azaz a robot helyzetének meghatározásához csak az odometriás eljárás állt rendelkezésünkre, ami a talaj egyenetlensége, a kerekek eltérő mérete, a rossz súlyelosztás miatt sokszor pontatlan lehet. A pontosabb navigáció elérése érdekében a vezérlő programnyelv kerékszinkronizációs függvényét használtam fel. A robot intelligens kerekei rendelkeznek egy elfordulás számláló

szenzorral, mellyel tulajdonképpen a robot által megtett út is számolható. A pontosság érdekében kalibrációt is használtam. Több tucat mérést végezve sikerült 6m-es távolságokon $0.83 \cdot 10^{-3}$ navigációs pontosságot elérnem [13]. Nagyobb egyenes szakaszokon - 20m - $1.4 \cdot 10^{-2}$ navigációs pontosságot értünk el.

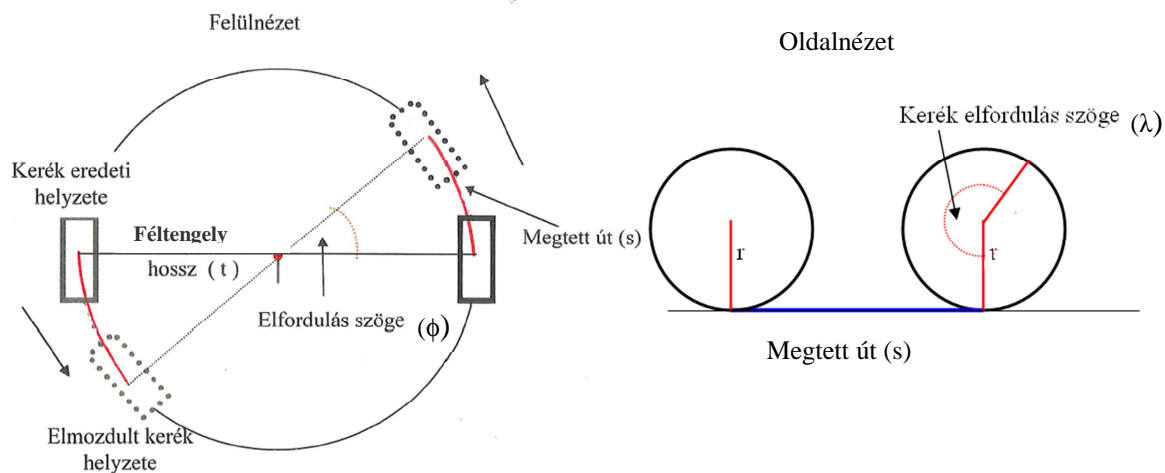
A robotokra épített iránytűk használatával még nagyobb pontosságot sikerült elérnünk ($3.0 \cdot 10^{-3}$), de a gyülekezési szimulációkban csak lokális navigációt használtunk.

Robotok forgásszögének meghatározása iránytű szenzor nélkül

A robotok a gyülekezési és területbejárési feladatok során nem csak egyenes pályán mozognak, hanem végre kell hajtaniuk különböző szögű elfordulásokat is. Természetesen a robotok navigációs pontosságát nagymértékben lerontja, ha a fordulásait pontatlanul teszik meg. Sokan használnak compass szenzorokat a forgásszögek pontos meghatározásához, de ezek is hibás értékeket adhatnak vissza, ha olyan helyen használják őket ahol sok az elektromos vezeték, nagy a környezet elektromágneses zaja.

Mi, a szimulációinkban egyébként is csak lokális helymeghatározó eszközöket akartunk használni, ezért a robot elfordulás szögeit egy egyszerű matematikai módszer segítségével állapítottuk meg.

A robotkerék sugarának, a tengelyhosszának, és a kerékelfordulás szögének ismeretében egyszerűen meghatározható a robot forgásszöge (a kerekeket összekötő tengelyének elfordulás szöge) (5. ábra).



