

TDK dolgozat

# Csoportosan repülő robotok – Alakzatrepülés

Szörényi Tamás

ELTE TTK Fizika BSc, biofizikus szakirány



Témavezetőm:

Dr. Vásárhelyi Gábor

tudományos munkatárs, ELTE TTK Biológiai Fizika Tanszék

Konzulens:

Virágh Csaba

doktorandusz, ELTE TTK Biológiai Fizika Tanszék

Budapest

2013

# Kivonat

Az ELTE TTK Biológiai Fizika Tanszékén csoportosan repülő robotokat fejlesztünk, felhasználva az állatvilágban megfigyelhető látványos csoportos mozgás matematikai modelljeit. Ebben a munkában az én feladatom az alakzatban repülés irodalmának megismerése, valamint a legalapvetőbb alakzatok önszerveződő csoportos modelljeinek elkészítése és implementálása volt, szimulációs és valós környezetben egyaránt. A rendszerben nincs központi vezérlés, az egyedek lokális információk alapján hoznak létre globális mintázatokat. Ezen ágens alapú algoritmusok megírásakor figyelembe kell vennünk a kopterek egyéni tulajdonságait, például a kommunikáció hatótávolságát és késleltetését, vagy a helymeghatározás pontatlanságát. A modellalkotáshoz fel kell használnunk a statisztikus fizika eszköztárát és ki kell aknáznunk a C programozás lehetőségeit. A modellek teszteléséhez szimulációs környezetet fejlesztettünk ki, ami grafikus felülettel is rendelkezik, így az egyedek mozgása nyomon követhető, és bizonyos paraméterek futási időben is állíthatók.

A dolgozatban két konkrét alakzatrepülési feladattal foglalkozom: szabályos sokszöggel és egyenessel. A dolgozat bemutatja a szimulációs környezetet és a modellalkotás lépéseit, betekintést nyújt az alakzatrepülés általunk készített modelljeinek megtervezésébe és működésébe. A szimuláció alapján optimalizált modelleket valós kvadrokopter flottával is sikerrel teszteltük. A dolgozatban mindezek mellett arra is szeretnék rávilágítani, hogy mennyire fontos nyitott szemmel járni, és megfigyelések alapján biológia inspirálta technológiákat fejleszteni.

# Köszönetnyilvánítás

Az egész kutatásunk egy nagy csapatmunka. Minden feladatot közösen megvitatunk, és mindenki véleményét figyelembe vesszük. Ezért szeretném hangsúlyozni, hogy bár a dolgozat a saját önálló munkám eredménye, rengeteg segítséget kaptam a kutatómunka és a dolgozat összeállítása során is a csapat tagjaitól. Ebből kifolyólag gyakran alkalmazom a többes szám első személyt a dolgozat szövegében.

Ezúton szeretnék köszönetet mondani témavezetőmnek, Dr. Vásárhelyi Gábornak és konzulensemnek, Virágh Csabának, valamint további munkatársaimnak: Tarcai Norbertnek és Somorjai Gergőnek. Végül, de nem utolsósorban köszönettel tartozom a kutatócsoport vezetőjének, Prof. Vicsek Tamásnak.

# Tartalomjegyzék

|                                                          |    |
|----------------------------------------------------------|----|
| Kivonat.....                                             | 2  |
| Köszönetnyilvánítás.....                                 | 3  |
| 1. Bevezetés.....                                        | 6  |
| 2. Szimulációs környezet.....                            | 11 |
| A koptereink felépítése.....                             | 11 |
| A program összetevői és működése.....                    | 12 |
| 3. Az alakzatba rendeződés modelljeinek megalkotása..... | 14 |
| Első lépések: a sablonfájlok.....                        | 15 |
| Általános kölcsönhatási tagok a modellekben.....         | 17 |
| Szabályos sokszögbe rendeződés modellezése.....          | 19 |
| Egyenesbe rendeződés modellezése.....                    | 28 |
| 4. Szimuláció kontra valóság: a kísérletek.....          | 31 |
| Út a stabil repülésig.....                               | 31 |
| A gyűrű modell kísérleti tesztje.....                    | 33 |
| 5. Kitekintés.....                                       | 37 |
| Irodalomjegyzék.....                                     | 38 |

# Ábrajegyzék

|                                                                                                                                                                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. ábra. Közel szabályos kör formába rendeződő tengeri halraj<br>( Forrás: <a href="http://www.mrns.hu/userfiles/image/halraj.jpg">http://www.mrns.hu/userfiles/image/halraj.jpg</a> ).....                                                                                    | 8  |
| 2. ábra. A barázdálódás második szakasza: a szedercsírából (morula) hólyagcsíra (blastula) alakul ki<br>( Forrás: <a href="http://upload.wikimedia.org/wikipedia/commons/c/c6/Blastulation.png">http://upload.wikimedia.org/wikipedia/commons/c/c6/Blastulation.png</a> )..... | 9  |
| 3. ábra. V-alakban repülő vadlúd (nyári lúd, Anser anser) csapat<br>( Forrás: <a href="http://labradorok.files.wordpress.com/2011/11/nc3a9vtelen1.jpg">http://labradorok.files.wordpress.com/2011/11/nc3a9vtelen1.jpg</a> ).....                                               | 10 |
| 4. ábra. A szimulációs program grafikus felülete.....                                                                                                                                                                                                                          | 14 |
| 5. ábra. A PID algoritmus együtthatóinak behangolása ( Forrás: [7] ).....                                                                                                                                                                                                      | 17 |
| 6. ábra. Szabályos hatszög körülírt köre ( Forrás: <a href="http://cms.sulinet.hu/get/d/57ea6020-8cf8-42bc-976d-7379cef32a0d/1/7/b/Normal/a20aa003.jpg">http://cms.sulinet.hu/get/d/57ea6020-8cf8-42bc-976d-7379cef32a0d/1/7/b/Normal/a20aa003.jpg</a> ).....                  | 20 |
| 7. ábra. A sokszög alakzatba álláshoz alkalmazott függvényünk grafikonja.....                                                                                                                                                                                                  | 22 |
| 8. ábra. A gyűrű alakzat szimulációja (1).....                                                                                                                                                                                                                                 | 25 |
| 9. ábra. A gyűrű alakzat szimulációja (2).....                                                                                                                                                                                                                                 | 25 |
| 10. ábra. A gyűrű alakzat szimulációja (3).....                                                                                                                                                                                                                                | 25 |
| 11. ábra. A gyűrű alakzat szimulációja (4).....                                                                                                                                                                                                                                | 25 |
| 12. ábra. A $P$ és $Q$ rendparaméterek időbeli változása automatikus $R = 17.7\text{ m}$ sugarú kör esetén,<br>súrlódás nélkül és súrlódással.....                                                                                                                             | 26 |
| 13. ábra. A $P$ és $Q$ rendparaméterek időbeli változása az automatikusan számoltnál kisebb $R = 14.7\text{ m}$ sugarú kör esetén, súrlódás nélkül és súrlódással.....                                                                                                         | 26 |
| 14. ábra. Az egyesített rendparaméter teljes futtatásra vett időátlaga a $C_{slip}$ súrlódási együttható<br>függvényében különböző $R$ és $D$ esetén, zaj nélkül.....                                                                                                          | 27 |
| 15. ábra. Az egyesített rendparaméter teljes futtatásra vett időátlaga a $C_{slip}$ súrlódási együttható<br>függvényében különböző $R$ és $D$ esetén, nagy zajjal.....                                                                                                         | 27 |
| 16. ábra. Virtuális térbeli szigmoid gödör.....                                                                                                                                                                                                                                | 29 |
| 17. ábra. Az egyenes modell szimulációja ideális beállításokkal.....                                                                                                                                                                                                           | 30 |
| 18. ábra. Ad-hoc kommunikációs lánc szemléltetése.....                                                                                                                                                                                                                         | 31 |
| 19. ábra. A gyűrű modell kísérleti tesztje 9 kopterrel (fotó).....                                                                                                                                                                                                             | 34 |
| 20. ábra. A gyűrű modell tesztelése 9 kopterrel.....                                                                                                                                                                                                                           | 35 |
| 21. ábra. Alakzatban repülés 60 másodperces részlete.....                                                                                                                                                                                                                      | 36 |
| 22. ábra. Az $\Omega$ , $P$ és $Q$ rendparaméterek az idő függvényében.....                                                                                                                                                                                                    | 36 |

# 1. fejezet

## Bevezetés

A természetben a legkülönbözőbb méretskálákon figyelhetjük meg a csoportos mozgás jelenségét. A mikroszópikus méretű egysejtűektől a hatalmas testű emlősökig, szinte minden szerveződési szinten létezik kollektív viselkedésforma. Nem véletlenül alakulnak ki ezek a csoportok. Számos előnye van, ha valaki többedmagával kezd hozzá egy adott feladathoz. Egy csoport nagyobb védelmet biztosít a támadókkal szemben, ugyanakkor a csoport egyedei hatékonyabban tudnak élelmet, vizet, vagy éppenséggel telephelyet, otthont keresni.

A "nagyobb létszám = nagyobb biztonság" elv az egyik alapvető oka az állatcsoportok létrejöttének. A legtöbb állatfaj elsődlegesen ezért keresi a társaságot, mivel így jelentősen növekszik a túlélési esélye. A kiváló memóriával rendelkező afrikai elefántok szárazság idején csordákban kelnek útra, hogy vízgyűjtő területeket keressenek az idősek útmutatása alapján. A szurikáta vagy rókamangusza sokféleképpen használja ki a csoport előnyeit. Együtt építik az otthonukat, közösen szereznek élelmet, és nagyon hatékony riadóláncot alakítanak ki, amikor a kicsikre vigyáznak. Az ízeltlábúak közül kiemelkednek a hangyák, valamint a darazsak és méhek. A hangyák és termeszek hihetetlenül összetett, bonyolult struktúrájú várat képesek együttes erővel felépíteni. Élelemszerzéskor felderítőik szétszóródnak, és ha egyikük táplálékra talál, visszamegy a bolyba, miközben illatcsíkot húz maga után. A bolyban értesülnek a sikerről, és a szagnyomot követve, láncban vonulnak a táplálékhoz, majd azt darabonként ugyanazon az útvonalon szállítják haza. A darazsak és méhek szintén közösen építik a fészket ill. kast és szerzik a táplálékot. Nem beszélve arról, hogy mennyire elrettentő tud lenni, ha a több száz darázs együtt zúg és védi a fészket.

Egyes ragadozó állatok is kihasználják a csoport előnyeit. A vadon élő kutyafélék többsége köztudottan rendkívül fejlett szociális csoportokban, falkákban él. A macskaféléknél ez nem terjedt el, közülük egyedül az oroszlánok állnak össze falkává. Egy falka vadászata sokkal nagyobb

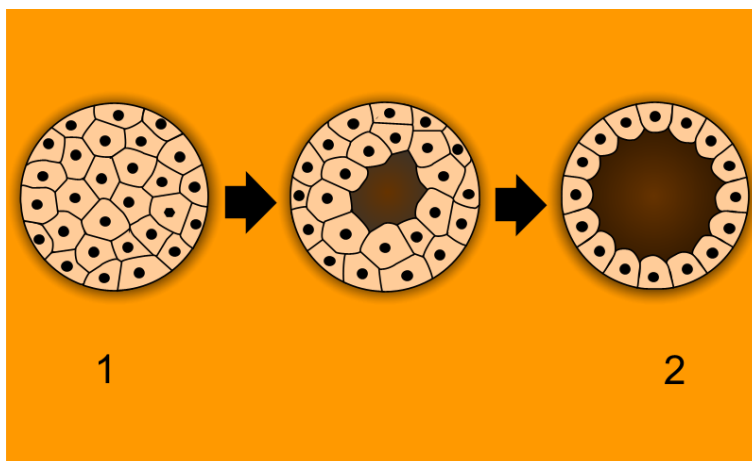
arányban hatékony, mint egy magányos ragadozóé. Az együttműködésnek köszönhetően könnyedén terelik a prédaállatokat, leválasztják a gyenge, sérült vagy öreg egyedeket, és többnyire bekerítéses módszerrel kapják el. A falka méretétől függően egy jól összeszokott csoport jóval nagyobb zsákmányt is le tud teríteni, mint amire egy önálló egyed képes lenne. Nem egyszer dokumentálták, hogy oroszlánok támadtak meg zsiráfokat, kafferbivalyokat, vagy farkasok fogtak el bölényeket. Arra is létezik példa, hogy két különböző faj csoportjai együtt tudnak működni. A trópusokon a delfinek precízen együttműködve légbuborékokkal terelik a halakat egy sűrű rajba [1]. Ezután egyikük hangjelzéssel értesíti a sirályokat, hogy felülről lehet lakmározni. Az eredmény az, hogy míg a sirályok a levegőből lecsapva a halakat felülről fogyasztják, azok igyekeznek lefele menekülni, mikor is beleszaladnak a delfinek csapdájába. Ezzel a módszerrel a sirályok és a delfinek is jól laknak. A fenti példák alapján belátható, hogy a csoport, a munkamegosztás és az együttműködés igencsak meghatározó tud lenni a természetben.

Előfordul, hogy egy rajban nem egyszerűen összetömörülnek, hanem közel szabályos alakzatba rendeződnek az egyedek. Miért teszik ezt? A megnövekedett biztonság, mondhatni természetes velejárója a csoportba verődésnek. Vagy mégsem? Nem mindegy, hogy a tömeg mennyire szervezett alakzatban áll. A rajban úszó tengeri halaknak a körülményekhez igazodva kell a rendezettségüket módosítani. Léteznek például vegyes, többszörös faji összetételű halrajok, melyek csupán összeverődnek egy tömegbe, és az egyedek csak lazán tartják a kapcsolatot egymással. A szervezettség gyakran változhat attól függően, hogy épp milyen tevékenységet végeznek az egyedek [2]. A skóciai St. Andrews Egyetem egy munkatársa által folytatott tanulmány szerint a csoportban élő tengeri állatok hasonló törvényszerűségek szerint alakítanak ki formációkat. A halak, vagy például a krillek is, úgy állnak alakzatba, hogy közben létszámtól függetlenül állandó felület/térfogat arányt tartanak meg. A rajforma kialakulását leginkább befolyásoló tényezőként az oxigénben gazdagabb vízhez való hozzáférést, valamint a ragadozók jelentette kockázatot állapították meg (1. ábra) [3] [4].



1. ábra. Közel szabályos kör formába rendeződő tengeri halraj

A szabályos alakzatok nem csak védelmi szerepet tölthetnek be, és egészen másfajta rendszerekben is előfordulhatnak. A barázdálódás során a megtermékenyített petesejtből (zigóta) létrejön a szedercsíra állapot (morula), majd ez követően a hólyagcsíra (blastula) (2. ábra). A szedercsíra állapot egyszerű sejtcsoportosulása a hólyagcsírában egy gömbfelszíni elrendezésbe megy át. A gömb alaknak azért is van jelentősége, mert így jobban el vannak választva a sejtek, el tud kezdődni a differenciálódás.



