

**Technická univerzita v Košiciach**  
**Fakulta elektrotechniky a informatiky**

# **Simulácia kooperácie multi-robotického systému**

**Bakalárská práca**

**2021**

**Bohdan Tanasov**

**Technická univerzita v Košiciach**  
**Fakulta elektrotechniky a informatiky**

**Simulácia kooperácie multi-robotického  
systému**

**Bakalárska práca**

Študijný program: Inteligentné systémy  
Študijný odbor: Informatika  
Školiace pracovisko: Katedra umelej inteligencie (KUI)  
Školiteľ: doc. Dr. Ing. Ján Vaščák

**Košice 2021**

**Bohdan Tanasov**

## **Abstrakt v SJ**

Hlavnou výzvou tohto projektu je efektívna a adaptívna koordinácia viacerých autonómnych mobilných robotov s menšími oneskoreniami. Práce možno rozdeliť do troch hlavných častí. Teoretické, v ktorom sa uvažuje o hlavných problémoch komunikácie, riadenia, výmeny informácií v systémoch s niekoľkými mobilnými robotmi. Praktické, ktoré popisuje podstatu projektu na dosiahnutie cieľa niekoľkými robotmi a tiež poskytuje schémy implementácie tohto projektu. A tretia časť popisuje experimenty, ktoré sa uskutočnili na testovanie každého z režimov prenasledovania. Podľa nášho názoru teda navrhovaný model poskytuje účinnú metódu na sledovanie cieľov pre aplikácie, ako sú sledovacie roboty, sledovanie ľudí a budovanie tímu.

## **Klúčové slová**

Navigácia s viacerými robotmi, sledovanie cieľov, správanie pri love, simulácia

## **Abstrakt v AJ**

The main challenge of this project is the efficient and adaptive coordination of multiple autonomous mobile robots with lower delays. The work can be divided into three main parts. Theoretical, in which the main problems of communication, control, information exchange in systems with several mobile robots are considered. Practical, which describes the essences of the project to achieve the goal by several robots, and also provides schemes for the implementation of this project. And the third part describes the experiments that were carried out to test each of the pursuit modes. In our opinion, the proposed model thus provides an effective method for tracking targets for applications such as tracking robots, tracking people, and team building.

## **Klúčové slová v AJ**

Multi-robot navigation, target tracking, hunting behavior, simulation

**TECHNICKÁ UNIVERZITA V KOŠICIACH**  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY  
Katedra kybernetiky a umelej inteligencie

**ZADANIE**  
**BAKALÁRSKEJ PRÁCE**

Študijný odbor: **Informatika**  
Študijný program: **Inteligentné systémy**

Názov práce:

**Simulácia kooperácie multi-robotického systému**

Simulation of Cooperation for a Multi-Robotic System

Študent: **Bohdan Tanasov**  
Školliteľ: **doc. Dr. Ing. Ján Vaščák**  
Školiace pracovisko: **Katedra kybernetiky a umelej inteligencie**  
Konzultant práce:  
Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

1. Spracovať prehľad metód kooperácie v rámci multi-robotických aplikácií.
2. Výber vhodnej metódy robotickej kooperácie a návrh jej realizácie.
3. Softvérovo navrhnuť systém pre kooperáciu robotov.
4. Vykonáť simulačné experimenty a ich vyhodnotiť.
5. Vypracovať dokumentáciu podľa pokynov vedúceho záverečnej práce.

Jazyk, v ktorom sa práca vypracuje: slovenský  
Termín pre odovzdanie práce: 28.05.2021  
Dátum zadania bakalárskej práce: 30.10.2020

.....  
prof. Ing. Liberios Vokárokoš, PhD.  
dekan fakulty



## **Čestné vyhlásenie**

Vyhlasujem, že som bakalársku prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice 28. 5. 2021

.....

*Vlastnoručný podpis*

## **Podakovanie**

Najskôr by som sa chcel podakovať vedúcemu bakalárskej práce doc. Dr. Ing. Jánovi Vaščakovi. Príležitosť komunikovať s docentom Vaščakom bola vždy otvorená, kedykoľvek som narazil na problémové miesto alebo mal otázku o svojom výskume alebo písaní. Dôsledne pripúšťal, aby tato práca bola mojou vlastnou, ale vždy, keď si myslel, že to potrebujem, ma nasmeroval správnym smerom.

Na záver musím svojim rodičom podakovať za to, že mi počas rokov štúdia a pri výskume a písaní tejto práce poskytovali neutíchajúcu podporu a neustále povzbudzovanie. Bez nich by tento úspech nebol možný. Ďakujem.

## **Predhovor**

Účelom tejto práce je podrobne preskúmať niekoľko robotických systémov. Viaceré robotické systémy sú užitočné na vykonávanie úloh, ktoré by jeden robot tažko splnil. V takom prípade je úlohou koordinovať niekoľko robotov v procese prenasledovania. Na prekonanie problémov, ktoré sa vyskytujú pri love, musí robot komunikovať. Roboty sú úplne autonómne a komunikujú bezdrôtovo. Cieľom projektu je demonštrovať užitočnosť koordinovaného správania v iných dôležitých aplikáciách. Získané praktické skúsenosti sa týkajú komunikácie, pridelovania úloh a učenia sa.

# **Obsah**

<b>Úvod</b>	<b>1</b>
<b>1 Formulácia úlohy</b>	<b>2</b>
<b>2 Metódy riadenia v multi-robotických systémoch</b>	<b>3</b>
2.1 Multi-robotické systémy . . . . .	3
2.2 Praktické využitie multi-robotických systémov . . . . .	4
2.3 Ovládanie multi-robotických systémov . . . . .	4
2.3.1 Centralizovaný systém . . . . .	5
2.3.2 Decentralizovaný systém . . . . .	5
2.4 Problématica riadenia . . . . .	5
2.4.1 Deliberatívne riadenie . . . . .	6
2.4.2 Reaktívne riadenie . . . . .	6
2.5 Multi-robotické systémy podľa typovosti . . . . .	8
2.6 Komunikačné štruktúry . . . . .	8
2.6.1 Interakcia prostredím . . . . .	8
2.6.2 Interakcia prostredníctvom snímania . . . . .	9
2.6.3 Interakcia prostredníctvom komunikácie . . . . .	9
2.7 Vybrané typy architektúr . . . . .	9
2.7.1 CEBOT . . . . .	10
2.7.2 ACTRESS . . . . .	10
2.7.3 SWARM . . . . .	10
2.7.4 GOFER . . . . .	11
2.7.5 ALLIANCE . . . . .	11
<b>3 Definícia problému prenasledovania koristi a metódy jeho realizácie</b>	<b>12</b>
3.1 Analýza problému . . . . .	12
3.2 Riadenie koristi . . . . .	13
3.2.1 Popis samotných blokov . . . . .	14

3.3	Režimy navigácie . . . . .	14
3.3.1	Režim detekcie koristi . . . . .	15
3.3.2	Režim prenasledovania a navigácie . . . . .	20
3.3.3	Režim znehybnenia (chytenia) koristi . . . . .	22
3.4	Výber vhodnej metódy robotickej kooperácie . . . . .	23
3.4.1	Centralizované vs. decentralizované . . . . .	23
3.4.2	Systém spätej väzby(Closed-Loop Control System) . . . . .	24
3.4.3	Deliberatívne vs. reaktívne riadenie . . . . .	25
<b>4</b>	<b>Simulačné experimenty</b>	<b>26</b>
4.1	Prehľad a príprava prostredia . . . . .	26
4.2	Ovládanie robota Koristi . . . . .	28
4.3	Detekcia značiek . . . . .	30
4.4	Navigácia pre systém viacerých robotov . . . . .	31
4.4.1	Variácia uhlovej rýchlosťi virtuálnej štruktúry . . . . .	32
4.4.2	Výsledky experimentu: 3 roboty s dosiahnuteľnou virtuálnou štruktúrou . . . . .	34
4.5	Proces prenasledovania . . . . .	36
<b>5</b>	<b>Záver</b>	<b>39</b>