5. ábra. A robot elfordulás szögének meghatározása

$$s = \lambda \cdot r \cdot \frac{\pi}{180}$$

$$s = \phi \cdot t \cdot \frac{\pi}{180}$$

$$\phi = \frac{\lambda \cdot r \cdot \pi}{180} \cdot \frac{180}{t \cdot \pi}$$

$$\phi = \lambda \cdot \frac{r}{t}$$

Ahol, $s \Rightarrow$ a kerék által megtett út hossza,

$r \Rightarrow$ a kerék sugara,

$t \Rightarrow$ a féltengely hossza,

$\lambda \Rightarrow$ a kerékelfordulás szöge,

$\phi \Rightarrow$ a tengelyelfordulás szöge.

(2)

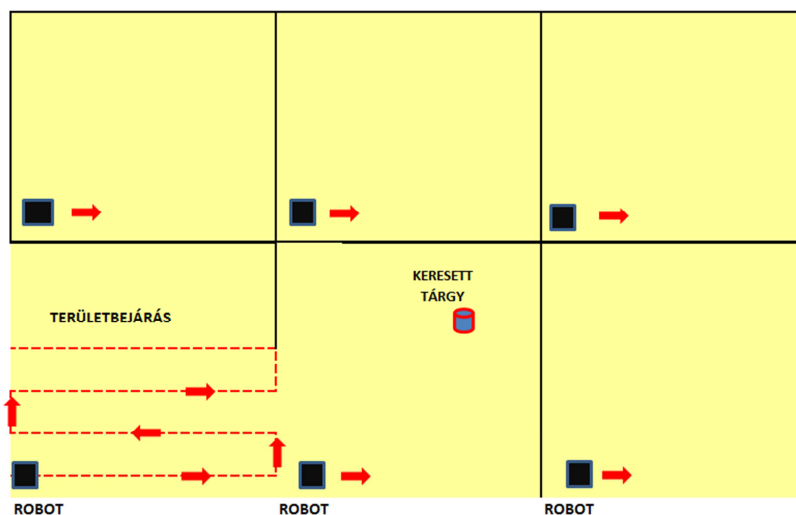
A (2) levezetéséből látszik, hogy az elfordulás szöge a tengelyhossz felének és a keréksugár arányának, valamint a kerekek elfordulás szögének ismeretében egyszerűen meghatározható. Ezek az adatok a robot rendelkezésére állnak. A konkrét feladatban ez azt jelentette, hogy 4° kerékelfordulás jelentett 1° tengelyelfordulást.

A navigációs pontosság növelése, a kommunikációs képességek növelése és robot

elfordulásaihoz szükséges eljárás elkészítése után már lehetőség nyílt egy komplex gyülekezési kísérlet megvalósítására.

Gyülekezés egy adott tárgy köré

A kísérletben $N = 6$ Dual NXT robot keresett meg egy tárgyat, és a tárgy felfedése után a robotok összegyűltek az objektum köré. A kísérlet kezdetekor kommunikációs szempontból a robotok gyűrű hálózatot alkottak úgy, hogy az R_i robot csak R_{i-1} és az R_{i+1} robottal tart kapcsolatot. Az R_1 robotnak az R_2 és R_6 robot volt a kommunikációs szomszédja. A szimulációs területet felosztottuk 6 részre, minden robot egy területrészt felderítéséért volt felelős. Ez a fajta területfelderítés megfigyelhető más kutatók hasonló munkájában is [14]. A robotok ütközésérzékelő szenzort és távolság érzékelő szenzort használtak a szimuláció során. A terület nagysága kb. $81m^2$ volt, egy robotra egy $3 \times 3m$ -es terület felderítése jutott (6. ábra). Amikor az R_i robot az ütközésérzékelőjével érzékelte a keresett tárgyat, leállította a keresést és egy broadcast üzenetben elküldte a tárgy derékszögű koordinátáit és az ő számát a többi robotnak. Az üzenet körbejárása után többi robot is befejezte a keresést, majd a saját helyzetükből és a tárgy koordinátáiból kiszámították a tárgy helyzetét. Ezután a kiszámított koordinátákat átszámították Polár koordinátákká és egyenként elindultak a tárgy felé. Mindig csak egy robot haladt a pályán a lehetséges ütközések elkerülése miatt.