2. ábra. A barázdálódás második szakasza: a szedercsírából (morula) hólyagcsíra (blastula) alakul ki

A zárt alakzatok mellett nyitott szabályos formák is létrejöhetnek az állatvilágban. Köztudott, hogy a vadludak V-alakban repülnek, s közben óriási távolságokat képesek megtenni pihenés nélkül (3. ábra). De miért éppen a V-alakot választották? Nos, egy internetes folyóirat ezzel kapcsolatos cikkében olvasható szellemes megjegyzés szerint "nem csak azért, mert a Q-alak túl nehéz lenne" [5]. A cikk felfedi, hogy két okot is találtak arra, hogy miért előnyös a ludaknak V-alakban repülni. Az egyik a könnyebb repülés. Az alakzatban repülő ludak – nyilván a legelöl haladó kivételével – az előttük haladó lúd szárnycsapásai által keltett ún. *szárnyvégi örvényeket* (wingtip vortices) használják ki, hogy kisebb energiabefektetéssel tudjanak repülni. Amikor a vezető lúd elfárad, lassan hátravonul, és egy másik lép a helyébe. Ezzel a módszerrel a repülési táv akár 71%-kal növekedhet az egyedül repülő madarakéhoz képest [6]. Akkor viszont miért nem repül az összes madár V-alakban? Az igazság az, hogy a technika megkövetel egy bizonyos méretet. A kisebb testű maradaknál a hatás nem olyan erőteljes, hogy érdemes lenne kihasználni, illetve a több ezer egyedből álló madárrajokban már nehéz is lenne rendet tartani, manőverezni. Ezenkívül az alakzatot csak a hosszú távokat repülő vándormadaraknak éri meg kialakítani, rövid távon csak bonyodalmakhoz vezet a folyamatos alakzatváltás. A másik érv a vadludak számára a V-alak mellett a vizuális kontaktus. A Rhode Island-i Egyetem kutatói úgy gondolják, hogy ha a V-formáció szöge  $29^\circ$  vagy nagyobb, akkor minden egyes lúd szemmel tudja tartani a többieket [5].

Ez igen fontos szempont, hiszen így mindenki közel egyszerre tud reagálni és alkalmazkodni a megváltozó körülményekhez, például egy társuk megsérülésekor vagy a vezetőjük fáradásakor. Világos, hogy ez csak akkor valósulhat meg, ha az egyes egyedek lokális érzékelési tartományába esnek a társaik, s ezért lehet alkalmas a ludak számára a V-alak.



3. ábra. V-alakban repülő vadlúd (nyári lúd, *Anser anser*) csapat

Leginkább a természet szeretete, és az állatvilág ezen különleges tulajdonságai vezettek engem oda, hogy elkezdtem érdeklődni a csoportos mozgás iránt. Az ELTE TTK Biológiai Fizika Tanszékén találtam meg Dr. Vicsek Tamás fizikus professzort, aki elmondta, hogy kutatásaikban az állatok csoportos mozgását igyekeznek leírni, és a megfigyelések alapján modellezni. Ez felkeltette az érdeklődésemet, mire ő Dr. Vásárhelyi Gáborhoz irányított, aki végül a témavezetőm lett.

A kutatás lényegi célja az, hogy csoportban repülő robotok komplex feladatok önálló megoldására legyenek képesek. A kutatócsoport 10 darab kisméretű kvadrokopterrel dolgozik. A kvadrokopterek négyrotoros szárny nélküli gépek, melyek a propellerek forgási sebességének változtatásával irányíthatók. A kutatás egyik fő irányvonala a csoportos mozgást leíró modellek alkalmazása. A megvalósításhoz több részfeladat egymás melletti elvégzése szükségeltetik, ezért a kutatás több ágon is fut. Az egyik része a mérnöki tervezés, a kopterek vázának és elektronikájának fejlesztése – ez főként Somorjai Gergő munkaköre. A kopterek szoftveres vezérlő algoritmusát főleg Vásárhelyi Gábor írja, Nepusz Tamás segítségével. A reptetési naplók elemzését, a hibák keresését döntően Tarcai Norbert végzi. Fontos, hogy legyen egy virtuális szimulációs környezet, amiben a lehető legvalóságosabban le tudjuk tesztelni, hogy nagyjából milyen viselkedés várható a kopterektől a valóságban. Ennek fejlesztéséért, valamint a repülések stabilitásának vizsgálatáért Virágh Csaba felelős. Én akkor kapcsolódtam be a projektbe, amikor még csak pár kopter volt összerakva, sok probléma volt még a stabil repüléssel, és a szimulációs környezet első verziója épphogy elkészült. A nekem való feladatkör keresésekor felmerült, hogy idővel feladatspecifikus

modellek készítése is szükségessé válik. Ezeket a vezérlő algoritmusba implementálva a koptereknek egy adott feladatot kell elvégezniük. Úgy döntöttem elvállalom az ezzel kapcsolatos feladatokat, s így ez lett munkám alapja. Első megbízásom az volt, hogy az egyedek önszerveződő módon szabályos alakzatokba, kezdetben sokszögbe és egyenesbe rendeződjenek.

Persze nem tudtam rögtön érdemi munkát végezni, előbb el kellett sajátítanom bizonyos szaktudást, hogy hozzáláthassak a konkrét feladatokhoz. Mindenekelőtt programozási tudásomat kellett elmélyítenem, mivel korábban nem volt benne túl nagy gyakorlatom. Meg kellett ismernem a repülés, a statisztikus fizika, a csoportos viselkedés szakzsargonját, nagy vonalakban meg kellett tanulnom a kopterek felépítését, a vezérlés szabályozásának elvét. Megtudtam, hogy milyen fizikai paraméterekkel jellemezhető egy csoportos mozgás, hogyan viszonyulnak egymáshoz a mozgásban résztvevő egyedek, és milyen lényeges különbségek vannak a természetes és a mesterséges rendszerek között. Szembesültem azzal, milyen nehézségekkel jár, ha a levegőben kísérletezünk. Mindezek után kezdtem csak el dolgozni, eközben viszont Virágh Csaba jóvoltából elkészült az első próba modell, ami azóta rengeteg változtatáson ment keresztül. A munka során egymásra épülő feladatokat kell megoldanunk, ezért továbblépés előtt lehetőség szerint ki kell javítani a felmerülő problémákat.

A dolgozatom az általunk használt szimulációs környezettel, és az abban vizsgálható modellekkel foglalkozik. Bemutatja az általam elkészített alakzatrepülési modellek felépítését és működését, valamint ezek valóságban történő tesztelését. Kitekintést nyújt a felmerülő problémákra, valamint az ilyen típusú kutatások gyakorlati hasznára.

## 2. fejezet

# Szimulációs környezet

Mielőtt a koptereket a valóságban teszteljük, nagyon fontos, hogy legyen egy virtuális szimulációs környezet, ahol a kritikus hibával járó működést észlelhetjük, és még az éles kísérlet előtt kijavíthatjuk. Ezért lényeges, hogy a szimuláció a lehető legjobban rekonstruálja a valós körülményeket, és tudjunk rá alapozni.

Egyelőre úgy kísérletezünk, hogy a kopterek felszállás után jó közelítéssel egy síkban, a magasság változtatása nélkül repülnek. Szerencsére így elegendő a szimulációt is két dimenzióban futtatni, ami nagyban megkönnyíti a számítások elvégzését, és csökkenti a paraméterek számát. Természetesen később szeretnénk három dimenzióra is kiterjeszteni mind a szimulációt, mind a kísérleteket, de addig még hosszú az út. Mielőtt áttérnénk 3D-be, a 2D-s repülést kell tökéletesítenünk, hogy a kopterek stabilan, jó hatékonysággal hajtsák végre a feladatokat.

A számítógépes szimuláció egy kiváló lehetőség arra, hogy biztonságos körülmények között kimerítsük a 2D-s repülés lehetőségeit, és kipróbáljunk különböző feladatokat. A jelenleg használt szimulációs környezetet Virágh Csaba fejlesztette ki.

Ez a fejezet két részre oszlik. Az első rész a kopterek fontosabb tulajdonságait foglalja össze a szimulációs környezetben bevezetett függvények érthetősége végett. A második rész kitér a szimulációs program lényegesebb összetevőire, és bemutatja a használatát.

### **A koptereink felépítése**

Ebben az alfejezetben röviden ismertetem az általunk használt kvadrokopterek felépítését, valamint a szimuláció szempontjából fontos tulajdonságait.

A koptereink *Mikrokopter MK L4-ME* típusú kvadrokopterek. Hely- és helyzetváltoztatásuk

viszonylag gyors, jó pontossággal tudják tartani a pozíciójukat, és képesek egy helyben forogni. A vezérlés egyik része a motorok fordulatszámának változtatása a mozgás irányának megfelelően, ezt a gyári hardver és szoftver teszi meg. Egy magasabb szintű vezérlő rendszer ennek küld egy bemeneti távirányítójel értéket, amely egy dőlésszöget kódol. A kopternek a manőverezéshez elég ezt a dőlésszöget megváltoztatni. A kívánt bemeneti távirányítójel értéket saját fejlesztésű szoftverünk számolja. Ez két fő részre osztható, mégpedig egy mozgáspálya-tervező és egy sebesség-szabályzó algoritmusra. A mozgáspálya-tervező algoritmus lényegében az éppen aktív modell alapján számol egy  $v_{pref}$  preferált sebesség értéket, majd ezt továbbadja a szabályzó algoritmusnak, ami kiszámolja az ennek eléréséhez szükséges virtuális távirányítójel értéket [7].

Ennek a vezérlésnek a hardveres központja egy *Gumstix Overo Water* típusú fedélzeti számítógép. A *UNIX* operációs rendszerrel ellátott gép processzorának órajele *720 MHz*, a memóriájának mérete pedig *512 MB*. Ez a számítógép a gyári panellel *SPI* soros porton keresztül kommunikál.

Ahhoz, hogy a tervező algoritmus ki tudja számolni a  $v_{pref}$  sebességet, szüksége van bizonyos bemeneti adatokra. Ezeket az adatokat a kopterekre szerelt különböző érzékelők szolgáltatják: GPS vevő egység, giroszkóp, nyomásmérő, gyorsulásmérő, magnetométer. Ezek segítségével az egyedek meg tudják határozni saját térbeli pozíciójukat, sebességüket, gyorsulásukat, szögsebességüket és irányukat. A kopterek a pozíció- és sebesség adatokat a kommunikációhoz használt *XBee 2.4 GHz* -es rádió adó-vevőn keresztül a közelben lévő többi kopter számára is elérhetővé teszik.

A felsorolt műszaki tulajdonságokat valamilyen módon figyelembe kell venni a szimulációban is, hiszen a helymeghatározás pontatlan, a kommunikáció és az adatok feldolgozása pedig időt vesz igénybe, ami késleltetéseket okoz. Nem szabad megfeledkezni a robotok fizikai korlátairól sem. A kopterek motorjainak van egy maximális teljesítménye, ami korlátozza a sebességüket, a gyorsulásukat pedig ezen kívül tehetetlenségük is befolyásolja. Éppen ezért a szimulációban is be kell állítanunk felső határt a sebességre és gyorsulásra.

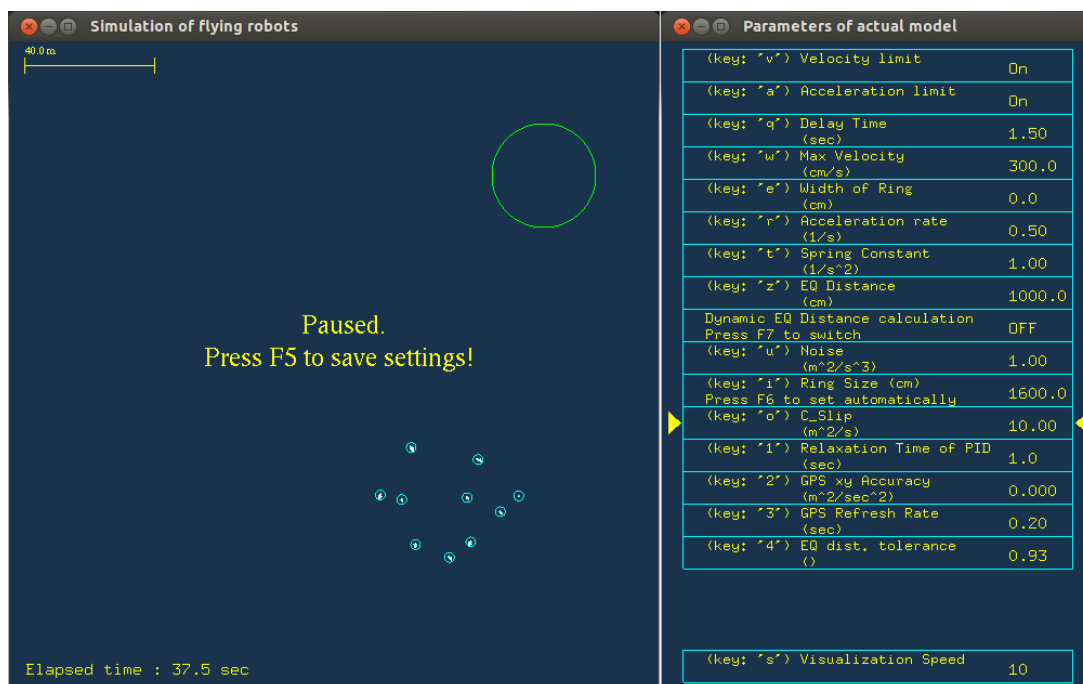
## **A program összetevői és működése**

Az általunk használt szimulációs program három fő összetevőből áll: egy modellező részből, a grafikus felületből és egy statisztika készítő részből. A program nagy része C nyelven íródott.

A modellező rész felhasználói szemszögből láthatatlan összetevő, mert csak forráskód formájában található meg. Ennek a résznek a megírását kell mindenképp elvégezni, amikor modellt készítünk a szimulációhoz.

A vizuális megjelenítés *OpenGL* [8] alapú. Segítségével nyomon követhető az egyedek

trajektóriája, és bizonyos paramétereket futási időben is állíthatunk. A grafikus felület egyes elemeit modellhez illően testre szabhatjuk a modellező részben, például a menü sorait (4. ábra).



4. ábra. A szimulációs program grafikus felülete

A statisztika készítő rész szorosan összekapcsolódik a modellező résszel, mivel minden egyes modellhez célszerű egyéni statisztikát készíteni. A beépített alap statisztikai függvényekről a következő fejezetben lesz szó. A program a statisztikai adatokat külön fájlba menti, és képes külső alkalmazás – esetünkben *gnuplot* – segítségével azokat grafikonon kirajzolni.

A program Linux operációs rendszert és egy *OpenGL* csomag (pl. *freeglut*) meglétét igényli. A fordítást egy ún. *Makefile* segítségével végezhetjük el, az alkalmazás ezután terminálból indítható. Futtatáskor meg kell adnunk neki legalább egy bemeneti paramétert, ami alapján kiválasztja az indítás módját. Háromféle módban tud elindulni:

- Nincs vizuális megjelenítés, de készül statisztika.
- Van vizuális megjelenítés, de nem készül statisztika.
- Van vizuális megjelenítés, és a mentett adatokkal statisztika is készül.