## Zoznam obrázkov

2–1	Stratégie riadenia navigácie v formácií. . . . .	6
3–1	Architektúra systému riadenia robotov. . . . .	14
3–2	Ilustrácia geometrie problému sledovania a navigácie. . . . .	15
3–3	Schéma modelovania obrázkov so značkami ArUco. . . . .	17
3–4	Určenie polohy robota vzhľadom na značku. . . . .	18
3–5	Určenie globálnej polohy robota pomocou polohy značky. . . . .	19
3–6	Režim detekcie koristi - schéma. . . . .	19
3–7	Režim sledovania a navigácií - schéma. . . . .	22
3–8	Režim zatváranie koristi - schéma. . . . .	23
3–9	Hybridná architektúra riadenia. . . . .	24
3–10	Systém spätej väzby. . . . .	25
4–1	Ilustrované cesty, použitie na testovanie. . . . .	29
4–2	Závislosť vplyvu veľkosti oblasti značky na chybe pri zistovaní jej polohy. . . . .	30
4–3	Pravdepodobnosť detekcie značky v závislosti od jej oblasti na obrázku pre 3 rôzne uhly naklonenia: $0^\circ$ , $45^\circ$ , $70^\circ$ . . . . .	31
4–4	Nie sú vynútené nežiaduce kmity dráhy robota prechodnej fázy. . .	32
4–5	Nežiaduce oscilácie dráhy robota, ak nie sú splnené predpokladané obmedzenia dynamického ciela (ak $\theta_T > P$ ). . . . .	33
4–6	Trajektória robotov. . . . .	34
4–7	Zmena vzdialenosťi $d_{S_i}$ medzi robotom i a zvoleným cielom (i = 1..3). . . . .	35
4–8	Výsledky simulácie . . . . .	37

## **Zoznam tabuliek**

4 – 1 Zoznam časovaní robota na pokrytie rôznych typov tratí . . . . .	28
4 – 2 Výsledkom simulácie na ploche 100m * 100m vs 60m * 60m . . . . .	37
4 – 3 Výsledky simulácie 3 prenasledovateľov v oblasti 100 m * 100 m . . . .	38

## Slovník termínov

**Virtuálna štruktúra** je štruktúra tvorená a udržiavaná skupinou robotov, ktorí prenasledujú cieľ

**$R_p$**  je zápis na diagramoch a obrázkoch robota Korist

**$R_1, R_2, R_3$**  je záznam o schémach a obrázkoch robotov zo skupiny prenasledovateľov

**ROS** Robot Operating System.

**Robot Korist** je robot, ktorým ovláda človek pomocou klávesnice a jeho cieľom je utekať pred skupinou prenasledovateľov.

**Robot Prenasledovateľ** je hlavný robot zo skupiny prenasledovateľov.

**r** konštantná minimálna vzdialenosť, na ktorú sa robot Prenasledovateľ nesmie priblížiť ku robotovi Korist.

**d(P,A)** je vzdialenosť od robota prenasledovateľa k robotovi koristi v konkrétnom čase.

**R** je určitá konštantná bezpečná vzdialenosť od robota Korist po robota Prenasledovateľ, ktorú sa skupina prenasledovateľov snaží udržať v sledovacom režime

**$l_x$**  je vzdialenosť.

## Úvod

Počas posledného desaťročia zaznamenala kooperačná robotika dôležitý vývoj. Systémy s viacerými robotmi sa používajú na rôzne aplikácie na vykonávanie úloh, ktoré sú pre jedného robota náročné alebo časovo náročné. V literatúre sa uvažuje o rôznych aspektoch spolupráce v robotike. Boli navrhnuté rôzne prístupy od Lyapunovovej teórie po teóriu štrukturálnej zložitosti. Všeobecne sa diskutuje o riadení formovania robota za prítomnosti a neprítomnosti prekážok. Literatúra o formovaní robotov je rozsiahla a navrhujú sa riešenia založené na behaviorálnych a modelových kontrolách. Metódy založené na správaní zvyšujú odozvu v reálnom čase a znižujú zložitosť, ktorá sprevádza presné modely. Tieto metódy však nepodporujú prostriedky na predpovedanie presných výsledkov robotov. Tieto metódy sú známe svojou robustnosťou. Sledovanie pohybujúceho sa ciela pomocou mobilného robota bolo zvážené mnohými výskumníkmi. Nedávno boli na splnenie tej istej úlohy zavedené systémy s viacerými robotmi. Na tento problém sa pozérame z rôznych uhlov pohľadu a riešime ho pomocou rôznych metód a techník. V tejto práci bolo navrhnuté sledovanie pohybujúceho sa objektu pomocou kamery na lovcom robotovi a značkovača na koristi. Pomocou techniky, ktorá je v tejto práci napísaná, je možné získať vektor na korist pre každého robota. Ďalším problémom v tejto časti bola záchrana formácie v procese prenasledovania. Roboty sa pohybujú vo formáciach riadených prostredníctvom formačných vektorov. A tiež problémom bolo kooperatívne zachytenie ciela viacerými robotmi, metódy realizácie boli v tejto práci napísané. V poslednej časti boli urobené rôzne skúsenosti a podané výsledky.

## 1 Formulácia úlohy

**Spracovať prehľad metód kooperácie v rámci multi-robotických aplikácií.** Táto časť bakalárskej práce obsahuje popis úloh, ktoré sa riešia pri vzájomnej interakcii viacerých robotov. Táto kapitola obsahuje prehľad robotov, kontrolné metódy, komunikačné metódy a problémy, ktoré sa vyskytujú pri vývoji algoritmov pre multi-robotický systém.

**Výber vhodnej metódy robotickej kooperácie a návrh jej realizácie.** Táto časť popisuje proces výberu najvhodnejšej metódy na implementáciu multi-robotického projektu na základe výhod a nevýhod rôznych metód uvedených v predchádzajúcej časti.

**Softvérovo navrhnutie systému pre kooperáciu robotov.** Táto časť bude obsahovať prehľad softvéru potrebného na vypracovanie bakalárskeho projektu a tiež popis metód a prístupov použitých pri práci na projekte.

**Vykonanie simulačných experimentov a ich výhodnotenie.** Táto kapitola predstavuje výsledky a výhodnotenie simulačných experimentov vykonalých pre rôzne situácie. Každá simulácia má svoj vlastný popis a ku každej je pridaný záver

**Vypracovať dokumentáciu podľa pokynov vedúceho záverečnej práce.** Posledný bod bude súvisieť s prípravou dokumentácie a popisom celého procesu vývoja tohto projektu.

## 2 Metódy riadenia v multi-robotických systémoch

V organizácii a správe systémov s niekoľkými mobilnými robotmi v súčasnosti existuje veľa rôznych prístupov. Nedá sa povedať, že existujú správne alebo nesprávne prístupy, každý zo systémov má svoje výhody aj nevýhody. Rozhodnutie o použití systému sa prijíma na základe konkrétnej úlohy projektu. V tejto časti sa pozrieme na rôzne metódy riadenia a organizácie multi-robotických systémov, ako aj na ich výhody a nevýhody.

### 2.1 Multi-robotické systémy

Systémy viacerých robotov sú dobré z mnohých ďalších dôvodov. Podľa Aparicio a Lima [1] systémy s viacerými robotmi ponúkajú robustnosť a prispôsobivosť, aké sa v systémoch s jedným robotom nenachádzajú. Ak je jeden agent v systéme poškodený alebo nefunguje správne, je možné úlohu dokončiť so zvyšnými agentmi. Túto myšlienku možno uplatniť aj pri údržbe. Agenti môžu byť obsluhovaní po jednom, čo vedie k skutočnosti, že systém nie je nečinný, ak sú ostatní roboti schopní pokračovať v plnení úlohy bez prítomnosti ich partnera. Viaceré roboty navyše dokážu pokryť veľkú oblasť a môžu sa špecializovať na menšie úlohy, ktoré postupujú k splneniu väčšej úlohy [1]. Preto je pravdepodobnejšie, že systémy s viacerými robotmi budú spoloahlivo vykonávať veľké a zložité úlohy. Napriek mnohým výhodám systémov viacerých robotov je potrebné vyriešiť niekoľko nevýhod. Najskôr je komunikácia medzi agentmi v systéme výpočtovo zložitá. Podľa Coesa, Nurbakhsh a Sikar , keď sa systémy s viacerými robotmi rozhodnú, musia brať do úvahy čas potrebný na cestu na konkrétné miesto, čas čakania na príchod ďalších robotov a čas na dokončenie úlohy. V dôsledku toho sú riadiace algoritmy veľmi zložité, pretože systém sa implementuje tažšie, ak na konkrétnom probléme pracuje viac agentov. Rádiová komunikácia dodáva systému ďalšiu vrstvu zložitosti, pretože si vyžaduje použitie prenosových obvodov a komunikačných protokolov. Preto, aj keď môže byť systém s viacerými robotmi efektívnejší, tento však vyžaduje väčšiu zložitosť než

systém s jedným robotom.

## 2.2 Praktické využitie multi-robotických systémov

Počas výskumu boli objavené aplikácie, ktoré priamo súvisia s mojím projektom. V jednom konkrétnom príklade boli na premiestnenie nábytku na konkrétné miesta použité dva veľké roboty. V praktickejšom príklade je možné použiť roboty na presun ťažkých materiálov na stavenisku. Je pravdepodobné, že roboty s takouto úlohou veľmi pomôžu a môžu zvýšiť efektivitu, ale roboty sú zvyčajne príliš drahé na to, aby boli finančné prospešné.