6. ábra. Robotok tárgykeresési és gyülekezési feladata

A feladat tehát két fázisból állt, területbejárás (tárgykeresés) és gyülekezési fázisból. A gyülekezési fázisban, ha az R_i robot megtalálta a tárgyat az üzenetváltások után az R_{i-1} majd az $R_{i-2} \dots R_1, R_N, R_{N-1} \dots R_{i+1}$ robotok vonultak a tárgyhoz. Az előző részekben már írtunk a navigációs pontatlanságról, ezért a robotok a kiszámított koordináták alapján sokszor nem találták meg a keresett tárgyat csak a közelébe kerültek. Ekkor egy lokális keresést hajtottak végre, ami abból állt, hogy körbefordultak és a távolságérzékelőjük mérte a környezetükben lévő tárgyakat. Egy minimum kiválasztás után a legközelebbi tárgy irányába fordult a robot és az ütközésérzékelője jelzéséig megközelítette azt.

Mivel az R_i robot (ez találta meg a tárgyat) kivételével, a robotok nem tudnak különbséget tenni a tárgy és a többi robot között, a feladat eredményeképpen bár a tárgy köré gyülekeztek össze, de ez azt jelentette, hogy néha már a tárgyat megtaláló robotot nézték a tárgynak és ezt közelítették meg.

A feladat továbbfejlesztése lehet, hogy az NXT 2.0-ás robotot használva a tárgy és a

robot megkülönböztetésére felhasználjuk annak színérzékelő szenzorját is.

A feladat algoritmusa miután a kommunikációs hálózat felállt:

Algorithm R_i robot

```
while (not object_found() ) do
    scan_field_using_odometry(onestep)
    if (object_found()) then
        coordinates=calculate_odometry()
        send_broadcast(coordinates, end_scan_field)
    endif
    if (received(coordinates, end_scan_field) ) then
        coordinates=received_coordinates()
        (angle,distance)=calculate_polar(coordinates)
        turn(angle)
        go_forward(distance)
        local_search_object()
        object_found()=true
    endif
end
endalgorithm.
```

3. NXT robotok gyülekezési szimulációja lokális navigációt és indirekt kommunikációt használva

A szimulációban egy tetszőleges területen szétszóródott NXT robotrajnak kellett összegyűlnie úgy, hogy a robotok nem használtak globális helymeghatározó eszközöket csak az előző fejezetben leírt odometriás eljárást és a tengelyelfordulást számoló eljárást. A feladat egyes fázisait szinkron módon hajtották végre, azaz minden robot egyszerre végezte el a részfeladatokat. A rajban minden robot szerepe és feladata ugyanaz, közelebb kell mennie a társrobotokhoz. A feladat végrehajtása során a robotok semmilyen kommunikációs csatornát nem használtak (bloetooth, hang alapú kommunikáció, fényjelek, stb.) csak egymás helyzetéből és mozgásából nyertek ki információt. Szokás ezt indirekt kommunikációnak is nevezni. Jól megfigyelhető ez az állatvilág csoportosan élő állatai között, mint pl. a halak között is.

A robotok csak korlátozott részt látnak az őket körülvevő térből, távolságérzékelőjük segítségével mérik meg a többi robottól való távolságukat. Sokszor a robotok, mivel kiterjedéssel rendelkeznek, takarják egymást társaik elöl ezzel is csökkentve a láthatóságot.

Számos kutató készített már ebben a témában olyan eljárásokat ahol a robotok korlátozott látással, szinkronmódon, lokális navigációt használtak. Legtöbben csak elméleti munkát készítettek valós robotokkal nem dolgoztak. Alapvető eljárásnak minősül a témakörben Ando [15] munkája, aki pontszerű, szinkronban működő, robotokra készített gyülekezési algoritmust. Néhányan, mint pl. Floccini és társai [16] limitált látótávolságú pontszerű robotokkal egy aszinkron működő algoritmust valósított meg, globális navigációval. Bolla és társai egy munkájában [17] már figyelembe vette azt is, hogy a valós robotoknak kiterjedése is van, nem átlátszóak, összeütközhetnek, blokkolhatják egymást. Munkájukban erre a problémára adtak megoldást, de hipotéziseik bizonyítására nem valós robotokat hanem MATLAB szimulációt használtak.