Mint említettem, a grafikus megjelenítés egyelőre 2 dimenzióra korlátozódik. A jelenlegi körülmények között azonban ez teljesen megfelelő, nem származik hátrányunk a 3D-s megjelenítés hiányából. Így nem csak a szimuláció, hanem a statisztika is könnyebben átlátható, informatívabb.

Ebben a környezetben készítjük el és próbáljuk ki tehát a modelljeinket. Hogy hogyan is építettük fel a modelleket, amelyek végül hasznos szimulációkat produkálnak, arról a következő fejezet szól.

## 3. fejezet

# Az alakzatba rendeződés modelljeinek megalkotása

Egy fizikai modell megtervezésekor alaposan át kell gondolni, hogyan állítjuk azt össze. Pontosan tudnunk kell, hogy a modell milyen célt szolgál, vagyis hogy milyen viselkedést várunk el tőle. Mindenekelőtt fel kell ismerni, hogy milyen komponensekre bontható az egyed(ek) mozgása, és nem szabad megfeledkezni az egyedek közti kölcsönhatásról sem. Az egyes komponensekhez többváltozós függvényeket definiálunk erőtvények mintájára, azonban ezek a függvények nem gyorsulást, hanem sebességet adnak meg. A matematikai kidolgozás során már célszerű elkülöníteni a változókat, a konstansokat és a paramétereket. Amennyiben realisztikus szimulációra törekszünk, figyelembe kell venni a valódi robotok tulajdonságait, például a késleltetést, a kommunikáció hatótávolságát, vagy a helymeghatározás pontosságát, továbbá a környezeti zajokat.

Az általunk használt szimulációs környezetben a kiinduláshoz segítséget nyújtanak az előre elkészített sablonfájlok, melyek célja, hogy a modellek forráskódjainak szerkezete egységes legyen. Ily módon könnyebben megoldható a hibakeresés és az összehasonlítás.

Az alakzatba rendezést az egyszerű, elemi alakzatokkal célszerű kezdeni, majd miután ezeket tökéletesen sikerült végrehajtani, áttérhetünk az ezeken alapuló bonyolultabb formákra is. Mi kezdetnek két alap elrendezést kezdtünk el megvalósítani: a sokszöget és az egyenest.

Ebben a fejezetben először röviden ismertetem a sablonok tartalmát, illetőleg a köztük levő lényeges különbségeket, majd rátérek a konkrét modellek megalkotására. Részletes áttekintést adok az általam készített alakzat rendező modellek működéséről. A leírás során mindvégig törekszem a programozási eljárás ismertetésének mellőzésére. A forráskód részleteinek közlését hanyagolom, helyette csak a matematikai formulákat ismertetem.

## Első lépések: a sablonfájlok

A szimuláció sablonfájljai minden olyan alap összetevőt tartalmaznak, amely bármely modell létrehozásához szükséges. Ilyenek az általános globális változók, melyek minden modell esetén érvényesek, például a robotok sugara ("veszélyzónával"), a kommunikáció hatótávolsága, a résztvevő egyedek száma, a sebességhatár nagysága, stb. Helyet kapnak még a futtatásban szerepet játszó függvények, mint a léptetés és a grafikus elemek kirajzolása. Néhány statisztikai függvény is be van építve, többek közt az ütközések számlálása, a korreláció és a sebesség átlaga és szórása.

Ezen sablonok óriási előnye, hogy nem kell minden egyes alkalommal nulláról kezdeni a munkát, amikor új modellt hozunk létre. Természetesen a sablonfájlokban szereplő függvényeket és változókat lehet, és sokszor kell is módosítani, hogy igazodjanak az aktuális modellhez. A legfőbb feladatunk azonban az, hogy erőtvényekkel, modellspecifikus függvényekkel töltsük meg a fájlokat. Ezek végeztével kiválaszthatjuk, hogy mely paramétereket szeretnénk futási időben is változtathatóvá tenni, majd ezekhez menüpontokat szerkeszthetünk, amik majd megjelennek a grafikus felületen. Ezen felül további információt hordozó alakzatokat is kirajzoltathatunk, például a tömegközéppont helyét vagy az egyedek közti kommunikációs kapcsolatokat.

Kétféle sablon áll rendelkezésre. Az egyik nemes egyszerűséggel a *template*, a másik a *realcopt\_template* nevet viseli. A kettő közti különbség a nevükből is látszik. Az utóbbi tartalmaz olyan kiegészítő függvényeket, melyek a valódi kopterek technikai tökéletlenségéből adódó, a dinamikát zavaró tényezőket írják le. Az előbbi ezeket mellőzi, és az egyedeket "végtelenül pontos és gyors" helymeghatározásra, valamint sebesség-változtatásra képes objektumokként kezeli.

Az említett zavaró tényezők a PID szabályozó algoritmus késleltetése, a GPS jel frissítési gyakorisága, és a GPS helymeghatározás pontossága. Tekintsük át most nagy vonalakban, hogy miként vesszük figyelembe a modellekben ezeket a tulajdonságokat!

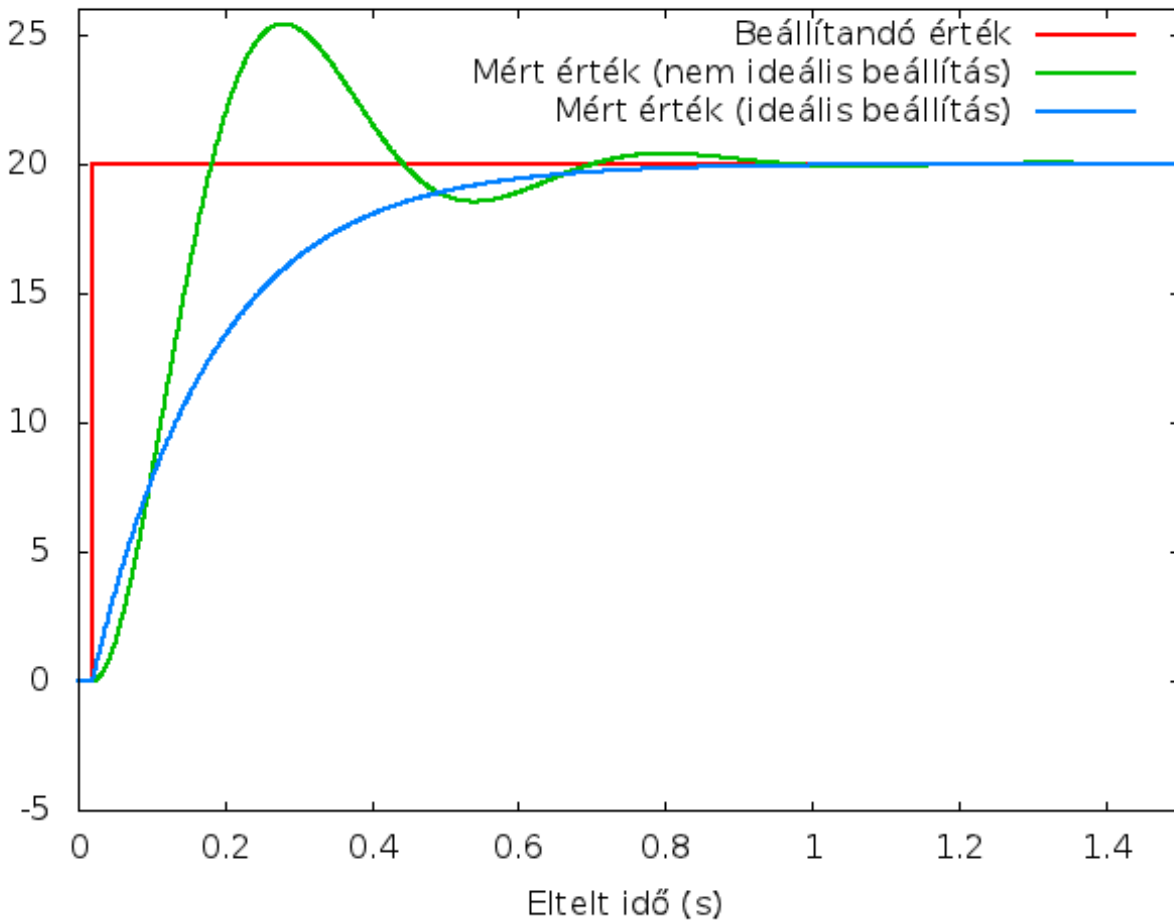
A robotok kívánt sebességének eléréséről ún. *PID szabályozás* gondoskodik. A rövidítés a szabályozó elvére utal, ugyanis a kiadott végrehajtó jel egy *P* proporcionális, egy *I* integrális, és egy *D* differenciális tagból tevődik össze. A *P* tag arányos az ún. hibajellel, az *I* tag a hibajel integráljával, a *D* tag pedig a hibajel deriváltjával, vagyis a végrehajtó jel a jelenlegi hiba, a múltbeli hibák, és a jövőben várható hibák függvénye [9]. A hibajel egy ismert ideális érték és a mért érték különbsége, esetünkben legyen  $e(t) = v_{pref}(t) - v_{kopt}(t)$ , a modellünk által számolt  $v_{pref}$  preferált sebesség és a kopterek repülés közben mért  $v_{kopt}$  sebesség különbsége. Ha az  $e(t)$  függvényt adjuk a PID bemenetének, akkor az  $u(t)$  kimenet a következőképpen alakul [7]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Ahol  $K_p$ ,  $K_i$  és  $K_d$  arányossági tényezők a rendszer paraméterei. A kopter valódi sebessége ezen  $u(t)$  kimeneti végrehajtó jel függvényében változik meg, az algoritmus tehát egy visszacsatolást tartalmaz [7]:

$$v_{kopt}(t) = \phi(u(t))$$

Ahol  $\phi(u)$  a rendszerre jellemző átviteli függvény. Tegyük fel, hogy  $v_{pref}$  értéke konstans. Ez esetben a  $K_p$ ,  $K_i$  és  $K_d$  paraméterek változtatásával a kopter sebessége különbözőképpen áll be erre a preferált sebességre. Ezt figyelhetjük meg az alábbi ábrán (5. ábra):



5. ábra. A PID algoritmus együtthatóinak behangolása. A grafikon sebességadatokat ábrázol az idő függvényében. A piros vonal a preferált sebesség, a zöld és kék vonalak pedig a mért sebességek különböző együttható-beállítások esetén.

Látható, hogy a paraméterek ideális behangolása esetén egy exponenciális jellegű konvergencia lép fel, mely egy  $\tau_{PID}$  relaxációs idővel jellemezhető. A szimulációban ezt vesszük alapul,  $\tau_{PID}$  értékét pedig paraméterként kezeljük, mint egyfajta késleltetést. Virágh Csaba diplomamunkájában [7] részletesen vizsgálta a  $\tau_{PID}$  különböző beállításainak hatását az egyedek viselkedésére, és kifejtette az így konstruált dinamikai rendszer univerzalitását. A robotjainkkal

végzett mérésekből kiderült, hogy  $\tau_{PID}$  értékét 1-2 s közöttire érdemes venni. Ez a beállítás azonban csak a mi rendszerünkre igaz, nem általános érvényű.

A GPS alapú helymeghatározásnak két lényeges tulajdonsága a jel frissítési gyakorisága és az érzékelt pozíció pontossága. Előbbi abban nyilvánul meg, hogy az egyedek milyen gyakran kapnak információt a saját pozíciójukról, és hogy mekkora időközönként tudják változtatni a  $v_{pref}$  sebességet. Utóbbi pedig annak mérőszáma, hogy legfeljebb mekkora lehet a valós és az érzékelt pozíció ill. sebesség közti különbség. A GPS pontatlanságát vagy bizonytalanságát nevezzük mi "belső zajnak". A szimulációban mindkettőt külön paraméterként vesszük figyelembe, hatásuk – ahogy a PID késleltetése is – be van építve a dinamikába. Az ezzel kapcsolatos részleteket tartalmazza egy 2013 őszén leadott angol nyelvű tudományos cikkünk [10].

A továbbiakban alapul veszem, hogy a modellekben jól működően használhatóak a "realcopt" függvények a fenti tulajdonságokra építve, és további indoklást nem fűzök hozzájuk.

## **Általános kölcsönhatási tagok a modellekben**

*Flocking*nak nevezzük nagy számú önhajtott részecske koherens kollektív mozgását [11]. A csoportos mozgás eme formájának leírására többféle matematikai modell létezik. Reynolds modelljében az egyedek a következő egyszerű szabályokat alkalmazzák repülés közben [12]:

- Elkülönülés – Szomszédokkal való ütközés elkerülése (rövidtávú taszítás).
- Sebesség összehangolása – Szomszédok mozgásával egy irányba történő haladás.
- Kohézió – Együtt maradás a társakkal (hosszútávú vonzás).

Fontos hangsúlyoznunk, hogy ebben a modellben nincs központi irányítás. Minden egyed csupán a saját maga által érzékelt környezettel tud kölcsönhatni, így a csoport mozgása autonóm módon szerveződik.

Ezt próbáljuk megvalósítani a szimulációban és a valódi koptereinkkel egyaránt. Ezért bármely modellünk dinamikai rendszere a raj feladatától függetlenül tartalmazza a fenti feltételek teljesítéséhez szükséges erőtagokat. Az ütközések elkerüléséért két általános kölcsönhatási tag felelős, melyek az egyedek között hatnak. A sebesség összehangolásában az egyik alkotóelem egy súrlódásos tag (mely az egyik "általános kölcsönhatás"), a másik pedig egy közös célhoz tartás. A célhoz tartás és a kohézió szinte tetszőleges potenciálfüggvény szerint megvalósulhat, az alkalmazott erőtvörényt a feladat jellege dönti el. Lényegében ezen a két függvényen múlik a feladat sikeres végrehajtása. A kohézió úgy alakul ki, hogy minden egyed a szomszédainak átlagos pozíciója felé, vagyis általában a tömegközéppont felé tart. Ezeket egészíti ki a kommunikáció késleltetése, a külső zaj, az előző alfejezetben említett PID késleltetés és a belső zaj.

Bár a dolgozatnak nem célja az összes erőtag ismertetése, a dinamika átlátásához szükséges ezek rövid leírása tájékoztatás szintjén. A tömegközéppont, ill. a cél felé tartás már konkrét modellekhez tartozik, így azokat külön alfejezetben tárgyalom. Ebben a részben pedig elsősorban az általános kölcsönhatási tagokat ismertetem, majd kitérek a késleltetés és a zaj szerepére.