Jednou z možností v súčasnosti je použitie koordinovaných robotov na skúmanie oblastí, ktoré by mohli byť pre človeka nebezpečné. Tieto roboty môžu autonómne zvyšovať bezpečnosť alebo vytvárať nebezpečné oblasti. Napríklad oblasť plná mín môže byť vhodná pre robotov vybavených senzormi určenými na detekciu mín. Potom, čo v mínovom poli, bude robot schopný rozpoznať mínu bez účasti ľudí. nájdení bane môže robot označiť polohu míny ďalším robotom. Ostatní roboti pomocou koordinačných algoritmov a pozičných senzorov budú môcť k míne pristúpiť a pomocou ju rozoberať alebo označiť. Roboti sa môžu navzájom rozprávať alebo môžu používať hlavného robota, ktorý dáva každému z nich smer [2]. Projekt, ktorý je rovnako jednoduchý ako koordinované tlačenie škatúľ robotom, slúži ako východiskový bod pre zložitejší a užitočnejší systém, ako sú napríklad robotické vyhľadávače mín.

## 2.3 Ovládanie multi-robotických systémov

Ovládanie systému s viacerými robotmi je vo všeobecnosti náročným problémom. K tejto otázke existujú dva prístupy: centralizované riadenie a decentralizované riadenie [3].

### 2.3.1 Centralizovaný systém

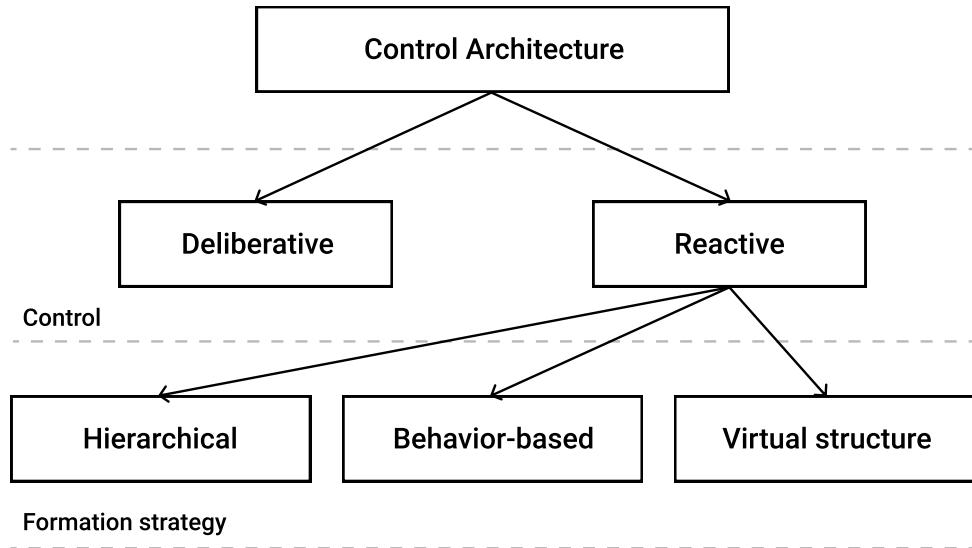
V centralizovanom systéme s niekoľkými robotmi sa ukladajú globálne informácie o stave celého systému. Systém zhromažduje informácie o všetkých robotoch a sleduje ich polohu v prostredí. Na základe informácií získaných od robotov dokáže zostaviť mapu. Tento systém je buď na stacionárnom hostiteľovi, alebo v jednom robote, ktorý zjavne funguje ako hlavný. Potom majster zorganizuje tím robotov, aby dosiahli spoločný cieľ a naplánuje úlohy pre jednotlivých členov tímu a dohliada na celý proces. Táto architektúra sa dá ľahko navrhnúť, ale nie je odolná voči výpadkom komunikácie a nepredvídateľným situáciám. Centralizované riadenie je zvyčajne vhodné pre obmedzený počet robotov, ktorí pracujú v známom a nemennom prostredí.

### 2.3.2 Decentralizovaný systém

Naproti tomu decentralizované systémy nezahŕňajú žiadneho riaditeľa, ktorý má úplné informácie o stave systému a riadi celý proces. Naopak, každý robot je autonómna jednotka, ktorá koná v súlade so stavom svojho prostredia. Robot si je samozrejme vedomý prítomnosti ďalších robotov a môže s nimi komunikovať na miestnej úrovni. Komplexné skupinové správanie vyplýva z interakcií medzi robotmi a prostredím. Táto architektúra je veľmi robustná, môže dobre fungovať v nepriateľskom prostredí a je škálovateľná, potenciálne obrovské množstvo homogénnych robotov môžu spolupracovať na dosiahnutí spoločného cieľa.

## 2.4 Problématika riadenia

Problém s riadením v navigácii mobilných robotov sa rieši dvoma spôsobmi: deliberačným a reaktívny riadením (Obrázok 2 – 1).



Obrázok 2 – 1 Stratégie riadenia navigácie v formácií.

#### 2.4.1 Deliberatívne riadenie

Prístup založený na plánovaní pohybu a trajektórie pohybu vyžaduje predbežné znalosti prostredia na plánovanie pohybov robotov [4]. Tento prístup využíva formalizmy ako Voronoiove diagramy alebo umelé potenciálne funkcie [5], pričom zohľadňuje všeobecné poznatky o životnom prostredí. Plánovanie pohybu je všeobecne optimálne z hľadiska efektívnosti / reaktivity. V skutočnosti sa pre mierku veľkého počtu robotov nemodifikujú dobre kvôli výpočtovej zložitosti. Vďaka predchádzajúcim znalostiam prostredia však roboti svoju misiu zvyčajne úspešne dokončia dobrým výkonom.

#### 2.4.2 Reaktívne riadenie

Pri reaktívnej metóde, roboty konajú iba podľa informácií svojich miestnych senzorov bez akýchkoľvek ďalších všeobecných znalostí. Techniky založené na správaní sú vynikajúcou ilustráciou reaktívnej kontroly. Globálna úloha robota je v skutočnosti rozdelená na súbor čiastkových úloh (vzorce správania). Podľa informácií o senzore je stratégia riadenia použitá na robota založená na jednom zvolenom vzorci sprá-

vania alebo je zlúčením niekoľkých vážených modelov. Keď aplikácia vyžaduje, aby roboty fungovali v reálnom čase (napríklad v nebezpečných prostrediach), je zrejmé, že reaktívne metódy sa stávajú oveľa zaujímavejšimi ako plánovanie pohybu [4].

**Hierarchická stratégia.** Pri prvom prístupe sa jeden alebo viac robotov považuje za vedúcich a iné roboti sú ďalší. Vedúci typicky sleduje danú trajektóriu, zatiaľ čo nasledovníci sledujú jej transformované súradnice. Tento prístup sa dá ľahko sledovať. Je však zaznamenané, že problém s hlavným robotom spôsobí zastavenie celého systému. V prístupe distribuovaného správania, neexistuje medzi robotmi hierarchia. Každý má svoje vlastné vnímanie a kontrolu a porucha robota nevedie k zlyhaniu skupiny [4].

**Stratégia správania.** Stratégia založená na správaní znamená, že každý robot má súbor správania (základné úlohy), ktoré musia byť vykonané. Výsledné skupinové správanie vyplýva zo základnej lokálnej interakcie bez explicitného vzoru celkového kooperatívneho správania. Tento prístup však bol kritizovaný za to, ako volí riadenie pre každého robota. Podľa informácií o vnímaní riadiaci systém v skutočnosti prepína medzi správaním (napríklad konkurenčný prístup) alebo kombinuje niekoľko ovládačov (napríklad motorický obvod). To prirodzene stázuje štúdium udržateľnosti stratégie globálneho riadenia [6].

**Stratégia virtuálnej štruktúry.** Virtuálna štruktúra (tretí prístup) považuje vzdelenie za jediné virtuálne telo. Tvar posledného menovaného je požadovaný tvar formácie a jeho pohyb sa prevedie na požadovaný pohyb každého vozidla. Virtuálna štruktúra bola implementovaná v niekoľkých dielach s využitím potenciálnych metód poľa: teda všetky prvky formácie sledujú pridelené uzly, ktoré prechádzajú do požadovanej konfigurácie. Na rozdiel od plánovania pohybu využívajú potenciálne funkcie použité pri prístupe k virtuálnej štruktúre iba okamžité a lokálne vnímanie robotov. Nedostatočné využitie potenciálnych prvkov pre tento druhý prístup zodpovedá zvyšujúcej sa zložitosti riadenia tvaru flotily v dynamickom prostredí. To v

skutočnosti znamená, že robot je vystavený často sa meniacemu počtu / amplitúde síl, čo vedie k ďalším lokálnym minimám, fluktuáciám atď. Preto je v tomto prípade veľmi tažké preukázať spoľahlivosť a stabilitu navigácie [6], [4].