A szimulációban résztvevő NXT robotok tulajdonságai

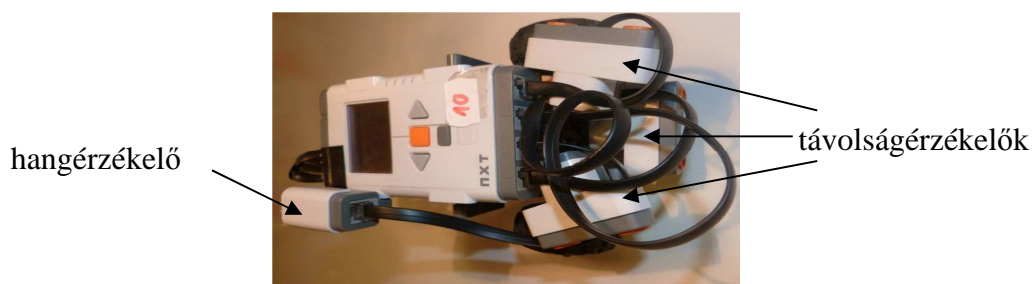
Munkánkban mi valós NXT robotokat használtunk. Jelöljük a robotokat a későbbiekben $R_1 \dots R_i \dots R_N$ -nel. Ahhoz, hogy a gyülekezési feladat végrehajtható legyen, a robotrajnak rendelkeznie kell bizonyos alaptulajdonságokkal, képességekkel. Ezek közül a legfontosabb tulajdonságokkal rendelkeznek az NXT robotok, ill. a 2. fejezetben leírt eljárások segítségével kifejlesztettük őket. Ilyen képességek a következők:

Autonóm működés képessége	-	NXT robot működik autonóm módon.
Szinkron működés képesség	-	hangérzékelő, hanggenerátor, belső idő számláló.
Kiterjedés miatt ütközésselkerülés - képessége	-	3 távolságérzékelő használatával.
Memória, processzor az egyes számítási műveletek végzéséhez	-	32 MB RAM, 256 Kb FLASH memória. Atmel® 32-bit ARM® AT91SAM7S256 processzor.
Limitált látótávolság képessége	-	távolságérzékelő szenzor 2-2,5m.
Lokális helymeghatározás	-	a 2. fejezetben leírt eljárás alapján.
Kommunikáció képessége	-	a feladat során nem fontos, csak indirekt kommunikáció van.

A szimuláció fázisai

A szimuláció egy kb. 1khz-es hangra indul, ez adja az első szinkronjelet a robotoknak a működéshez. A robotok rendelkeznek 1 hangérzékelő szenzorral, ami érzékeli ezt. A robot rendelkezik ezen felül még 3 távolságérzékelővel is, a középső főként a társrobotok helyzetének meghatározására a másik kettő pedig a haladás közbeni ütközések elkerülésére szolgál. (7. ábra)

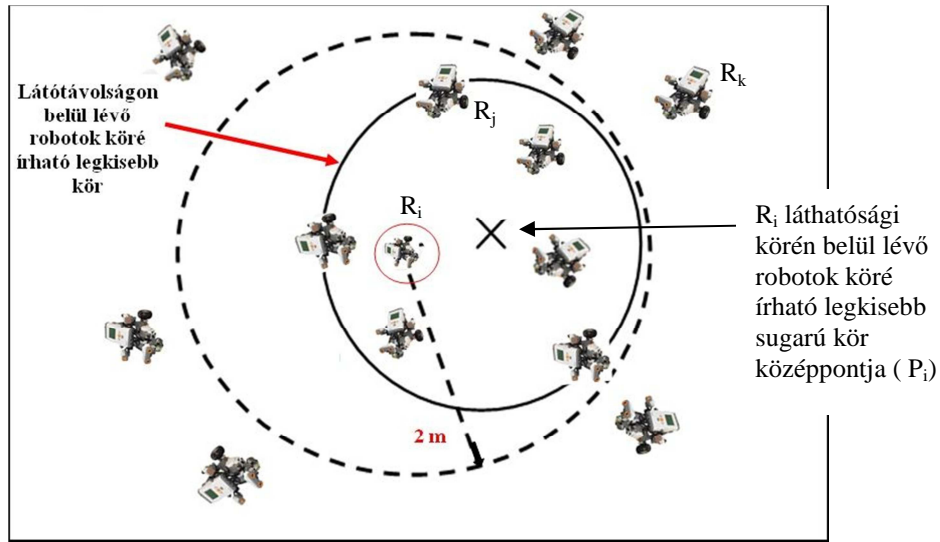
A továbbiak során a robotok a gyülekezési feladatot három fázisra osztják. Az első fázisban a robotok körbe forognak és felderítik azon társaikat, akik láthatósági távolságon belül vannak és nincsenek takarva a robot elöl más társuk által. Nevezzük ezt a fázist *körbenézésnek*.