A két általános kölcsönhatási tag közül az egyik egy helyfüggő párpotenciál, a másik pedig egy súrlódás jellegű tag. Az általunk definiált *párpotenciál* olyan rövid távú taszítás, mely akkor kezd el hatni, amikor két egyed közti távolság egy adott értéknél kisebb. Szigorúan véve viszont nem nevezhető potenciálnak, mivel nem erőt, hanem sebességet ad meg. A különböző szempontok alapján meghatározható és paraméterként tetszőlegesen állítható távolság küszöbértéket *egyensúlyi távolságnak* (*equilibrium distance*) nevezzük, és a továbbiakban  $R_0$ -val jelöljük. A taszítást mi egy egyszerű lineáris erőtvörvény mintájára választottuk abból a megfontolásból, hogy ilyen zajos rendszerben a pozíció mérés hibáiból adódó fluktuációk így a legkisebbek. Ha a kopterek nagy pontossággal, késleltetés nélkül tudnák mérni saját és társaik pozícióját, akkor magasabb fokú függvényeket is alkalmazhatnánk, például a Lenard-Jones potenciált. Az általunk használt centrális lineáris párpotenciál az alábbi alakú [10]:

$$\mathbf{v}_{ij}^{pot} = \left\{ \begin{array}{ll} -D(|\mathbf{x}_j - \mathbf{x}_i| - R_0) \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} & , |\mathbf{x}_j - \mathbf{x}_i| < R_0 \\ 0 & , |\mathbf{x}_j - \mathbf{x}_i| \geq R_0 \end{array} \right\}$$

Ahol  $\mathbf{x}$  az egyedek helyvektora és  $D$  egy rugóállandóval analóg szorzótényező, ezért erre a kölcsönhatásra gyakran *rugóerőként* hivatkozunk. A kölcsönhatás lokalitása abból látszik, hogy csak  $R_0$ -on belül érvényes a törvény.

A *súrlódásos* tag az egyedek egymáshoz viszonyított relatív sebességétől függ. A célja az egyedek sebességének egymáshoz relaxálása, valamint annak megelőzése, hogy a fenti párpotenciál túl nagy oszcillációkat okozzon. Ehhez az alábbi függvényt definiáltuk [10]:

$$\mathbf{v}_{ij}^{slip} = C_{slip} \cdot \left\{ \begin{array}{ll} \frac{\mathbf{v}_j - \mathbf{v}_i}{|\mathbf{x}_j - \mathbf{x}_i|^2} & , |\mathbf{x}_j - \mathbf{x}_i| \geq 1\text{ m} \\ \frac{\mathbf{v}_j - \mathbf{v}_i}{(1\text{ m})^2} & , |\mathbf{x}_j - \mathbf{x}_i| < 1\text{ m} \end{array} \right\}$$

Ahol  $\mathbf{v}$  az egyedek sebességvektora, a  $C_{slip}$  együttható pedig a kölcsönhatás erősségét szabályozza. Annak érdekében, hogy kis távolságoknál elkerüljük a nullával való osztást, 1 m-nél maximalizáljuk a függvény értékét.

Mindkét általános kölcsönhatási tagra igaz, hogy adott  $i$ -edik egyedre az összes  $j$ -edik szomszéd (azaz  $j \neq i$ ) egyed járulékának összege hat.

Most néhány mondat erejéig rátérek a *késleltetés* és a *zaj* szerepére. Mire egy egyed feldolgozza egy szomszédja hely- és sebesség adatait, az információ már elavult lesz, mivel a kommunikáció nem végtelen gyorsan zajlik le, ráadásul a számítások elvégzése is időt vesz igénybe. Az így okozott késleltetést konstansnak tekintjük, értékét mérések alapján határozzuk meg. A szimulációban ezt úgy vesszük figyelembe, hogy az egyedek a szomszédaik állapotától függő számításokat egy paraméterként állítható  $t_{del}$  idővel korábbi időpillanatra végzik. Repülés közben a kopterek technikai tökéletlensége miatt felléphetnek apróbb kilengések, és bármikor előfordulhat, hogy valami véletlenszerű külső hatás éri őket, például egy szélhökés vagy idegen rádióhullám. Ezeket a fluktuációkat a szimulációban egy Gauss-eloszlású  $\eta(t)$  zajvektor hozzáadásával modellezzük, melynek amplitúdóját paraméterként állíthatjuk [10].

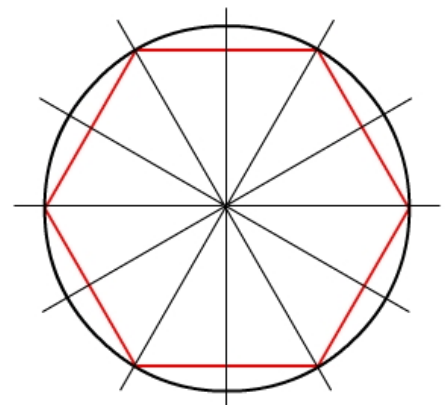
## **Szabályos sokszögbe rendeződés modellezése**

Adott  $N$  darab egyed, melyek ismernek egy célt, és képesek bizonyos távolságon belül egymással kommunikálni. Milyen erőtványeket definiáljunk számukra ahhoz, hogy azokat alkalmazva egy szabályos  $N$  oldalú sokszögbe rendeződjének a cél körül?

Erre a legkézenfekvőbb megoldás, ha megpróbáljuk őket a sokszög körülírt körére terelni, miközben egymástól egyenlő távolságot tartanak (6. ábra). Ha a sokszög egy oldala  $a$  hosszúságú, akkor az  $N$  oldalú szabályos sokszög körülírt körének  $R$  sugarára fennáll az alábbi összefüggés:

$$R = \frac{a}{2 \cdot \sin(\pi/N)}$$

Ez jó kiindulási alap, de mekkorának válasszuk a sokszög oldalait, és hogyan tereljük a körvonalra az egyedeket? A sokszög oldalainak hossza lehet például az  $R_0$  egyensúlyi távolság, ha azt szeretnénk, hogy a kopterek távolsága legyen állandó. Ha a kör sugarát manuálisan akarjuk változtatni, akkor a fenti összefüggést átrendezve a sokszög oldalainak hosszát a sugár függvényében kiszámolhatjuk. A szimulációban a kör sugarának automatikus és manuális beállítását egy kapcsolóval váltogathatjuk.



6. ábra.  $N=6$  egyed által meghatározott szabályos hatszög körülírt köre

Az alakzatba álláshoz létrehozhatunk egy, a célból kiinduló centrális függvényt, melynek csak a körvonal mentén és a célpontban van egyensúlyi pontja. A korábban tárgyalt oszcillációkra gondolva azonban érdemes paraméterként megadni egy ún. "kényelmi zónát" a körtől kis távolságra, melyben a robotok ugyanúgy egyensúlyban vannak, mint a körvonal mentén. Így az egyensúlyi tartomány már nem egy kör, hanem egy gyűrű lesz, ezért neveztük el a sokszögbe rendeződés modelljét **gyűrű modell**nek.

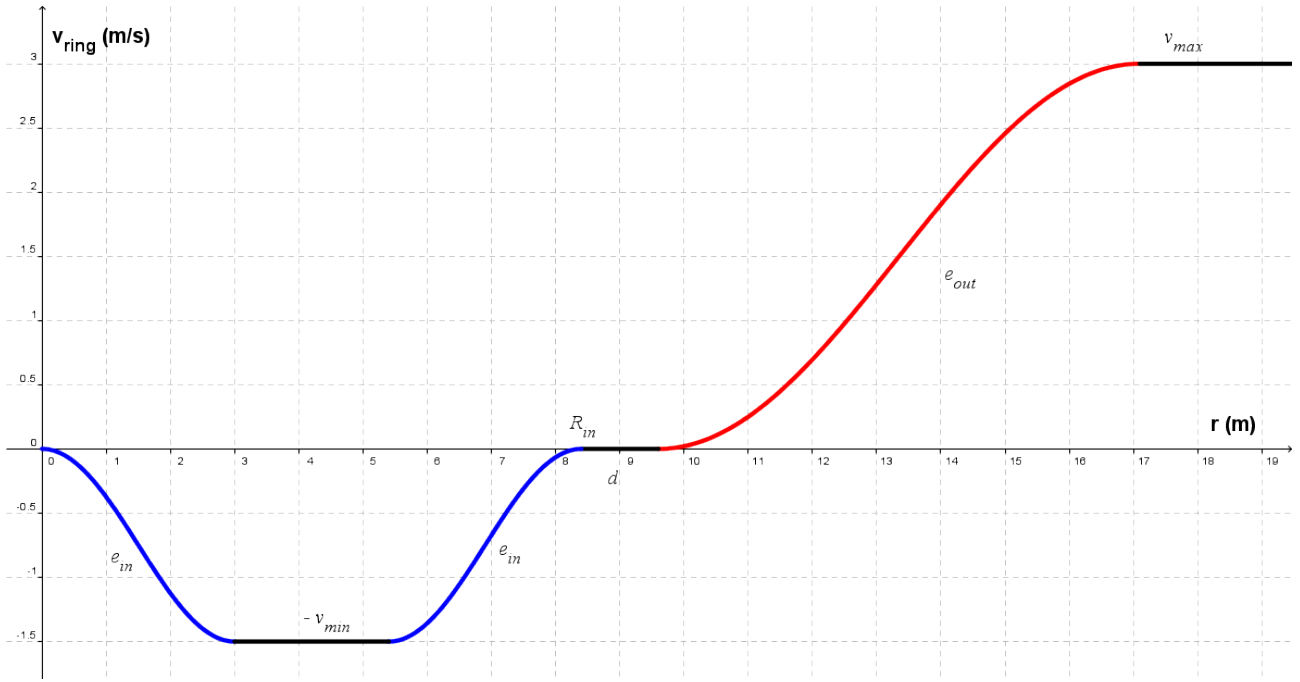
A célhoz tartást a következő képlettel definiáltuk [10]:

$$\mathbf{v}_i^{TRG} = v_{ring} \left( \left| \mathbf{x}_{TRG} - \mathbf{x}_i \right| \right) \cdot \frac{\mathbf{x}_{TRG} - \mathbf{x}_i}{\left| \mathbf{x}_{TRG} - \mathbf{x}_i \right|}$$

Ahol  $\mathbf{v}_i^{TRG}$  az  $i$ -edik egyed sebességének célhoz tartás járuléka,  $\mathbf{x}_{TRG}$  a célpont helyvektora,  $\mathbf{x}_i$  pedig az egyedek helyvektorai, és  $v_{ring}$ -et keressük. A  $v_{ring}$  függvény alakjának keresésekor több szempontot kell figyelembe vennünk. Egyrészt az egyedeknek van sebesség- és gyorsulás korlátjuk, amit nem léphetnek át. Továbbá arra kell törekednünk, hogy a lehető legsimábban, gördülékenyen, ütközések nélkül menjen végbe az alakzatba rendeződés. Az is nagyon fontos, hogy a cél közelében ne alakuljon ki lökdösődés, ne legyenek nagy oszcillációk. Mindezeket figyelembe véve úgy gondoltuk, a legjobb megoldás egy sugaras lépcsős függvény lenne. Ez az egyedekre úgy hat, hogy a gyűrűn kívül egy befelé vonzó erőt, a gyűrűn belül pedig egy kifelé taszító erőt éreznek, a gyűrű sávjában pedig egyensúlyban vannak. Ebből az egyensúlyból csak a külső és belső zajok, valamint a kölcsönhatások téríthetik ki.

Az általunk kitalált és alkalmazott függvény több finomításon átesett már, jelen dolgozat írásának idején az alábbi alakot ölti (a színek a 7. ábra könnyebb érthetőségét segítik):

$$v_{ring}(r) = \left\{ \begin{array}{ll} \frac{v_{min}}{2} \cdot \left[ \sin\left(\frac{\pi}{e_{in}} r + \frac{\pi}{2}\right) - 1 \right] & , 0 \leq r < e_{in} \\ -v_{min} & , e_{in} \leq r < R_{in} - e_{in} \\ \frac{v_{min}}{2} \cdot \left[ \sin\left(\frac{\pi}{e_{in}} (r - R_{in} + e_{in}) - \frac{\pi}{2}\right) - 1 \right] & , R_{in} - e_{in} \leq r < R_{in} \\ 0 & , R_{in} \leq r < R_{in} + d \\ \frac{v_{max}}{2} \cdot \left[ \sin\left(\frac{\pi}{e_{out}} (r - R_{in} - d) - \frac{\pi}{2}\right) + 1 \right] & , R_{in} + d \leq r < R_{in} + d + e_{out} \\ v_{max} & , R_{in} + d + e_{out} \leq r \end{array} \right\}$$



7. ábra. A sokszög alakzatba álláshoz alkalmazott függvényünk grafikonja

A fenti kifejezésben szereplő tagok jelentése:

- $r$  – Az egyed céltól mért távolsága:  $r = |\mathbf{x}_{TRG} - \mathbf{x}_i|$ . (Vízszintes tengely.)
- $v_{ring}$  – Az egyedek ideális sebességének előjeles nagysága a céltól  $r$  távolságra. (Függőleges tengely, a célhoz tartás szorzó faktora.)
- $v_{min}$  – Az egyedek gyűrűn belül megengedett legnagyobb sebessége. (A függvény minimuma, a két kék szakasz közötti vízszintes szakasz.)
- $e_{in}$  – A 0-ról  $v_{min}$ -re (vagy fordítva) történő ideális sebességváltozás alatt megtett út. (A kék szakaszok vízszintes tengelyre vett vetülete.)
- $R_{in}$  – A gyűrű belső sugara. (A második kék szakasz végén  $r$  értéke.)
- $d$  – A gyűrű szélessége, vagyis a külső és belső sugár különbsége. (A második kék szakasz és a piros szakasz közötti rövid vízszintes szakasz  $r$ -en.)
- $v_{max}$  – Az egyedek felső sebességhorlátja a gyűrűn kívül. (A függvény maximuma, a piros szakasz utáni vízszintes rész.)
- $e_{out}$  – A 0-ról  $v_{max}$ -ra (vagy fordítva) történő ideális sebességváltozás alatt megtett út. (A piros szakasz vízszintes tengelyre vett vetülete.)

Ezekből  $d$  és  $v_{max}$  futási időben paraméterként állíthatók,  $v_{min}$ ,  $R_{in}$ ,  $e_{in}$  és  $e_{out}$  pedig a következőképpen számíthatók:

$$R_{in} = R - \frac{d}{2}, \quad e_{out} = \frac{v_{max}}{\mu_{acc}}, \quad e_{in} = \min\left(e_{out}, \frac{R_{in}}{2}\right), \quad v_{min} = e_{in} \cdot \mu_{acc}$$

Ahol  $R$  a körülírt kör sugara, a  $\mu_{acc}$  paraméter pedig hasonló egy relaxációs idő

reciprokához, és a piros színnel jelölt szakasz közepén a függvény meredekségét adja meg.

Persze az, hogy a kopterek egy körvonalra állnak, nem biztosítja, hogy sokszög alakzatot vesznek fel. Az egyenlő távolságok tartásához az előző alfejezetben ismertetett párpotenciált használjuk úgy, hogy az egyensúlyi távolságot a sokszög oldalainak hosszával feleltetjük meg. Így az egyedek tulajdonképpen "szétlökődnek" egymást, amíg mindenki egyenlő távolságra nem lesz a szomszédaitól.