## 2.5 Multi-robotické systémy podľa typovosti

Skupinu robotov definujeme ako homogénnu, ak sú možnosti jednotlivých robotov rovnaké a inak skupina je heterogénnia. Heterogenita vo všeobecnosti prináša zložitosť, pretože alokácia úloh sa stáva zložitejšou a agenti majú väčšiu potrebu modelovať ďalších robotov v skupine. Existuje koncept pokrytie úlohy, ktorý meria schopnosť daného člena tímu dosiahnuť danú úlohu. Tento parameter predstavuje index dopytu po spolupráci: keď je pokrytie úloh vysoké, úlohy je možné splniť bez väčšej spolupráce, ale inak je spolupráca nevyhnutná. Pokrytie úlohy je maximálne v homogénnych skupinách a klesá, keď sú skupiny čoraz viac heterogénne (t. j. v krajinom prípade môže danú úlohu vykonať iba jeden agent v skupine). V literatúre v súčasnosti prevládajú diela, ktoré predpokladajú homogénne skupiny robotov. Niektoré pozoruhodné architektúry však dokážu zvládnuť heterogenitu, napr. ACT-RESS a ALLIANCE. V heterogénnych skupinách môže byť alokácia úloh určená individuálnymi schopnosťami, ale v homogénnych systémoch môže byť potrebné, aby sa agenti diferencovali do rôznych rolí, ktoré sú vždy známe v čase návrhu alebo dynamicky vznikajú za behu [4].

## 2.6 Komunikačné štruktúry

Komunikačná štruktúra skupiny určuje možné spôsoby interakcie. Charakterizujeme tri základné typy interakcií, ktoré môžu byť podporované.

### 2.6.1 Interakcia prostredím

Najjednoduchší typ interakcie sa vyskytuje, keď je samotné prostredie komunikačné médiu (v skutočnosti, zdieľanej pamäti) a medzi agentmi nie je explicitná komu-

nikácia alebo interakcia. Niektorí výskumníci tiež nazývali túto modalitu s "spolu-prácou bez komunikácie"[7].

### 2.6.2 Interakcia prostredníctvom snímania

Interakcia prostredníctvom snímania patrí do miestnych interakcií, ktoré sa vyskytujú medzi agentami v dôsledku toho, že agenti sa navzájom cítia, ale bez explicitnej komunikácie. Tento typ interakcie si vyžaduje schopnosť agentov rozlišovať medzi inými činiteľmi v skupine a ďalších objektoch v prostredí, ktoré sa v niektorých literárnych zdrojoch nazýva "súvisiace uznanie"[8]. Interakcia prostredníctvom sondovania je potrebná na simuláciu iných agentov. Vzhľadom na obmedzenia hardvéru je interakcia cez snímanie často emulovaná pomocou rádiovej alebo infračervenej komunikácie.

### 2.6.3 Interakcia prostredníctvom komunikácie

Tretia forma interakcie spočíva v explicitnej komunikácii s inými agentmi budť prostredníctvom zmenených, alebo vysielaných zámerných správ (príjemca správy môže byť známy alebo neznámy). Pretože architektúry, ktoré umožňujú túto formu komunikácie, sú podobné komunikačným sietiam, vzniká veľa štandardných problémov z oblasti sietí, vrátane návrhu sietových topológií a komunikačných protokolov. Napríklad na komunikáciu medzi robotmi môže sa použiť prístupový protokol k médiám (podobný protokolu Ethernet). Aj som stretol komunikáciu pomocou protokolu „hello-call“, pomocou ktorého ustanovujú „retázce“ s cieľom rozšíriť svoje efektívne komunikačné rozsahy a bezdrôtovým protokolom CSMA/CD (Carrier Sense Multiple Access with Collision Detection) pre distribuované robotické systémy [9].

## 2.7 Vybrané typy architektúr

Všetky systémy implementujú určitú skupinovú architektúru. Teraz popíšeme niekoľko obzvlášť dobre definovaných reprezentatívnych architektúr spolu s prácamí

vykonanými v každom z ich rámcov. Je zaujímavé poznamenať, že tieto architektúry zahŕňajú celé spektrum od tradičnej AI po vysoko decentralizované prístupy.

### 2.7.1 CEBOT

CEBOT (CELLular roBOTics System) je decentralizovaná, hierarchická architektúra inšpirovaná bunkovou organizáciou biologických entít. Systém sa konfiguruje dynamicky v tých základných autonómnych „bunkách“ (robotoch), ktoré je možné fyzicky spojiť s inými bunkami, a dynamicky konfigurujú svoju štruktúru na „optimálnu“ konfiguráciu v reakcii na meniace sa prostredia. V hierarchii CEBOT existujú „hlavné bunky“, ktoré koordinujú čiastkové úlohy a komunikujú s ostatnými hlavnými bunkami [10] [11].

### 2.7.2 ACTRESS

V systéme ACTRESS (robot a syntetický systém na báze ACTor) tvoria „roboti“ vrátane 3 robotov a 3 pracovných staníc (jeden ako rozhranie k ľudskému operátorovi, jeden ako obrazový procesor a druhý ako globálny manažér prostredia) heterogénnu skupinu, ktorá sa snaží vykonávať úlohy, ako je tlačenie objektov, ktoré nemôže vykonat žiadny z robotov sám. Komunikačné protokoly na rôznych úrovniach abstrakcie poskytujú prostriedky, na ktorých sú postavené mechanizmy „skupinového obsadenia“ a rokovacie mechanizmy založené na viacstupňových protokoloch vyjednávania. Študujú sa rôzne problémy, ako napríklad efektívna komunikácia medzi robotmi a manažérmi prostredia, predchádzanie kolíziám [11].

### 2.7.3 SWARM

SWARM je distribuovaný systém s veľkým počtom autonómnych robotov. (Upozorňujeme, že práca na systémoch SWARM sa začala ako práca na bunkových robotických systémoch, kde mnoho jednoduchých agentov obsadzovalo jedno- alebo dvojrozumné prostredie a boli schopní vykonávať úlohy, ako je generácia vzorov a samo-

organizácia). Inteligencia SWARM je „vlastnosťou systémov neinteligentných robotov vykazujúcich kolektívne inteligentné správanie“. Samoorganizácia v SWARM-e je schopnosť distribuovať sa „optimálne“ pre danú úlohu, napríklad prostredníctvom formovania geometrických vzorov alebo štruktúrnej organizácie. SWARM vykazuje distribuovanú architektúru, zvyčajne bez rozdielov medzi členmi. Interakcia prebieha tak, že každá bunka reaguje na stav [11].

#### **2.7.4 GOFER**

Architektúra GOFER bola použitá na štúdium distribuovaného riešenia problémov viacerými mobilnými robotmi vo vnútornom prostredí s využitím tradičných techník AI. V systéme GOFER centrálny systém plánovania a plánovania úloh (CTPS) komunikuje so všetkými robotmi a má globálny prehľad o úlohách, ktoré sa majú vykonať, aj o dostupnosti robotov na vykonávanie úloh. CTPS generuje štruktúru plánu (šablóna pre inštanciu plánu) a informuje všetkých dostupných robotov o očakávaných cieľoch a štruktúrach plánu. Roboty používajú na určenie svojich rolí algoritmus pridelenia úloh [12].

#### **2.7.5 ALLIANCE**

Architektúru ALLIANCE vyvinul Parker [13] s cieľom študovať spoluprácu v heterogénnom, malom až strednom tíme prevažne nezávislých a volne prepojených robotov. Roboty sa považujú za schopné s určitou pravdepodobnosťou vycítiť účinky ich vlastných činov a činov iných agentov prostredníctvom vnímania a explicitnej rozhlasovej komunikácie. Jednotliví roboti sú založené na ovládači založenom na správaní s rozšírením pre aktiváciu „skupín správania“, ktoré plnia určité úlohy. Tieto súbory sú aktivované motivačným správaním, ktorého aktivity sú určené robotovou informovanosťou ich spoluhráčov [11] [4].

### 3 Definícia problému prenasledovania koristi a metódy jeho realizácie

V tejto časti definujeme problém, hovoríme o spôsoboch jeho realizácie a implementujeme jednu z možných variácií pri riešení problému prenasledovania koristi pomocou multi-robotického systému.

#### 3.1 Analýza problému

V tejto bakalárskej práci sa zaoberá problém prenasledovania pomocou správania viacerých robotov. Cielom je implementovať riadiace stratégie, ktoré robotom umožňujú dosiahnuť a uzavrieť nepredvídateľne sa pohybujúcu korist. Existuje množstvo metód na riešenie problému kooperatívneho prenasledovania založeného na viacerých robotoch. Takéto konvenčné metódy sa delia do dvoch kategórií, ako sú metódy založené na polohe a metódy založené senzoroch.