7. ábra. A gyülekezésben résztvevő robotok

A körbenézés során minden robot a tengelyelfordulás meghatározására alkalmas eljárás segítségével 360° fordul és megjegyzi azoknak a robotoknak a helyzetét, amelyek látótávon belül vannak. Ez a távolság 2m. A robot távolságérzékelőit tesztelő mérések során azt állapítottuk meg, hogy eddig a távolságig biztonságosan mér a robot távolságérzékelője és a pontatlansága 1% alatt van. Mi UltraSonic szenzort használtunk, de gyártanak a robotokhoz IRSeeker szenzorokat melyek nagyobb távolságra is pontosan mérnek. Természetesen vannak

olyan robotok - jelölje ezt R_k - amelyek az R_i látóterületén kívül esnek. Azok a robotok melyek minden robottól 2m-nél nagyobb távolságra helyezkednek el, nem vesznek részt a szimulációban.



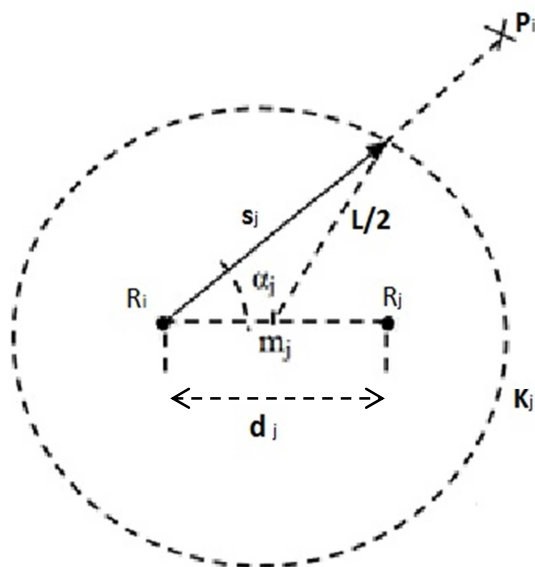
8. ábra. A gyülekezés második fázisa, a robotok új helyzetének kiszámítása

A gyülekezés második fázisában a robotok a felfedezett robottársak helyzete és a saját helyzetük alapján meghatározzák a látóterületén belül lévő robotok köré írható legkisebb sugarú kör középpontját jelölje P_i (8. ábra). Nevezzük ezt a fázist *Új pozíciók számítása* fázisnak. Minden robot elvégzi ezt a számítást és mivel az egyes robotok más-más robot társakat láthatnak, az új irány robotonként változó lehet. Ebben az irányba fordulnak a robotok és megindulnak a kiszámított középpont felé. Ezek a lépések garantálják a gráf zsugorodását minden lépésben. Az indulás előtt azonban egy újabb számítást végeznek. Ez azért kell, hogy ne szakadjon meg az eddig már kialakult ún. látóterületi gráf, azaz ha két robot látja egymást, akkor az új pozícióba jutás végén is lássák egymást. Ezt az adott robot minden egyes látható robottársánál külön megvizsgálja. Vegyük az R_i és R_j robotot, amelyek látják egymást a t időpontban. Ha azt akarjuk, hogy ez a két robot a $t+1$ időpontban is lássa egymást, akkor R_i a lépése során nem hagyhatja el a 9. ábrán K_j -vel jelölt szaggatott kört. Ennek a körnek a középpontja a két robot közötti felezőpont (m_j), sugara pedig $L/2$ (1m), azaz a látóterületi határ fele. Az R_i és R_j robotok távolságát jelölje d_j . Ez a feltétel igaz az R_j lépésénél is. Ilyen feltételek mellett a látóterületi gráf élei biztosan nem szakadnak meg, ez elemi geometriai eszközökkel bebizonyítható (lásd 9. ábra). Az s_j távolságot az alábbi képlettel számolhatjuk:

$$s_j = (d_j/2)\cos \alpha_j + \sqrt{(L/2)^2 - ((d_j/2)^2(\sin \alpha_j)^2)} \quad (3)$$

Ezt az R_i minden egyes j . társára ki kell számítani és kiválasztani a legkisebbet. Ez lesz a legszigorúbb korlát a lépés során. Tehát a lépés hossza

$$R_i \text{ lépéshossza} = \min \{ R_i \text{ és } P_i \text{ távolsága, } \min(s_j) \} \quad (4)$$



9. ábra. R_j lépéshosszának meghatározása

A szimuláció harmadik fázisa az új pozícióba felé haladás. Nevezzük ezt *mozgái* fázisnak. A szimuláció ezen szakaszában minden robot a kiszámított irányba fordul és az előbb leírt lépéshosszt teszi meg. Az első szinkronjel mint írtuk egy hangjelzés volt a többi szinkronjel a robotok egy belső időszámláló függvénye adja az egyes feladatok végrehajtásához. A harmadik fázisban, mivel a robotok kiterjesztéssel is rendelkeznek (22x18x14cm) meg kell oldani az összeütközések elkerülését is. A robot 3 távolságérzékelőjével megközelítőleg 160° – os szögben lát maga elé. Abban az esetben, ha egy társ robot keresztezi az útját és 30 cm távolságnál közelebb kerül, a robot megáll és ebben a fázisban már nem megy tovább. Az így kialakulható holtponthelyzetek feloldására ezek a robotok még az új körbenzési fázis előtt, a közel került társrobotok irányával ellentétesen 30cm hátrátolhatnak.

A szimuláció fázisai ezután ciklikusan ismétlődnek a feladat befejezéséig. A feladatot egy kezdőfeltételben előre megadottak szerint tekinthetjük befejezettnek. Az összegyűlés vagy egy meghatározott időkorlátig, vagy egy meghatározott lépésszámig tart, vagy akkor ér véget, ha egy megfelelő kontakt gráf alakul ki.

4. Következtetések

A két szimulációban megmutattuk, hogy milyen képességeit kell az NXT robotoknak fejleszteni ahhoz, hogy részt vehessenek komplex gyülekezési szimulációban. Az első gyülekezési eljárás során egy tárgy körüli összegyűlést tovább lehet fejleszteni színérzékelő szenzor segítségével, így megkülönböztethető a robottárs és keresett tárgy.

A második szimulációban résztvevő robotok képességeit úgy alakítottuk, hogy a robotok legalább elvileg képesek legyenek egy megfelelő kontakt gráfot kialakítani. A szimulációt olyan speciális helyzetekre végeztük el egyelőre, ahol három és hat NXT robot helyezkedett el egy szabályos háromszögszög és hatszög csúcsain úgy, hogy minden robot látott mindenkit. A robotok összebb gyűltek, de a kiszámított (várt) pozícióktól eltérő helyzetekbe kerültek. Ez a szenzorok hibáiból a számítási műveletek kerekítési pontatlanságaiból is adódik.

A jövőben a pontosabb helymeghatározás érdekében szeretnék a robotokon egy infravörös keresőt az IRSeeker távolságérzékelőt használni ez pontosabb, nagyobb

távolságokra is használható, a látószöge 240° ami képessé teszi az ütközésfigyelésre is. Az új eszköz segítségével ki szeretném terjeszteni a kísérletet több robotra és általános esetekre is.