Azonban ha nagyon pontosan akarják tartani a távolságot, akkor a késleltetés és zajok miatt öngerjesztő oszcillációk alakulhatnak ki. Ezért vezettünk be egy ún. *tolerancia* faktort, mely a rugóerő hatótávolságát a számított ideális egyensúlyi távolságnál kisebbre csökkenti. Ez szintén egy akár futási időben is állítható paraméter, és azt adja meg, hogy a párpotenciál az egyensúlyi távolságnak hányszorosáig hasson.

A modellünkben a gyűrű alakzatot az egyedek nem a célhoz érve kezdik kialakítani, hanem már az oda vezető úton. Ezt úgy érjük el, hogy a körülírt kör középpontjának nem a célterület közepét, hanem az egyedek lokális tömegközéppontját választjuk, és a tömegközéppontot irányítjuk a cél felé. Így a gyűrűfüggvény a tömegközépponthez tartást írja le:

$$\mathbf{v}_i^{CoM} = v_{ring} \left( \left| \mathbf{x}_i^{CoM} - \mathbf{x}_i \right| \right) \cdot \frac{\mathbf{x}_i^{CoM} - \mathbf{x}_i}{\left| \mathbf{x}_i^{CoM} - \mathbf{x}_i \right|}$$

A lokális tömegközéppont célhoz tartása pedig így néz ki:

$$\mathbf{v}_{i,CoM}^{TRG} = v_{target} \left( \left| \mathbf{x}_{TRG} - \mathbf{x}_i^{CoM} \right| \right) \cdot \frac{\mathbf{x}_{TRG} - \mathbf{x}_i^{CoM}}{\left| \mathbf{x}_{TRG} - \mathbf{x}_i^{CoM} \right|}$$

A  $v_{target}$  függvény a gyűrűfüggvényhez nagyon hasonló:

$$v_{target}(r_{i,CoM}) = \left\{ \begin{array}{ll} 0 & , 0 \leq r_{i,CoM} < R_{TRG} \\ \frac{v_{max}}{2} \cdot \left[ \sin \left( \frac{\pi}{e} (r_{i,CoM} - R_{TRG}) - \frac{\pi}{2} \right) + 1 \right] & , R_{TRG} \leq r_{CoM} < R_{TRG} + e \\ v_{max} & , R_{TRG} + e \leq r_{i,CoM} \end{array} \right\}$$

Ennek eredménye, hogy az egyedek a tömegközépponthez tartás és a célhoz tartás kölcsönhatásokkal és zajokkal kiegészített vektori összege szerinti sebességgel mozognak. A kopterek

sebességének nagysága természetesen maximalizálva van a meghatározott értéken. Látványban ez annyit tesz, hogy az egyedek az indulást követően egyből elkezdnek körbe rendeződni, és már út közben felveszik a sokszög alakzatot. Ezt az alakzatot tartva jutnak el végül a célhoz.

Ahhoz, hogy az alakzatba állás jóságát számszerűleg is jellemezni tudjuk, definiáltunk egy külön erre a modellre érvényes rendparamétert. A rendparaméter két szempontot vesz figyelembe. Egyrészt azt, hogy az egyedek milyen messze vannak a körülírt kör vonalától, másrészt azt, hogy mekkora a szomszédaikkal bezárt szögük. Tudjuk, hogy tökéletes alakzat esetén az egyedeknek a céltól pontosan  $R$  távolságra kell állniuk, és két szomszédos egyed által bezárt középponti szög nagysága  $2\pi/N$  kell legyen. A rendparaméterünk az ettől mért eltéréseket jellemzi.

A rendparaméter két összetevőjének matematikai alakja:

|                                                                                      |                                                                                          |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Távolság-rendparaméter:                                                              | Szög-rendparaméter:                                                                      |
| $P = \frac{1}{N} \sum_i \text{Min} \left( 1, \left  \frac{R-r_i}{R} \right  \right)$ | $Q = \frac{1}{2\pi} \sum_{i \neq j} \left  \frac{2\pi}{N} - \Delta \varphi_{ij} \right $ |

Ahol  $N$  a résztvevő egyedek száma,  $R$  a körülírt kör sugara,  $r_i$  az  $i$ -edik egyed távolsága a kör középpontjától, és  $\Delta \varphi_{ij}$  az  $i$ -edik egyed és a vele legkisebb pozitív középponti szöget bezáró  $j$ -edik egyed által meghatározott középponti szög (vagyis az  $i$ -edik egyednek a jobb oldali szomszédjával bezárt szöge).

Ezek átlagaként definiáljuk az egyesített rendparamétert:

$$\Omega = \frac{P+Q}{2}$$

Mivel a rendparaméter összetevői a relatív eltéréseket számolják, az értéke 0 és 1 közé esik. A definíciónk szerint  $\Omega=0$  esetén van tökéletes rend, és  $\Omega=1$  esetén nagy rendezetlenség.

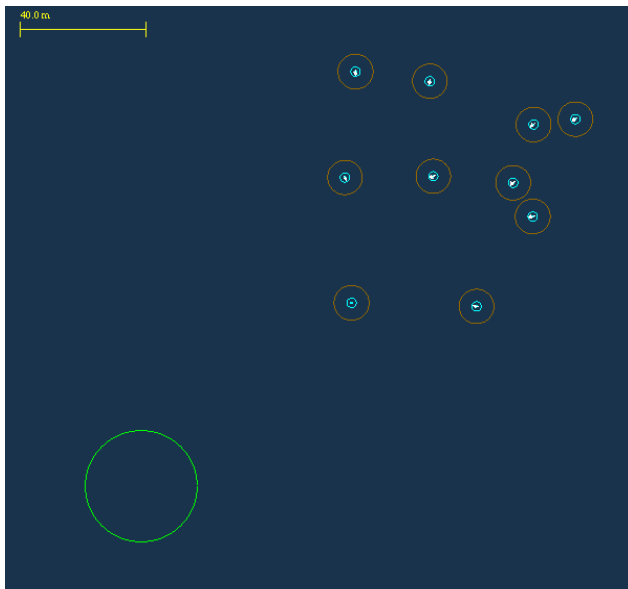
A modell teszteléséhez különböző paraméter beállításokkal futtatásokat végeztünk, a statisztikai adatokat pedig kimentettük külön fájlokba.

A tesztelések alatt a korábban bevezetett tolerancia faktor csökkentése helyett, a stabilitás érdekében megváltoztattuk a körülírt kör sugarának automatikus számolását az alábbi módon:

$$R = \frac{R_0 + 1 \text{ m}}{2 \cdot \sin(\pi/N)}$$

Ezzel a módosítással is biztosítottunk egyfajta toleranciát anélkül, hogy a tolerancia faktor értékét kisebbre vettük volna (tehát a tolerancia faktor végig 1 volt).

Ideális beállításokkal végzett, grafikus megjelenítéssel történő futtatásról készültek az alábbi képernyőfelvételek (8-11. ábra).



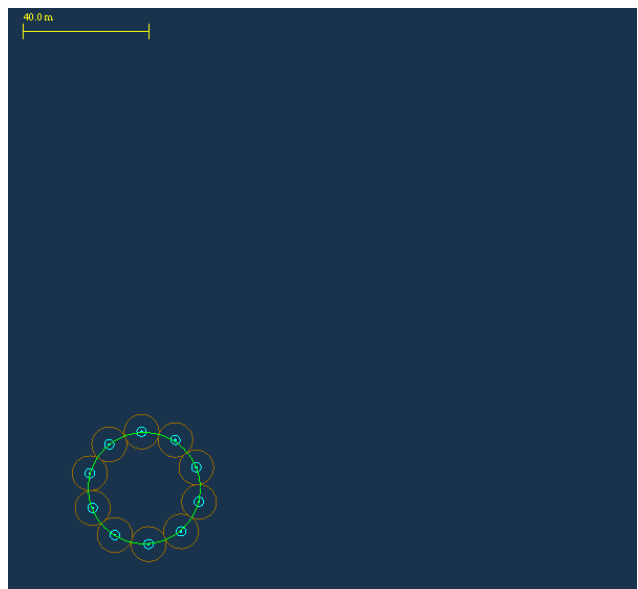
8. ábra. A gyűrű alakzat szimulációja (1)



9. ábra. A gyűrű alakzat szimulációja (2)



10. ábra. A gyűrű alakzat szimulációja (3)



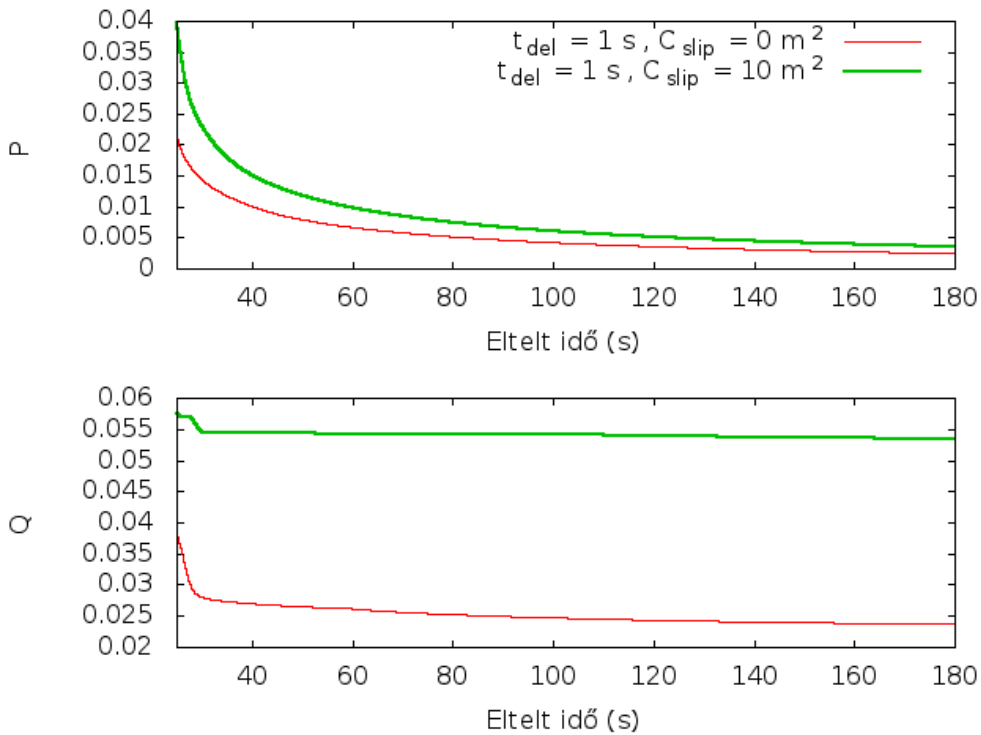
11. ábra. A gyűrű alakzat szimulációja (4)

*Jelmagyarázat: A nagy zöld kör a célterületet jelzi, erre kell ráállniuk az egyedeknek. Az egyedek sebességvektorát kis fehér nyilak jelzik. A kis kék körök az egyedek veszélyzónái. Ha két egyed veszélyzónája összeér, azt ütközésnek számoljuk. A barna színű körök az egyedek egyensúlyi távolságának felével megegyező sugarú körök.*

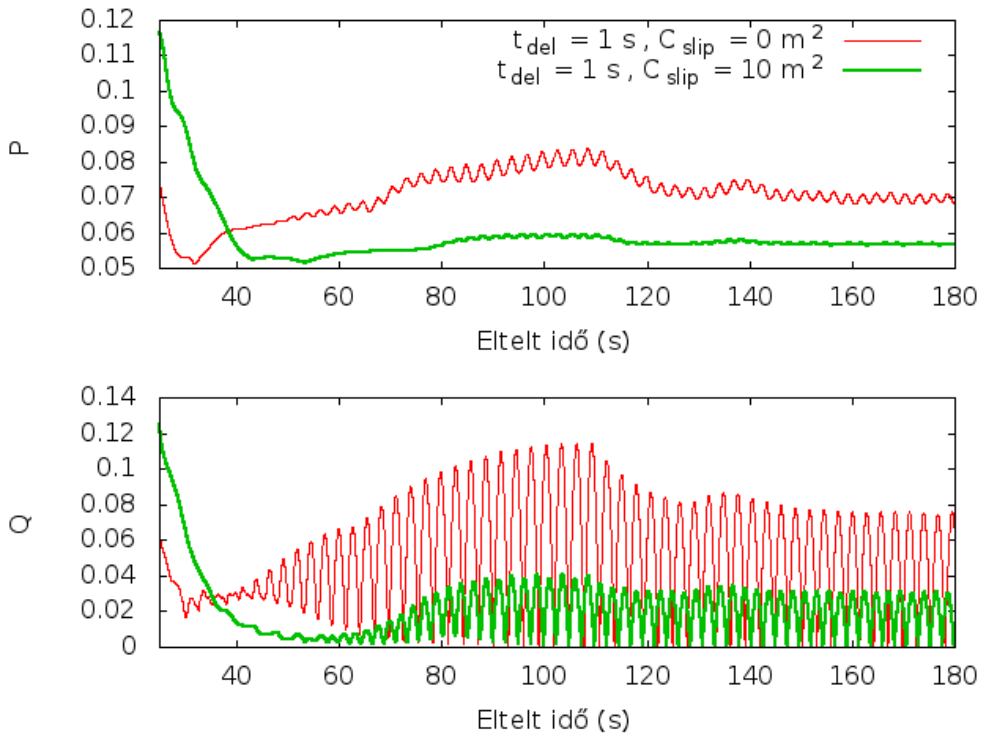
Látható, hogy késleltetés és zaj nélküli esetben zökkenőmentesen, szépen kialakul a tökéletes sokszög alakzat. A valóságos helyzethez egyre jobban közelítő beállításokkal azonban fokozatosan bonyolódik a helyzet, a formáció instabilabb. Ezért nagyon fontos, hogy az alkalmazott függvények és a modell paraméterei jól legyenek kitalálva és behangolva. A megfelelő beállítások megtalálásához sok szimulációt kell végrehajtani, statisztikákat kell készíteni és azokat elemezni.

Az alábbiakban néhány,  $N=10$  egyeddel végzett statisztikai mérés eredményeit sorolom fel (12-15. ábra). A statisztikák készítéséhez a szimulációkat 180 másodpercnyi időlépésre futtattuk le.

Vizsgáltuk a felfebb definiált  $P$  távolság-rendparaméter és  $Q$  szög-rendparaméter időbeli alakulását. A futtatásokat automatikus és az automatikusnál kisebb kör mérettel, valamint súrlódással ( $C_{slip} = 0$ ) és súrlódás nélkül ( $C_{slip} \neq 0$ ) is elvégeztük, minden esetben zaj nélkül ( $\eta = 0$ ).