Lokalizovaným spôsobom je cieľové miesto známe vopred, v ktorom sa na prenasledovania cieľov používajú techniky spolupráce založené na umelej inteligencii. Zatiaľ čo v metódach založených na senzoroch väčšina výskumníkov používa tradičnú teóriu riadenia na vykonávanie úlohy zamerania na cieľ v neznámom prostredí pomocou prichádzajúcich informácií zo senzora. Rozhodol som sa použiť druhú metódu založenú na údajoch senzorov, pretože techniky založené na polohe závisia iba od informácií o polohe cieľov, čo v prostredí v reálnom čase, kde sa polohy často menia, prakticky nie je možné.

Korist môže vykonávať tri druhy pohybu:

1. Neakceleračný pohyb: korist sa pohybuje s konštantným orientačným uhlom a rýchlosťou.
2. Zrýchľujúci pohyb: orientačný uhol koristi sa časovo mení.

3. Inteligentný pohyb: orientačný uhol koristi sa časovo mení, ale inteligentne, aby sa dalo uniknúť pred predátormi.

Inteligentné koristi si vyžadujú prepracovanejšie stratégie. Máme nasledujúce predpoklady

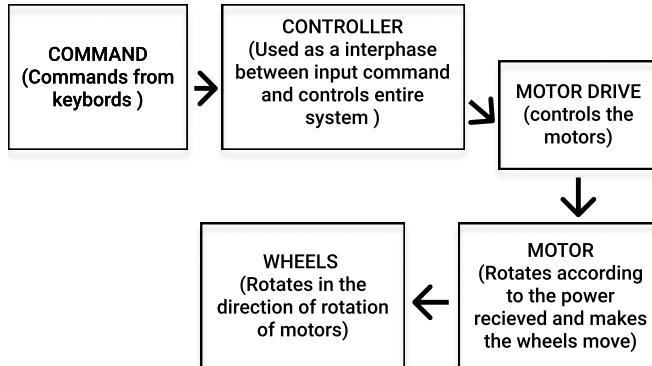
1. Každý robot má senzorický systém, ktorý umožňuje zisťovať prekážky a odhadnúť polohu koristi podľa robota.
2. Rýchlosťi robotov vyhovujú  $v_i > v_p$ , pre  $i = 1, \dots, N$ . Kde  $v_i$  - rýchlosť lovca a  $v_p$  - rýchlosť koristi.
3. Roboti a korist sa pohybujú konštantnou rýchlosťou.
4. Cesta, ktorú korist prešla, je spojité.

Realizáciu projektu možno rozdeliť na dve prakticky nezávislé časti:

- Režim riadenia koristi - pohyb koristi ovládaný človekom pomocou manipulátora.
- Režim navigácie - organizácia pohybu, sledovania a prenasledovania koristi v neznámom prostredí skupinou robotov, ktorí navzájom komunikujú.

### 3.2 Riadenie koristi

Architektúra systému sa skladá z troch hlavných blokov: príkazový blok, riadiaci blok a blok mobility. Príkazový blok sa skladá z klávesnice, blok mobility pozostáva z motorov, vodičov motorov a kolies. Pri všetkých metódach ovládania používa programovací jazyk Python na odosielanie príkazov z mikrokontroléra do ovládača motora.



Obrázok 3 – 1 Architektúra systému riadenia robotov.

### 3.2.1 Popis samotných blokov

**Príkazový blok.** Príkazový blok je súčasťou ovládacieho rozhrania, ktoré umožňuje používateľovi komunikovať so zariadením pomocou tlačidiel a odosielat nimi požadovaný príkaz. Tento ovládač sa skladá z piatich tlačidiel pre päť príkazov: hore, dole, doľava, doprava a stop. Tlačidlá hore a dole sa používajú na príkaz robota, aby sa pohyboval dopredu a dozadu. Lavé tlačidlo slúži na ovládanie robota pri odbočovaní doľava a pravé tlačidlo pri odbočovaní doprava. Na zastavenie robota je možné použiť tlačidlo Stop.

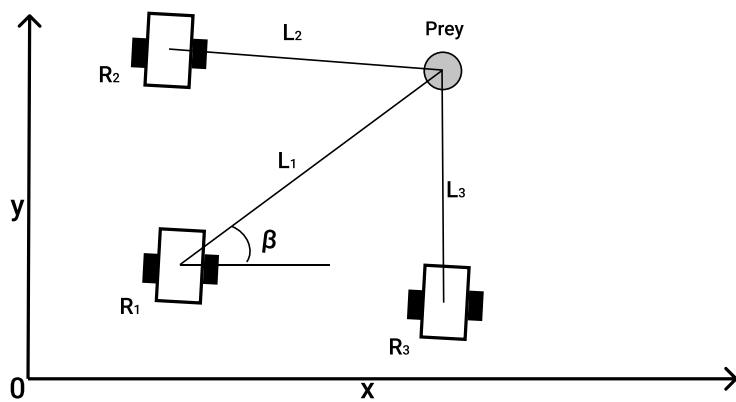
**Blok radiča.** Blok radiča je prepojený s rozhraním, dekóduje príkazy a vydáva riadiace signály do príslušných modulov, aby sa robot mohol pohybovať. Udržuje tiež časovanie a postupnosť generovania riadiacich signálov. Prijíma rozhodnutia na základe príkazov na riadenie pohybu robota v libovoľnom smere.

**Blok mobility.** Blok mobility sa skladá z motorov a kolies. Tento blok prijíma príkazy z ovládača a pomocou rovníc diferenčného pohonu ich transformuje na hodnoty pre každý z motorov a priraduje tieto hodnoty, ktoré umožňujú pohyb robota.

## 3.3 Režimy navigácie

Úloha prenasledovania koristi je rozdelená na časti. Každý robot sa pohybuje v troch naviacačných režimoch nasledovne

1. Režim detekcie koristi.
2. Režim prenasledovania a navigácie: Počas tohto režimu je robot riadený na sledovanie koristi.
3. Režim znehybnenia(chytenia) koristi: Toto je konečný režim, v ktorom roboty vytvárajú figúrku, ktorá korist uzavrie.



**Obrázok 3 – 2** Ilustrácia geometrie problému sledovania a navigácie.

Tieto režimy sú v ďalších častiach opísané osobitne.

### 3.3.1 Režim detekcie koristi

Pretože mobilný robot by mal vykonávať činnosť, je potrebné, aby robot vedel, aká je jeho poloha. Preto musí mať implementovaný nejaký spôsob lokalizácie. Tieto metódy závisia od typu robota, od typu senzorov robota, od typu súradnicového systému alebo od typu prostredia, v ktorom robot pracuje. V oblasti lokalizácie mobilných robotov sa osvedčujú trigonometrické metódy, inerciálne navigačné metódy alebo matematicky komplikovaná pravdepodobnosťná metóda.

V súčasnosti má takmer každý novo vyvinutý mobilný robot zabudovaný kamerový systém. Preto je vhodné ho použiť na určenie polohy robota v prostredí. Vizuálna lokalizácia patrí do skupiny trigonometrických metód. Vizuálny systém môže obsahovať jednu alebo dve kamery a mal by zisťovať niektoré značky v prostredí pomo-

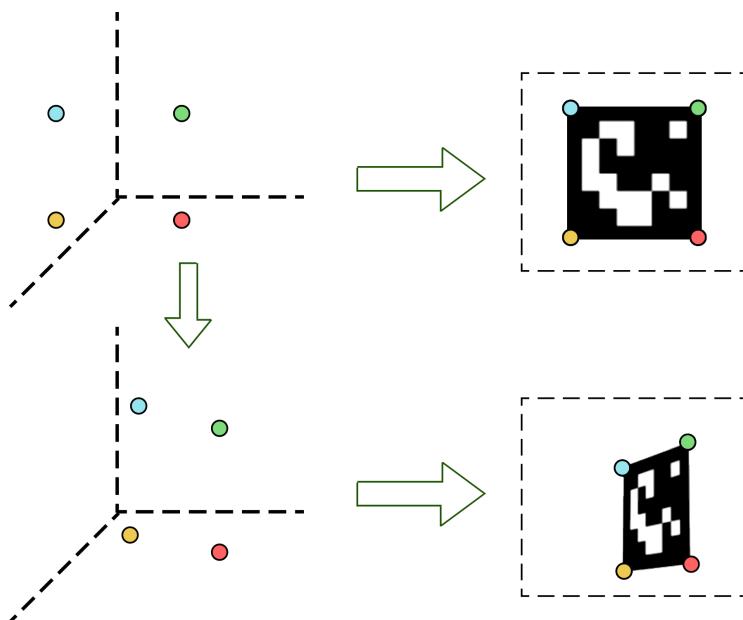
cou nástrojov počítačového videnia. Tieto značky môžu byť umelé alebo prirozené (vrátane charakteristických vlastností prostredia, ktoré je možné získať pomocou rôznych algoritmov). Ak vizuálny systém pozostáva iba z jednej kamery, rovnako ako v našom prípade, poloha robota vo vzťahu k značke sa počíta z dvoch snímok prostredia, v ktorom je počas pohybu robota jedna značka na rôznych miestach obrazu.