5. Köszönetnyilvánítás

A kutatás a TÁMOP 4.2.4.A/2-11-1-2012-0001 azonosító számú Nemzeti Kiválóság Program – Hazai hallgatói, illetve kutatói személyi támogatást biztosító rendszer kidolgozása és működtetése országos program című kiemelt projekt keretében zajlott. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Ezúton szeretnénk köszönetet mondani Mészáros Ádám és Csernovszki Zoltán volt KF GAMF-os hallgatóknak, valamint Éva Róbertnek, akik a szimulációk készítésében és a mérésekben segítettek.

Irodalomjegyzék

- [1] G.Beni, J. Wang, Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, 1989.
- [2] Clough, B.: UAV Swarming? So What are Those Swarms, What are the Implications, and How Do We Handle Them? Proceedings of the AUVSI Unmanned Systems Symposium, , Orlando, FL., July 2002.
- [3] Lin, K.C.: Controlling a Swarm of UCAVs ~ A Genetic Algorithm Approach, Final Report for VFRP, Information Directorate AFRL, 2001.
- [4] <http://www.agent.ai/main.php?folderID=3&articleID=854&ctag=articlelistapp=1&iid=1>
- [5] F. Mondada, G. C. Pettinaro, A. Guignard, I.W. Kwee, D Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, M. Dorigo, Swarm-Bot: A New Distributed Robotic Concept, Autonomous Robots 17, pp.193–221, 2004
- [6] M. Dorigo, Swarm-Bots and Swarmanoid: Two Experiments in Embodied Swarm Intelligence, Web Intelligence and Intelligent Agent Technologies, WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on, 2009
- [7] <http://index.hu/tech/tudomany/repulo3066/>
- [8] <http://web.mit.edu/press/2010/seaswarm.html>
- [9] <http://www.kurzweilai.net/university-of-pennsylvania-swarm-of-nano-quadrotors>
- [10] K. Sohrabi, W. Merrill, J. Elson, L. Girod, F. Newberg, W. Kaiser, Scalable self-assembly for ad hoc wireless sensor networks, IEEE Transactions on Mobile Computing , pp.317-331, 2002.
- [11] M. Leopold, M. B. Dydensborg, P. Bonnet, Bluetooth and sensor networks: A reality check, in Proc. 1st ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 103–113, 2003.
- [12] A. Pásztor, T. Kovács, and Z. Istenes, Swarm Intelligence Simulation with NXT Robots Using Piconet and Scatternet, Applied Computational Intelligence and Informatics pp.199-204 0.1109/SACI.2009.5136241, 2009.
- [13] A. Pásztor, T. Kovács, and Z. Istenes, Compass and Odometry Based Navigation of a Mobile Robot Swarm Equipped by Bluetooth Communication, ICCO CONTI 2010, IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics, Timisoara, Romania, page 565-570, 2010.
- [14] I. Rekleitis, A. P. New, and H. Choset, “Distributed coverage of unknown/unstructured environments by mobile sensor networks”, Multi-Robot Systems. From Swarms to Intelligent Automata, vol. III, L. E. Parker et al., Eds.,

- Netherland: Springer, pp. 145–155, 2005.
- [15] H. Ando, I. Suzuki, M. Yamashita „Formation and Agreement Problems for Synchronous Mobile Robots with Limited Visibility” Syp. of Intelligent Control, pp.(453-460) 1995.
 - [16] P. Flocchini, G. Prencipeb, N. Santoroc,P. Widmayer, Gathering of asynchronous robots with limited visibility”, Theoretical Computer Science, pp.147 – 168. 2005.
 - [17] K. Bolla, T. Kovacs, G. Fazekas, Gathering of fat robots with limited visibility and without global navigation. In Swarm and Evolutionary Computation, pp. 30-38, Springer Berlin Heidelberg, 2012.

Szerzők

Pásztor Attila: Informatika Szakcsoport, GAMF Kar, Kecskeméti Főiskola. 6000 Kecskemét, Izsáki út 10, Magyarország.

E-mail: pasztor.attila@gamf.kefo.hu

Czuprák Zsolt: Informatika Szakcsoport, GAMF Kar, Kecskeméti Főiskola. 6000 Kecskemét, Izsáki út 10, Magyarország

E-mail: czuprak.zsolt@gmail.com