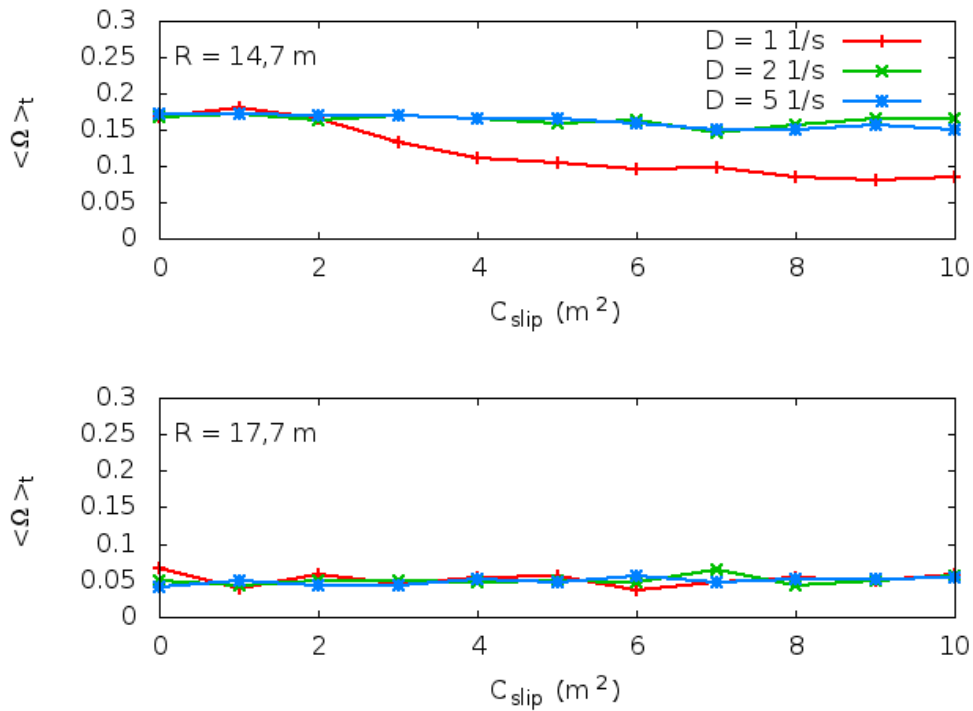


12. ábra. A  $P$  és  $Q$  rendparaméterek időbeli változása automatikus  $R = 17.7$  m sugarú kör esetén, súrlódás nélkül és súrlódással. A többi paraméter értéke:  $t_{del} = 1$  s,  $\tau_{PID} = 1$  s,  $R_0 = 10$  m,  $\mu_{acc} = 0.5$  s<sup>-1</sup>,  $v_{max} = 3$  ms<sup>-1</sup>,  $d = 0$  m,  $D = 1$  s<sup>-1</sup>.

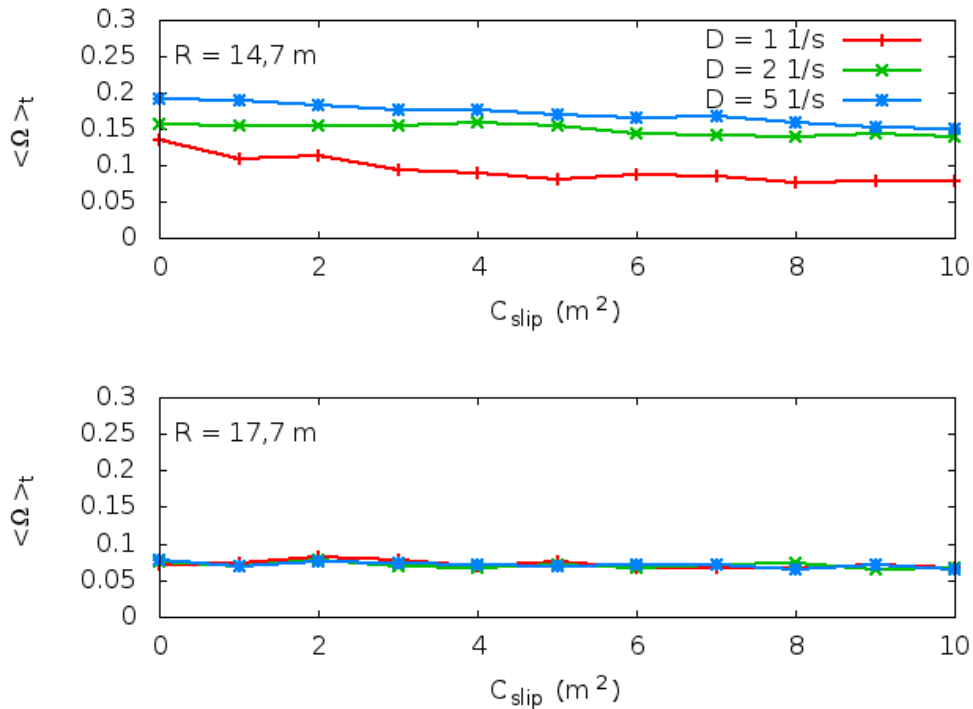


13. ábra. A  $P$  és  $Q$  rendparaméterek időbeli változása az automatikusan számoltnál kisebb  $R = 14.7$  m sugarú kör esetén, súrlódás nélkül és súrlódással. A többi paraméter:  $t_{del} = 1$  s,  $\tau_{PID} = 1$  s,  $R_0 = 10$  m,  $\mu_{acc} = 0.5$  s<sup>-1</sup>,  $v_{max} = 3$  m/s,  $d = 0$  m,  $D = 1$  s<sup>-1</sup>.

Az egyesített rendparaméter a  $C_{slip}$  súrlódási együttható függvényében vizsgáltuk, különböző  $D$  rugóállandók esetén, automatikus és az automatikusnál kisebb kör méretre is. Először zaj nélkül ( $\eta = 0$ ), majd nagy külső zaj jelenlétében ( $\eta \neq 0$ ) végeztük el a méréseket.



14. ábra. Az egyesített rendparaméter teljes futtatásra vett időátlaga a  $C_{slip}$  súrlódási együttható függvényében különböző  $R$  és  $D$  esetén, zaj nélkül. A többi paraméter értéke:  $t_{del} = 2$  s,  $\tau_{PID} = 1$  s,  $R_0 = 10$  m,  $\mu_{acc} = 0.5$  s<sup>-1</sup>,  $v_{max} = 3$  ms<sup>-1</sup>,  $d = 0$  m.



15. ábra. Az egyesített rendparaméter teljes futtatásra vett időátlaga a  $C_{slip}$  súrlódási együttható függvényében különböző  $R$  és  $D$  esetén, nagy zajjal. A többi paraméter értéke:  $t_{del} = 2$  s,  $\tau_{PID} = 1$  s,  $R_0 = 10$  m,  $\mu_{acc} = 0.5$  s<sup>-1</sup>,  $v_{max} = 3$  ms<sup>-1</sup>,  $d = 0$  m.

Azt láthatjuk, hogy az automatikus kör méret esetén a rendparaméterek értéke sűrűlódással és anélkül is nagyon hamar beáll egy 0 közeli konstans értékre, vagyis gyorsan kialakul a rend. A sűrűlódás nagysága ez esetben nem befolyásolja jelentősen a rendparaméterek értékét.

Egészen más a helyzet az automatikusnál kisebb kör méret esetén, mivel a párpotenciál miatt nem férnek el jól az egyedek az alakzatban, s ez meg is jelenik a rendparaméterekben. A kisebb kör méreteknél az egyesített rendparaméter értékei nagyjából kétszer akkorák, mint az automatikus körnél. Ekkor a  $C_{slip}$  nagysága már sokat számít. Lényegesen csökkenti az oszcillációkat, melyek a rendparaméterekben is megjelennek, és minél nagyobb a  $C_{slip}$ , annál jobban stabilizálja, rendben tartja a rendszert. Érdekes megfigyelni, hogy a zaj jelenléte nem hat számottevően a rendparaméterekre. A rugóállandókra nézve azt állapíthatjuk meg, hogy nagyobb  $D$ -hez nagyobb  $\Omega$  tartozik. Ez érthető, hiszen nagy  $D$  érték beállításával a kopterek erőteljesebben lökik el magukat egymástól, és ez növeli az oszcillációkat, csökkenti a rendet.

A gyűrű modell, habár elég jól működik, még egyáltalán nem tekinthető befejezettnek. A legnagyobb probléma a teljes lokalitás követelménye. Céljaink között előkelő helyen szerepel az, hogy olyan rendszert hozzunk létre, melynek egyedei globális információk ismerete nélkül, vagyis kizárólag saját maguk által gyűjtött lokális információk alapján hoznak döntéseket. Az ily módon csoportosan végrehajtott feladatot nevezhetjük teljesen önszerveződő, autonóm folyamatnak.

A jelenlegi gyűrű modell ezt még nem jól teljesíti, mert a teljes egyedszámot globális információként kapják. Ha ugyanis a kommunikáció hatótávolsága kisebb, mint a létrehozni kívánt alakzat mérete, akkor az egyedek nem látnak mindenkit, és ezért egyes számításokat hibásan végeznek el, ha lokális egyedszámot számolnak. A cél közelében így káosz alakul ki, az alakzatba rendeződés nem valósul meg. Ezenkívül a késleltetések és zajok miatt bekövetkező fluktuációk csökkentése is fontos.

De mire lehet jó egy halom repülő robot, ami be tud állni egy körbe? Az elképzelhető alkalmazási területek közé tartozik például egy térfigyelő rendszer egy épület körül. Egy központi térrész több szögből történő egyidejű megfigyelésére is alkalmas lehet, így akár valódi 3D-s felvételeket is lehetne készíteni. Említhetünk még bekerítési, terület lezárási, illetve szállítási feladatokat, amikor kiemelten fontos, hogy az egyedek jól tartsák az alakzatot.

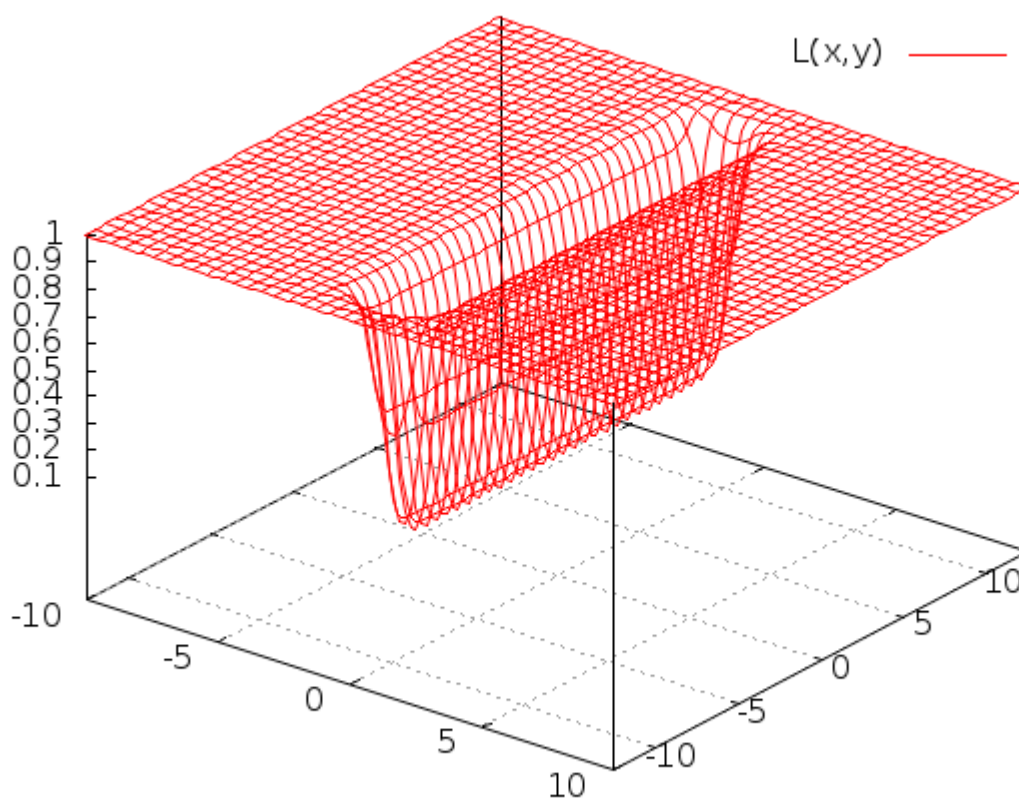
A gyűrű modellt tehát nem csak látványelemként lehet használni, hanem találhatunk számára valódi alkalmazásokat is. A leglényegesebb talán mégis az, hogy megalapozza a bonyolultabb rendszerek felé vezető utat. Rávilágít a nehézségekre és a megoldandó problémákra, és ezek megoldása közben rengeteget haladunk előre. Biztosra veszem, hogy mire tökéletesítjük a gyűrű modellt, a koptereink sokkal többre lesznek képesek, mint egyszerűen sokszögbe rendeződni.

## Egyenesbe rendeződés modellezése

Az egyenesbe rendeződést kicsit máshogyan kell megközelíteni, mint a sokszöget. Először is, a sokszög zárt alakzat, az egyenes pedig nyitott. Ráadásul azonos  $N$  egyedszám és  $R_0$  egyensúlyi távolság esetén az egyenes sokkal nagyobb távolságot felölölő alakzat, mint a sokszög, emiatt a lokális problémájának megoldása nagyon fontos. A tömegközéppont és a cél szerepe is megváltozik, nem alkalmazhatóak centrális erőtvények.

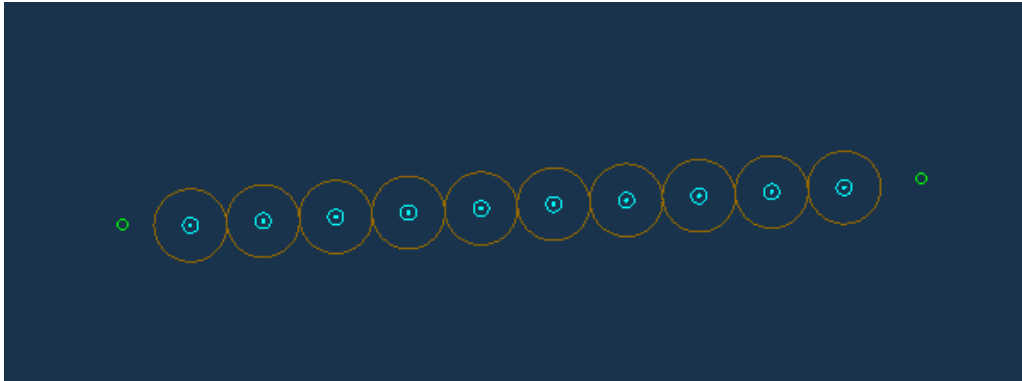
Éppen ezért nagyon összetett feladatról van szó, amelyre többféle megoldás is létezhet, és nekünk meg kell találni a számunkra leginkább megfelelőt. Az **egyenes modell** fejlesztése egyelőre kezdeti stádiumban van, még csak az olyan próbálkozásaink működnek, melyekhez globálisan biztosított teljes egyedszám információ szükséges.