Ďalej je potrebné zabezpečiť, aby vzor kódujúci jedinečný identifikátor značky nemohol byť vytvorený rotáciou iného vzoru používaneho v prostredí. Tieto požiadavky sú splnené v systéme značiek zvaných ArUco, ktorý je založený na AR-Tag a je dodávaný s knižnicou funkcií na detekciu a lokalizáciu značiek, ktorú sme používali na zisťovanie a identifikáciu robotickej koristi v prostredí.

Simulačný algoritmus pozostáva z nasledujúcich postupných krokov (obrázok 3-3):

1. Priradenie interných a externých parametrov kamery (veľkosť snímača v pixeloch a milimetroch, ohnisková vzdialenosť objektívu, koeficienty skreslenia, umiestnenie kamery).
2. Priradenie dvojrozmerných súradníc rohov známky v súlade s jej pôvodným obrázkom.
3. Priradenie trojrozmerných súradníc rohov známky v súlade s jej špecifikovanými fyzickými rozmermi.
4. Priradenie vymodelovanej 3D polohy značky pomocou vektora posunutia a rotácie.
5. Výpočet nových trojrozmerných súradníc rohov značky vynásobením počiatocných súradníc rotačnej a translačnej matice.
6. Projekcia trojrozmerných bodov rohov značky do dvojrozmerných bodov na modelovanom obrázku pomocou parametrov kamery.

7. Výpočet funkcie transformácie perspektívy z dvojrozmerných súradníc pôvodu na projektované rohové súradnice.
8. Vykreslenie obrazu známky v nových dvojrozmerných súradniach na upravenom obrázku pomocou perspektívnej transformácie.
9. Odhad polohy 3D značiek pomocou detektora značiek na modelovanom obrázku.
10. Vyhodnotenie chyby porovnaním odhadovanej polohy značky s vopred určenou polohou v kroku 4.
11. Opakovanie krovok 4 až 10 pre ďalšie polohy značiek.



**Obrázok 3 – 3** Schéma modelovania obrázkov so značkami ArUco.

Interný binárny kód značky je definovaný piatimi slovami, každé z nich obsahuje päť bitov. Na zakódovanie identifikátora sa používa mierna úprava Hammingovho kódu. Preto sú z piatich bitov slova iba dva nosiče užitočných informácií. Počet jedinečných značiek sa teda rovná 1024 [14]. Po úspešnej detekcii značky knižnice

ArUco poskytujú informácie o polohe vzhľadom na optický stred kamery. Pre rotáciu poskytuje systém ArUco štandardnú transformačnú maticu alebo štvorec. Na účely lokalizácie musia byť tieto údaje transformované do Eulerových uhlov. Z hľadiska umiestnenia mobilných robotov je potrebné určiť rotáciu okolo osi z:

$$\theta = \arctan [2.(X.Y - W.Z), W^2 - X^2 - Y^2 + Z^2] \quad (3.1)$$

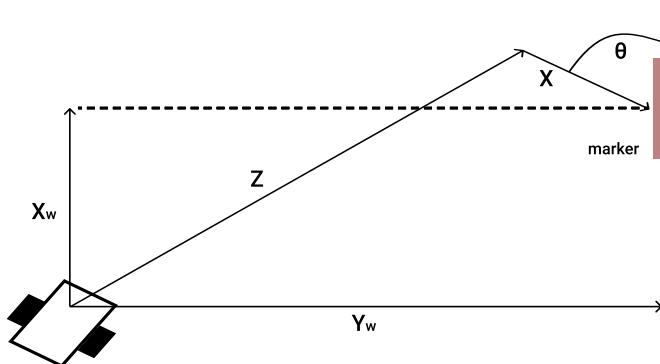
Kde  $W$ ,  $X$ ,  $Y$  a  $Z$  sú prvky popisujúce štvorček poskytovaný funkciami knižnice ArUco. Poloha robota voči značke je daná:

$$x_w = x.\cos(\theta) - z.\sin(\theta); \quad (3.2)$$

$$y_w = z.\cos(\theta) - x.\sin(\theta); \quad (3.3)$$

$$\theta_w = \theta \quad (3.4)$$

Kde  $x$  je poloha značky vzhľadom na optické stretnutie fotoaparátu pozdĺž osi x,  $z$  je vzdialenosť medzi fotoaparátom a rovinou kolmou na optickú os fotoaparátu a je pretínajúc stredom značky a  $\theta$  je uhol medzi fotoaparátom a značkou (obrázok 3-4).



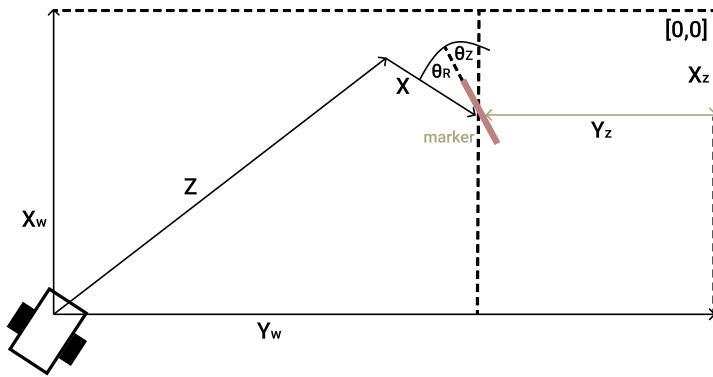
**Obrázok 3 – 4** Určenie polohy robota vzhľadom na značku.

Globálna poloha robota  $x_w$ ,  $y_w$  a  $\theta_w$  podľa obrázku 3-5 je potom:

$$x_w = x.\cos(\theta_R + \theta_Z) - z.\sin(\theta_R + \theta_Z) + x_Z; \quad (3.5)$$

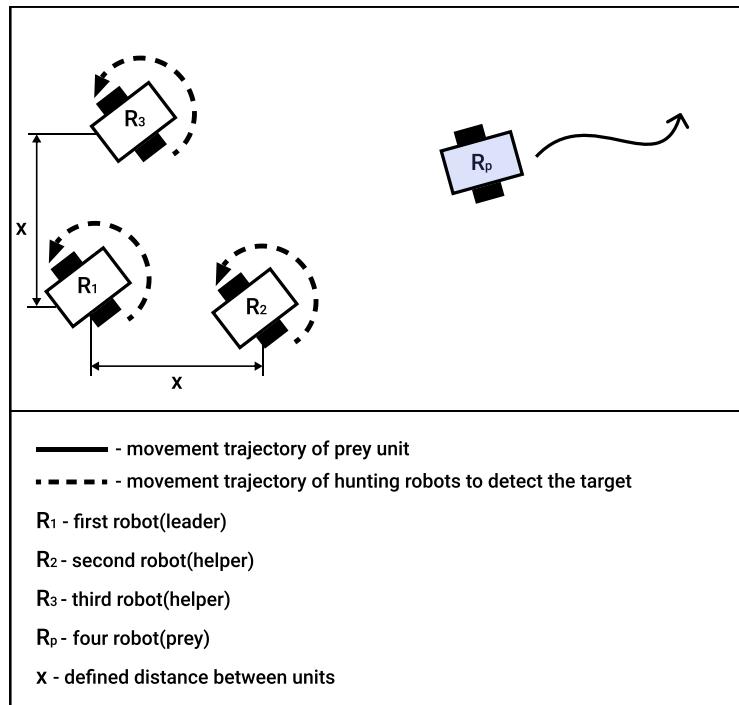
$$y_w = z.\cos(\theta_R + \theta_Z) - x.\sin(\theta_R + \theta_Z) + y_Z; \quad (3.6)$$

$$\theta_w = \theta_R + \theta_Z \quad (3.7)$$



**Obrázok 3 – 5** Určenie globálnej polohy robota pomocou polohy značky.

Pomocou údajov získaných výpočtom polohy robota vo vzťahu ku koristi - vykonajú roboty detekciu a sledovanie koristi.



**Obrázok 3 – 6** Režim detekcie koristi - schéma.

### 3.3.2 Režim prenasledovania a navigácie

Tento režim je pravdepodobne najdôležitejším režimom v rámci problému s lovom. Roboti začnú korisť sledovať zo svojich počiatočných pozícií. Výpočet relatívnej vzdialenosť od cieli a uhla vychýlenia sa vykonáva pomocou systému značiek AruCo, ktorý je podrobne popísaný v predchádzajúcej časti.

Pretože sa prenasledovateľ bude pohybovať v smere uvedenom v oddiele 3.3.1, je potrebné, aby prenasledovateľ upravil svoju rýchlosť iba v závislosti od svojej relatívnej polohy k cielu. Aby sme zaručili, že prenasledovateľ nenarazí na korisť, zavedieme bezpečnú vzdialenosť  $r$  ( $0 < r < R$ ), takže keď je vzdialenosť medzi prenasledovateľom a koristou menšia ako  $r$ , prenasledovateľ by mal spomaliť o  $\delta$ , aby sa zväčšil ich vzdialosť rýchlo. Rýchlosť sa potom nastavuje odlišne podľa nasledujúcich troch podmienok:

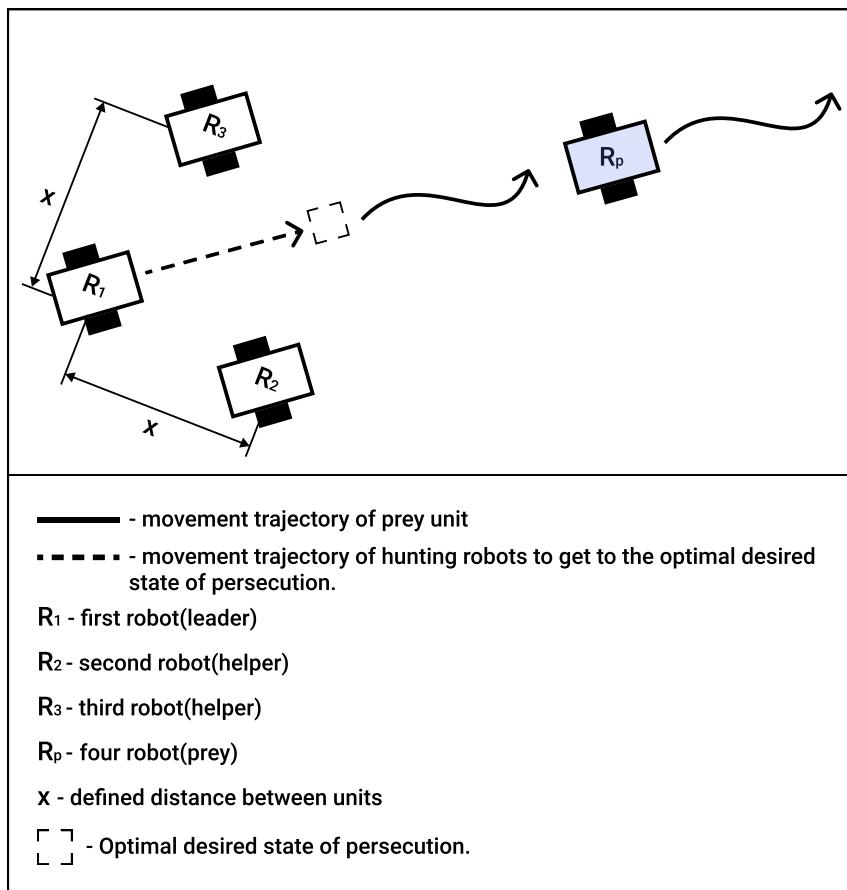
- $d(P, A) > R$ . V tejto situácii je vzdialenosť od prenasledovateľa po korisť veľmi veľká, čo sa často stáva v počiatočnej fáze sledovania, prenasledovateľ musí prenasledovať korisť zrýchlením. Rýchlosť prenasledovateľa sa v každom kroku prudko zvýsi o to, aby sa dosiahlo rýchle zmenšenie vzdialnosti. Z dôvodu obmedzenia maximálnej zmeny rýchlosť by mala byť maximálna rýchlosť prenasledovateľa obmedzená, aby sa zabránilo kolízii do ciela.
- $r \leq d(P, A) \leq R$ . Toto je požadovaný stav prenasledovateľa. Pretože prenasledovateľ je už v bezpečnom dosahu ku koristi, musí iba nasledovať korisť porovnaním jej rýchlosťi s rýchlosťou koristi.
- $d(P, A) < r$ . V takom prípade sa prenasledovateľ drží príliš blízko ku koristi. Prenasledovateľ preto zníži svoju rýchlosť o maximum  $\delta$  m/s, aby získal väčšiu manévrovatelnosť z hľadiska bezpečnosti. Algoritmus 1 sumarizuje navrhovanú schému sledovania koristi-predátora.

V prírode môže lovcovi občas chýbať snímanie koristi v dôsledku oklúzie. Pri robo-

tickom sledovaní môže komunikácia medzi robotmi z času na čas zlyhať. V týchto situáciách s neistotou vnímania musí prenasledovateľ predvídať trajektóriu ciela, aby mohol pokračovať v úlohe sledovania. Predpovedaním trajektórie ciela môže prenasledovateľ približne sledovať cieľ pri neistote vnímania, a preto zvyšuje odolnosť a kontinuitu sledovania. Keď sa vnímanie vráti do normálneho stavu, prenasledovateľ rýchlo napraví chyby sledovania bez vykonania fázy naháňania, ktorá zvyčajne trvá dlhšie. V tomto projekte začleňujem do našej sledovacej schémy prediktívne schopnosti. Keď je korist vnímaná, prediktívny model sa učí predpovedať jej aktuálnu polohu na základe jej predchádzajúceho správania. Akonáhle korist nie je vnímateľná, naučený model sa použije na predpovedanie neznámej polohy. Existujú tri metódy:

1. Jednoduchá inferencia.
2. Autoregresívny model.
3. Echo stavové siete.

V tomto projekte používa sa jednoduchá inferencia. Keď sú informácie o koristi na krátky čas nedostupné, existuje veľmi jednoduchá hypotéza, že zachová rovnaký trend lineárneho pohybu s konštantnou rýchlosťou. Inými slovami, posledná pozorovaná rýchlosť a smer budú fungovať ako odhad neznámej rýchlosťi a smeru počas tohto časového obdobia. Tento predpoklad je veľmi jednoduchý a intuitívny. Pochádza z prirodzeného sledovania človekom. Ľudia budú pokračovať v sledovaní za predpokladu, že cieľ bude pokračovať v ceste rovnakým smerom a rovnakou rýchlosťou, keď nebudú môct cieľ vidieť. Tento predpoklad je ale užitočný iba v situáciách, keď pozorovanie chýba veľmi krátko. Ďalšou výhodou tejto metódy je, že nie je potrebné žiadne učenie, takže je výpočtovo efektívnejšia ako ostatné dve predikčné metódy.



Obrázok 3 – 7 Režim sledovania a navigácia - schéma.

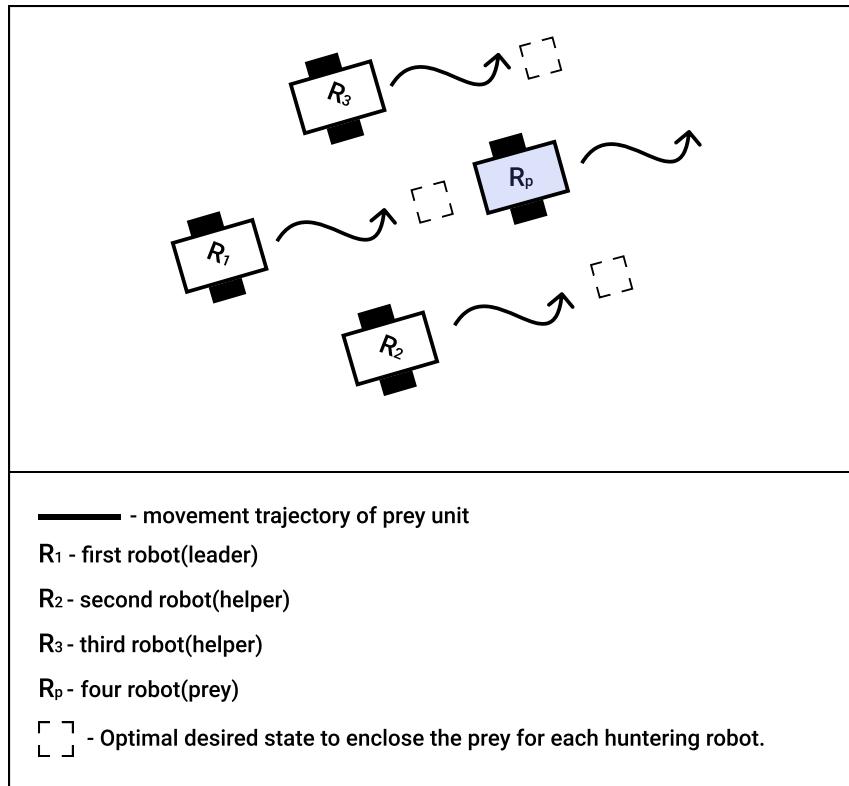
### 3.3.3 Režim znehýbnenia (chytenia) koristi

Toto je posledný režim navigácie, kde je cieľom robotov uzavrieť korist. Problém je tu zložitejší, pretože cieľom je sformovať postavu okolo pohybujúceho sa objektu. Tento režim sa aktivuje pre každého robota osobitne, keď robot dosiahne bod v rámci danej vzdialenosť od koristi (povedzme 10). Kruh obklopujúci korist má ako polomer 10. Tvorba kruhu sa ustanovuje v nasledujúcich krokoch:

1. Vzhľadom na počet robotov a vzdialosť  $l_0$  sa počíta neskoro  $l_1$ , vzdialosť medzi dvoma po sebe nasledujúcimi robotmi vo formácii.
2. Keď roboti dosiahnu bod so vzdialenosťou  $l_0$  od koristi, aktivuje sa režim formovania kruhu. Roboty vykonávajú zložitý pohyb, aby udržali vzdialosť od

koristi konštantnú a rovnú  $l_0$  a súčasne sa vytvorili jednotný kruh okolo koristi.