Első közelítésben megpróbáljuk az egyedeket egy virtuális egyenes gödörbe terelni, az egyenlő távolságok tartását pedig szintén a párpotenciállal oldjuk meg. A gödör megkonstruálásához a gyűrű modellben alkalmazott célhoz tartás függvényt használjuk. Több ilyen függvény összeolvasztásából egy az alábbi  $L(x,y)$  szigmoidos gödörhöz nagyon hasonló eredményt kapunk (16. ábra).



16. ábra. Virtuális térbeli szigmoid gödör

A kivitelezés nem egyszerű feladat, még csak részeredményeink vannak ezzel kapcsolatban. Egy ideális beállításokkal végzett próba futtatásról készült az alábbi képernyőfelvétel (17. ábra):



17. ábra. Az egyenes modell szimulációja ideális beállításokkal

Azt láthatjuk, hogy az egyedek a két kis zöld körrel jelzett célpontok között szépen beállnak egy egyenesre, egymástól egyenlő távolságra.

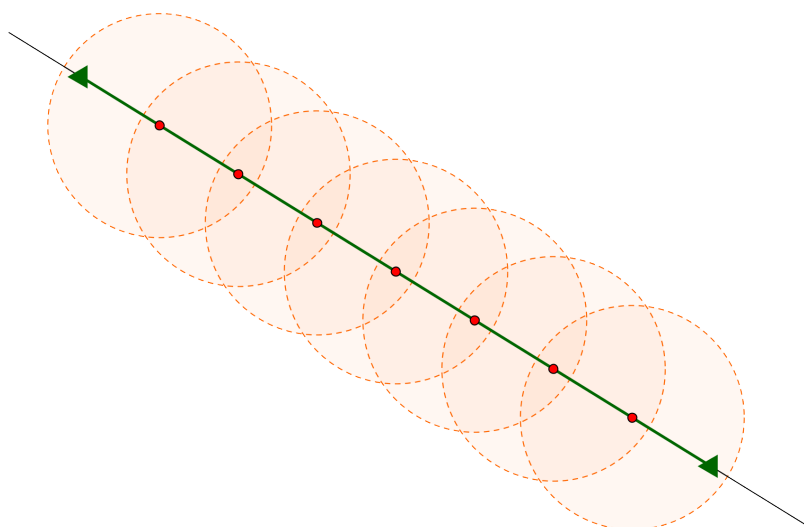
Az egyensúlyi távolságot a következőképpen számolták:

$$R_0 = \frac{|\mathbf{x}_{TRG2} - \mathbf{x}_{TRG1}|}{N + 1}$$

Ahol  $\mathbf{x}_{TRG}$  a célpontok helyvektorai.

Az eredmény látványos, és kezdetnek megteszi, de még sok csiszolnivaló van rajta. Ha az egyenes mérete ugyanis nagyobb a kommunikáció hatótávolságánál, akkor a rendeződés csak akkor valósul meg, ha a koptereknek explicit módon megadjuk a teljes egyedszámot. Célunk, hogy az egyedek akkor is létre tudják hozni a teljes méretű egyenes alakzatot, ha egy vagy több társukkal nem tudják tartani a kapcsolatot.

Az egyenes modell alkalmazható többek között nagy kiterjedésű területek átfésülésére, a kopter csapat szűk helyen történő átrepítésére, illetőleg ad-hoc kommunikációs lánc létrehozására, ahogyan az a lenti ábrán látható (18. ábra). Ehhez egyrészt kulcsfontosságú a lokáltság megoldása, valamint kombinálni kell a dinamikus egyensúlyi távolság állítást az adaptív egyedszám választással. Ez azt jelenti, hogy az egyedek először felmérik, hogy hány kopterrel lehetne ideálisan megvalósítani a láncot, majd egymás között "megbeszélik", hogy kik hajtsák végre a feladatot. A felszálló egyedek pedig egymásra, a szélsők pedig még a célpontra is figyelve határozzák meg az egyensúlyi távolságukat.



18. ábra. Ad-hoc kommunikációs lánc szemléltetése. A zöld háromszögek két célpontot jelölnek, melyek között kommunikációt szeretnénk biztosítani. A kis piros színű körök a közvetítő egyedek, ezek kommunikációs hatótávolságát a narancssárga színű, szaggatott szélű körök jelzik.

Az ilyen bonyolult rendszerekig még hosszú az út, de mindig egyre jobb és jobb modellek készülnek, amivel egyre közelebb kerülünk a céljainkhoz.

## 4. fejezet

# Szimuláció kontra valóság: a kísérletek

A modellek működését jól tudjuk vizsgálni a szimulációs környezetben is, azonban nem hagyatkozhatunk kizárólag a virtuális tesztekre. Természetesen a célunk az, hogy a valóságban hozzuk létre az autonóm drón csapatot, nem pedig szimulációban. Épp ezért elengedhetetlen, hogy a valódi koptereken is kipróbáljuk az algoritmust, mivel csak így tudunk meggyőződni arról, hogy a gyakorlatban is a szimulációban látottak szerint viselkednek az egyedek. Az ilyen tapasztalatok begyűjtéséhez időközönként kísérleteket végzünk, melyeket különféle módszerekkel dokumentálunk is. Lehetőség szerint mozgó- és állóképeket rögzítünk, elmentjük a kopterek log fájljait, valamint reptetési naplót vezetünk. Ezekből később hasznos következtetéseket tudunk levonni, amik segíthetnek az algoritmus jobbá tételében.

Ebben a fejezetben leírom, hogy milyen tesztek előzték meg a konkrét modellek kipróbálását, majd rátérek a gyűű modell tesztelésének eredményeire. A reptetések során felmerülő problémákat a teljesség igénye nélkül tárgyalom, mert kifejtésükhöz ismertetni kellene a kopterek pontos hardver és szoftver információit, ami viszont nem célja jelen dolgozatnak.

### Út a stabil repülésig

Ezen alfejezet címe alapján úgy hangzik, mintha sikerült volna megoldanunk a robotjaink teljesen stabil, vagyis oszcillációktól mentes repülését. Ez azonban távol áll a valóságtól. Mindössze azt mondhatjuk el, hogy eljutottunk egy olyan szintre, ahol már kevésbé kell aggódnunk a kopterek biztonsága miatt, mert az ütközések és lezuhanások veszélye viszonylag kicsi. Az algoritmus jól

működik a megfelelő egyensúlyi távolsággal, a kísérletek során jelentkező hibák nem az algoritmusból erednek. Az alábbiakban összefoglalom azokat a fontosabb vezérléstechnikai problémákat, amelyek megoldása szükséges volt ennek a szintnek az eléréséhez.

A kísérleti berendezés leírásával kezdem. Rendelkezésünkre áll 10 db, a második fejezetben ismertetett felépítésű kvadrokopter, ezekhez 2 távirányító, egy célpontnak használt adó-vevő, egy számítógéphez csatlakoztatható adó-vevő, tartalék akkumulátorok és egyéb kiegészítők. A kopterek két csoportra vannak osztva a könnyebb megkülönböztetés végett. A távirányítók úgy vannak beállítva, hogy csak az egyik csoportnak küldjenek jeleket. A távirányítóval vezérelhető a motorok beindítása, a fel- és leszállás, valamint vészhelyzet esetén át lehet venni az irányítást egy-egy kopter felett, felülbírálván az algoritmust. A földi állomásként működő számítógépen az adó-vevő egység és egy *Ground Control* nevezetű szoftver segítségével nyomon követhetjük a kopterek állapotát repülés közben. Ennek segítségével láthatjuk például az egyedek magasságát, sebességét, gyorsulását, és szinte azonnal észlelhetjük, ha valami hiba lép fel.

A repülés stabilitásának vizsgálatához nem muszáj az összes robotot egyszerre reptetni. Sokáig a mérések többségét kis egyedszámú, legfeljebb 5 fős csoportokkal végeztük. Kezdetben még a helyes fel- és leszállás is gondot okozott. A levegőben a pozíciótartás és a sebesség szabályozás során is felléptek problémák, és az algoritmusba beépített biztonsági protokollok sem működtek megfelelően. Rengeteg mérés kellett hozzá, mire a kopterek szenzorait jól be tudtuk hangolni, és az algoritmus hibáit is kijavítottuk.

Egyetlen önálló robottal hajtottunk végre például sebességteszteket, magasságtesztet, és hibakeresési teszteket. Az egyik sebességteszt során a kopternek két célpont között kellett mozognia. Lassú, 2 m/s -os induló sebességet állítottunk be, és minden forduló után 1 m/s -mal növeltük, amíg el nem érte a 8 m/s -ot. A tapasztalatunk az volt, hogy minél nagyobb sebességgel haladt a kopter, annál kevésbé tudott a két cél közti egyenes mentén közlekedni, és annál pontatlanabban ért célba. A magasságteszt esetében csupán arra voltunk kíváncsiak, hogy változik-e a pozíciótartás stabilitása különböző magasságokban. Nem merítettük ki a kopter elméleti határait, de azt találtuk, hogy 100 méterig nincs gond ezzel.

Az algoritmus biztonsági protokolljai azt határozzák meg, hogy milyen műveletet hajtson végre a robot, ha valamilyen hibajelet kap, vagy elromlik a kommunikáció. Ezeket több mérés során is teszteltük. Egyik alkalommal a hibakeresés úgy zajlott, hogy elindítottunk egy koptert, majd kis idő elteltével virtuális távirányítójel-hiányt generáltunk. A protokoll szerint ilyen esetben a kopternek 10 másodpercig egy helyben kell maradnia, majd függőleges vonalban leszállnia. Ezzel szemben a tesztek alatt furcsa dolgok történtek. Volt, hogy csak 4 másodpercet várt a leszállás előtt, vagy azonnal leszállt, máskor pedig elindult egy látszólag tetszőleges irányba. Végül sikerült kideríteni a szokatlan viselkedés okait, és sok javítást eszközöltünk.

A repülések olyan hiányosságokra és hibákra mutattak rá, amiket a szimulációkból nem derítettünk volna ki. A naplófájlok elemzését és a látottak végiggondolását követően ezeket sikeresen befoltoztuk. Ennek eredményeképpen a robotok már elég megbízhatóan a kívánt magasságba emelkednek a talajról, és leszálláskor biztonságosan érnek földet. Pozíciójukat szélcsendben szinte tökéletesen, és enyhe szélben is még elfogadható pontossággal tartják. A biztonsági protokollok az utóbbi időkben jól működnek, és a kopterek sebesség szabályozása is viszonylag gördülékenyen megy.

Ezek után álltunk csak neki a modellek tesztelésének.

## **A gyűrű modell kísérleti tesztje**

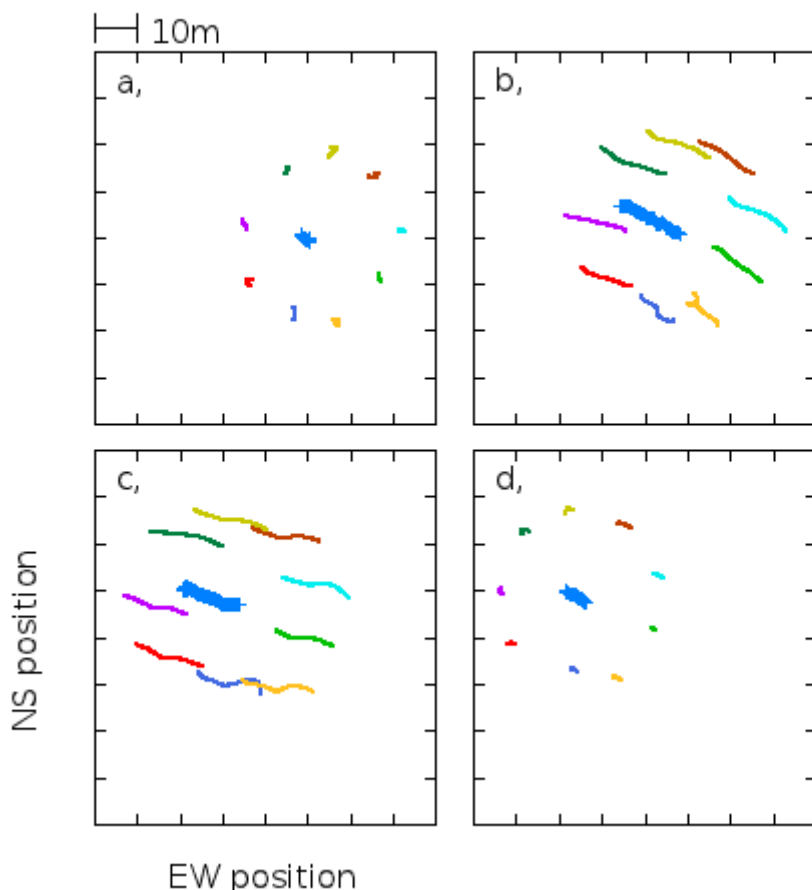
A gyűrű modellt már több alkalommal kipróbáltuk a valódi kopterekkel. Kísérleteztünk 5, 9 és 10 kopterrel is, és meglehetősen pozitív eredmények születtek. Annak ellenére, hogy néha az egyik kopter hibát észlelt és leszállt, a sokszögbe rendeződést minden esetben végrehajtották az egyedek, alkalmazkodtak egyik társuk kieséséhez.

Felszállás után kezdetben nagyok voltak az oszcillációk, de idővel stabilizálódott a rendszer, és csak kisebb rezgések alakultak ki. Az alakzatról fel lehetett ismerni, hogy kört próbálnak alkotni az egyedek, de teljesen szabályosan nyilvánvalóan nem lehet beállni. Ez valószínűleg a GPS pontatlanságának és a korábbi fejezetekben már említett zavaró tényezőknek tudható be. Az alábbi fényképfelvételen azért nem látszik jól az alakzat, mert biztonsági okokból a koptereket különböző magasságokban reptettük (19. ábra).



*19. ábra. A gyűrű modell kísérleti tesztje 9 kopterrel. A követendő célpontot a földön egy munkatársunk egyenletes sebességgel viszi kézben.*

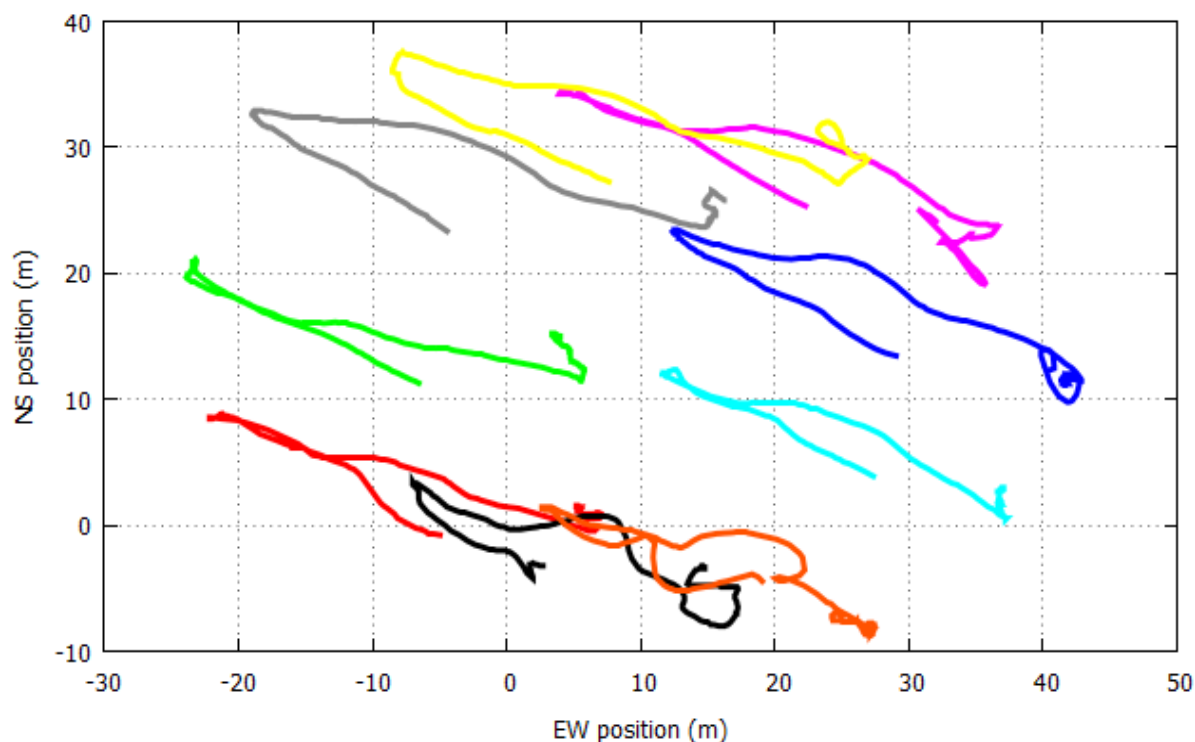
A kísérletek során a kopterek mindig egy log fájlba rögzítik a repülés közbeni adataikat. A log fájlok alapján elemezhetjük a kopterek mozgását, és kirajzoltathatjuk pozíciójukat derékszögű koordináta-rendszerben (20. ábra).



20. ábra. A gyűrű modell tesztelése 9 kopterrel. A grafikonok az egyedek és a célpont pozícióját ábrázolják felülnézetből, a vízszintes tengely a Kelet-Nyugati, a függőleges tengely az Észak-Déli irány. Minden kis négyzet 30 másodperces idősort ábrázol. A vastagabb sötétkék ponthalmaz a követendő cél helyzetét, a többi vékonyabb ponthalmaz a kopterek helyzetét jelöli.

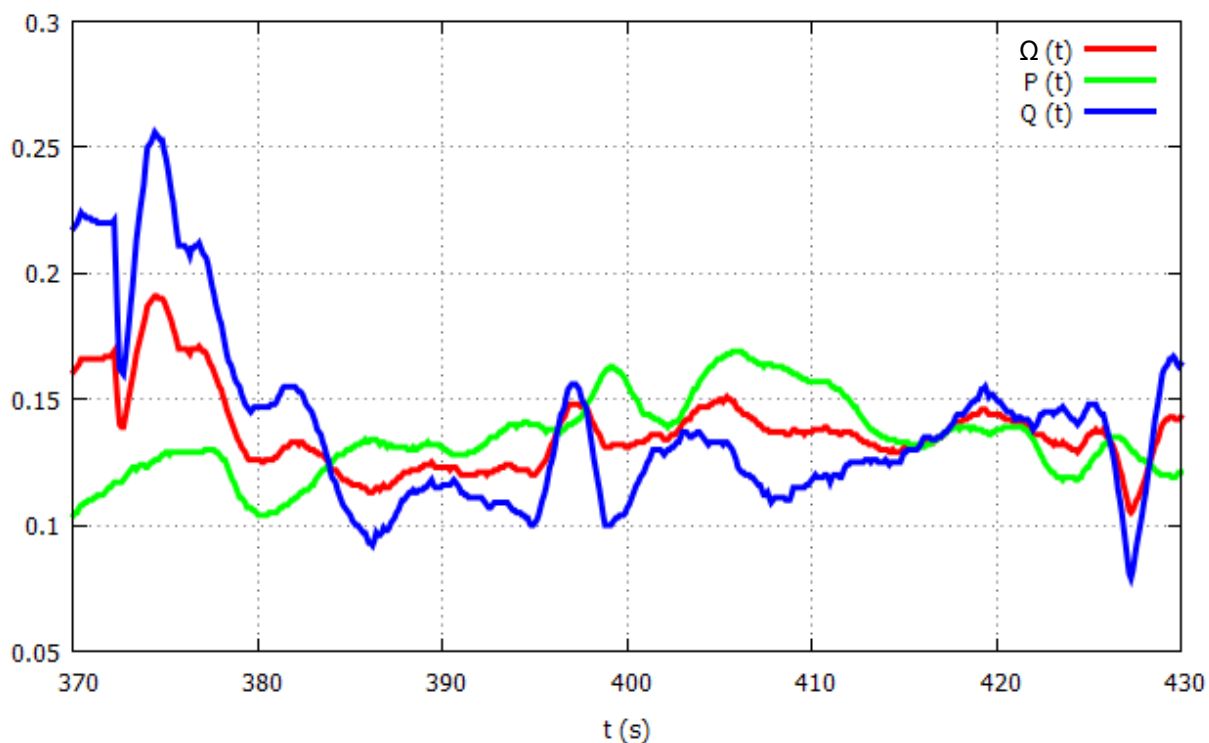
Látható, hogy a kopterek létrehozták a gyűrűt, és a célpont követése közben is jól tartották az alakzatot. Az alakzattartó célkövetést többször is megismételtük, és mindig hasonló eredményeket kaptunk.

Egy másik kísérlet során rögzített adatok egy részletét kiértékeljük a rendparaméterek szerint is, hogy össze tudjuk hasonlítani a szimulációban tapasztaltakkal. A mérést 9 kopterrel végeztük, a körülírt kör sugara automatikus számolással  $R = 16.1 \text{ m}$  volt, a többi paraméter pedig:  $R_0 = 10 \text{ m}$ ,  $\mu_{acc} = 0.5 \text{ s}^{-1}$ ,  $v_{max} = 3 \text{ ms}^{-1}$ ,  $d = 0 \text{ m}$ ,  $D = 1 \text{ s}^{-1}$ ,  $C_{slip} = 10 \text{ m}^2$ . A késleltetéseket és zajokat természetesen nem mi állítottuk be, hanem a körülményektől függően változtak. A következő ábrán az alakzatban repülés egy 60 másodperces részletét követhetjük nyomon (21. ábra).



21. ábra. Alakzatban repülés 60 másodperces részlete. A színes vonalak a kopterek mozgását ábrázolják felülnézetből (Kelet-Nyugat)-(Észak-Dél) irányokhoz rögzített koordináta-rendszerben.

A robotok 0.2 másodpercenként írnak adatot a log fájlba, így 1 perc alatt 300 rekord kerül rögzítésre. A rendparamétereket mind a 300 időlépésre kiszámítottuk az adatokból, majd az eredményről grafikont készítettünk (22. ábra).



22. ábra. Az  $\Omega$ ,  $P$  és  $Q$  rendparaméterek az idő függvényében, 60 másodpercnnyi adat alapján.

Ha összehasonlítjuk a 22. *ábra* grafikonjait a 12. és 15. *ábra* grafikonjaival, azt állapíthatjuk meg, hogy a kísérlet ezen szakaszában a rendparaméterek értékei a szimulációban számítottak többszörösei. Rövid ideig a  $Q$  szög-rendparaméter viszonylag rossz, 0.25 körüli értékeket is elért, majd döntően 0.1 és 0.15 közé szorult. A  $P$  távolság-rendparaméter is hasonlóan alakult azzal a különbséggel, hogy jóval alacsonyabb értékről indult, és kisebb maximumot ért el. Az  $\Omega$  egyesített rendparaméter 0.15 körüli értékei közel kétszer akkorák, mint a 15. *ábrán*.

A paraméterek 60 másodperces időintervallumra vett átlaga és szórása:

$$\langle P \rangle = 0.135 \pm 0.016 \quad , \quad \langle Q \rangle = 0.139 \pm 0.036 \quad , \quad \langle \Omega \rangle = 0.137 \pm 0.016$$

Az összehasonlítás valójában persze nem teljesen jogos, hiszen az egyedek száma különbözik, a késleltetések és zajok pedig folyamatosan változtak a mérés során. Azt is hozzá kell tenni, hogy a szimulációban az egyedek nem egy mozgó célpontot követtek, hanem egy álló célpont körül álltak gyűrűbe. Az eltéréseket valószínűleg a szenzorok pontatlansága, a késleltetések és a környezeti zajok okozták. Ezeket figyelembe véve a valós mérések rendparaméterei nem mondhatók rossznak, mert a feladatot jól végrehajtották a kopterek, az alakzat jóságát pedig csak technikai korlátok csökkentették.

Ezzel megbizonyosodhattunk arról, hogy joggal hagyatkozhatunk a szimulációra, mivel a szimulált és a valós tesztek a különbségek ellenére is hasonló kimeneteket produkáltak.

A kísérletek gyűrű modell szempontjából pozitív kimenetele ösztönzőleg hat. Ezek után még tovább kell tökéletesítenünk a modellt, és az újakat is a lehető legprecízebben kell kidolgoznunk, mert természetesen nem elégszünk meg egy "nagyjából kör" alakzattal. Azonban az elvégzett mérésekre már építhetünk, és a tapasztalatokat felhasználhatjuk. Ezekből a kísérletekből tudjuk, hogy nem feleslegesen fektetünk energiát a fejlesztésekbe.

## 5. fejezet

# Kitekintés

A dolgozat során több helyen is említettem felmerülő problémákat. Ezek egy része továbbra is megoldásra vár, ami valamelyest kijelöli a jövőbeli fejlesztéseink irányát.

A legfontosabb közülük talán a lokalitás. Ez egy kulcsfontosságú részlet, hiszen az egész kutatás az önszerveződés, önrendeződés aspektusában folyik. Elsődleges prioritású feladat tehát a meglévő modellek kijavítása lokalitási szempontok alapján. A megoldások keresésekor igencsak hasznos lehet, ha tanulmányozzuk az állatvilágban megfigyelhető csoportos mozgásokat. Sokszor a mozgásminták elemzésén túl nagy segítséget jelenthet a fajok behatóbb ismerete. Abból is látszik, mennyire kifizetődő a természetből ötleteket meríteni, hogy napjainkra már számos biológia inspirálta technológia létezik. Ezért mi is igyekszünk egyre nyitottabbak lenni a természet felé.

A hibák kijavításán túl természetesen más jövőbeli céljaink is vannak. Törekszünk a bonyolultabb alakzatok önszerveződő létrehozására, szeretnénk megvalósítani akadály kikerülést, objektum bekerítést. Elkezdtünk foglalkozni az ún. metaegyed alkalmazhatóságával, ami például arra használható, hogy folyamatos megfigyelés alatt tartsunk egy területet. A metaegyedek elméletben dinamikusan egymást váltogatnák úgy, hogy egyikük mindig a levegőben marad. Ha azt is meg tudnánk oldani, hogy a leszálló egyed egy dokkolón landolva automatikusan töltsen az akkumulátorát, nem okozna gondot a rövid repülési idő.

Folyamatosan dolgozunk a szimuláció jobbá tételén, és a robotok hardveres fejlesztésén. Felmerült az ultrahang szenzor használata a koptereken, melynek előnye, hogy a környezetet nagy pontossággal képes feltérképezni. Idővel a szimuláció vizuális megjelenítését is szeretnénk kiterjeszteni 3D-be.

A felsorolt hosszú- és rövidtávú céljaink elérésével reményeink szerint egyre közelebb jutunk ahhoz, hogy komplex feladatok megoldására képes autonóm robotrajt hozzunk létre.

# Irodalomjegyzék

- [1] **Benoit-Bird, K. J. and Au, W. W. (2009).** *Cooperative prey herding by the pelagic dolphin, *Stenella longirostris**, The Journal of the Acoustical Society of America 125 : 125.
- [2] **Vicsek, T. and Zafeiris, A. (2012).** *Collective motion*, Physics Reports 517 : 71-140.
- [3] Cell\_Press (2010). *Fish schools and krill swarms take on common shape*, ScienceDaily: <http://www.sciencedaily.com/releases/2010/09/100916121320.htm> .
- [4] **Brierley, A. S. and Cox, M. J. (2010).** *Shapes of krill swarms and fish schools emerge as aggregation members avoid predators and access oxygen*, Current Biology 20 : 1758-1762.
- [5] Soniak, M. (2012). *Why Do Flocks of Geese Fly in a “V” Shape?*, Mental Floss: <http://mentalfloss.com/article/13062/why-do-flocks-geese-fly-%E2%80%9C%E2%80%9D-shape>
- [6] **Thien, H.; Moelyadi, M. and Muhammad, H. (2008).** *Effects of Leaders Position and Shape on Aerodynamic Performances of V Flight Formation*, arXiv preprint arXiv:0804.3879 .
- [7] **Virág, C. (2013).** *Repülő robotraj - szimuláció és kísérlet*, Diplomamunka, ELTE TTK Fizikus MSc .
- [8] **OpenGL.** *OpenGL, The Industry's Foundation for High Performance Graphics*, <http://www.opengl.org/> .
- [9] **Araki, M. (2002).** *PID control*, Control systems, robotics and automation 2.
- [10] **Virág, C.; Vásárhelyi, Gá.; Tarcai, N.; Szörényi, T.; Somorjai, G.; Nepusz, T. and Vicsek, T. (2013).** *Flocking algorithm for autonomous flying robots*, arXiv preprint arXiv:1310.3601 .
- [11] **Toner, J. and Tu, Y. (1998).** *Flocks, herds, and schools: A quantitative theory of flocking*, Physical Review E 58 : 4828.
- [12] **Olfati-Saber, R. (2003).** *A unified analytical look at Reynolds flocking rules*, Submitted to the American Control Conference 2004, Boston, MA .

# Függelék

## Saját munkám tisztázása

Készítettem egy modellt, melynek segítségével  $N$  db egyed egy  $N$  oldalú szabályos sokszögbe képes rendeződni önszerveződő módon. Kidolgoztam a modell matematikai hátterét, és megírtam a hozzá tartozó program kódot C nyelven, mely a szimulációs környezetbe és a vezérlő algoritmusba is implementálható. A modellt a csapattársaimmal valódi kvadrokoptereken is teszteltük, majd az eredményeket összevetettem a szimulációval.

Elkezdttem dolgozni a robotok egyenesbe rendeződését megvalósító modellen is, azonban a dolgozat írásakor ez még nem készült el teljesen. További információk a saját és a csapat munkájával kapcsolatban a dolgozat 9-10. oldalán.

### Megjegyzés:

A dolgozat a 2013 november elején aktuális kutatási eredményeket tartalmazza.

Tartalom lezárva: 2013. november 10.