3. Po dosiahnutí formácie - vykonajú roboti rovnaký typ pohybu ako korist obklopená robotmi.



Obrázok 3 – 8 Režim zatváranie koristi - schéma.

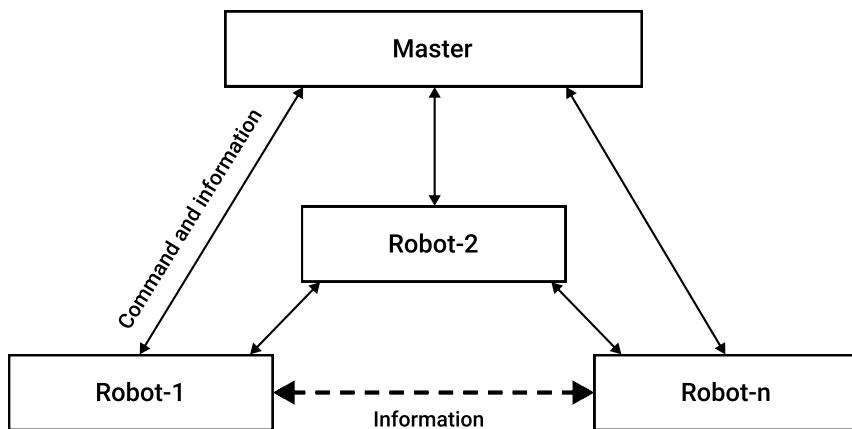
### 3.4 Výber vhodnej metódy robotickej kooperácie

V tejto časti sa budeme zaoberať metódami použitými pri práce nad projektom, analyzujeme tiež výhody a nevýhody rôznych metód a urobíme odôvodnenú voľbu v prospech určitých riešení s prihliadnutím na špecifikáciu vypracovaného projektu.

#### 3.4.1 Centralizované vs. decentralizované

Pre návrh na vysokej úrovni, používa sa zmiešaný riadiaci systém. To riadenie na vysokej úrovni spočíva v ovládaní globálnych parametrov a koordinácií robotov pomo-

cou základnej stanice, ako je spracovanie priatých snímok z kamery, koordinácia a aktivácia režimov pohybu robota, koordinácia pohybov skupiny robotov. Zároveň sú roboty schopné navzájom si prenášať miestne parametre a údaje senzorov. Tento režim bol vybraný, pretože s úplným centralizovaným riadením môžu nastať problémy s rýchlosťou príjmu údajov od všetkých robotov súčasne. A nepoužíva sa decentralizovaný systém, pretože je ťažké v ňom koordinovať prácu všetkých robotov a spracovanie informácií prebieha na samotnom robotovi, čo zvyšuje požiadavky na výpočtovú schopnosť takého robota. Ako výsledok návrh na vysokej úrovni bude využívať systém spätej väzby.

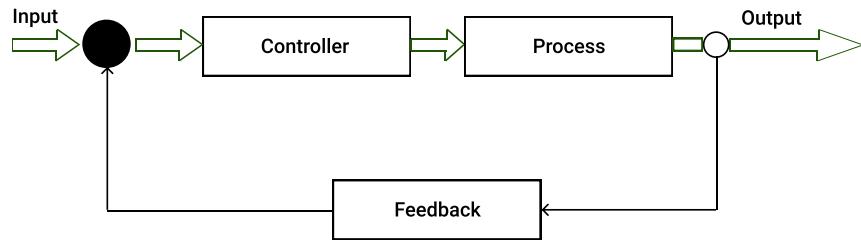


Obrázok 3 – 9 Hybridná architektúra riadenia.

### 3.4.2 Systém spätej väzby(Closed-Loop Control System)

Dizajn s uzavretou slučkou využíva systém spätej väzby. Spätná väzba umožňuje systému určiť súčasné a budúce reakcie systému na základe súčasných alebo minulých informácií. Je to prospešné, pretože sa zvyšuje presnosť a spoloahlivosť výstupu, pretože systém sa môže sám prispôsobiť tak, aby získal požadovaný výkon. Systém s otvorenou slučkou môže robiť úpravy iba na základe vopred naprogramovaných nastavení a nemôže zohľadňovať zmeny prostredia alebo situácie. Rozhodol som sa

implementovať riadiaci systém uzavretej slučky pre robotov. Poloha koristi a robotov sa bude meniť od behu k behu, preto je potrebné mať spätnú väzbu. Okrem toho je dôležité zhromažďovať údaje o spätej väzbe o rýchlosti robota, aby bolo možné vykonať úpravy, aby sa roboty pohybovali správne.



**Obrázok 3 – 10** Systém spätej väzby.

### 3.4.3 Deliberatívne vs. reaktívne riadenie

V tomto prípade sa používa reaktívny riadiaci prístup, pretože zámerný riadiaci prístup založený na plánovaní pohybu a trajektórie pohybu si vyžaduje predbežné znalosti prostredia, aby bolo možné plánovať pohyby robotov, čo pre tento projekt nie je možné. Pomocou reaktívneho riadenia som sa rozhodol implementovať hierarchickú pohybovú stratégiu, v ktorej, ako sme diskutovali v časti 2.4.2, je jeden robot považovaný za vedúceho a ostatní roboti sú druhí. Vodca zvyčajne sleduje trajektóriu a nasledovníci sledujú jeho transformované súradnice.

## 4 Simulačné experimenty

Táto časť obsahuje popis výkonov experimentov. Každý experiment je pripravený v simulačnom prostredí Webots. Knižnica Python je dodávaná ako súčasť tejto bakalárskej práce na vykonávanie experimentov a na meranie hodnôt.

### 4.1 Prehľad a príprava prostredia

**Simulátor robotov Webots [15].** Webots je simulátor robotov s otvoreným zdrojovým kódom, ktorý poskytuje kompletne vývojové prostredie pre modelovanie, programovanie a simuláciu robotov. Tisíce inštitúcií po celom svete ho používajú na výskum a výučbu. Webots bol vyvinutý v spolupráci so Švajčiarskym federálnym technologickým inštitútom v Lausanne, dôkladne testovaný, dobre zdokumentovaný a nepretržite udržiavaný od roku 1998. Toto je najefektívnejší spôsob, ako rýchlo dosiahnuť profesionálne výsledky.

#### Vlastnosti Webots-u

- Vývoj vlastných robotov alebo práca s pripravenými možnosťami.
- Vývoj 3D prispôsobeného prostredia (povrch, prekážka, cieľ, obloha, fyzický model správania).
- Vývoj logiky robota.
- Simulácia a testovanie.
- Záznam videa alebo snímka obrazovky simulácie.

**Výskumné oblasti Webots-u.** Mnoho projektov mobilnej robotiky sa v priebehu rokov používali Webots v nasledujúcich oblastiach:

- prototypy mobilných robotov (vedecký výskum, automobilový priemysel, letectvo a aeronautika, výroba vysávačov a hračiek, koníčky atď.);

- výskum spôsobov pohybu robotov (roboty s mechanickými končatinami, humanoid, štvornohí roboti atď.);
- multi-agentový výskum (inteligencia rojov, skupiny spolupracujúcich mobilných robotov atď.);
- výskum adaptívneho správania (genetický algoritmus, neurónové siete, AI atď.);
- školenie v robotike (prednášky z robotiky a programovania v C / C ++ / Java / Python atď.);
- robotické súťaže - napríklad "Rat's Life".

**Simulácia vo Webots-e.** Simulácia vo Webots-e pozostáva z nasledujúcich komponentov:

- svetový súbor webots (.wbt), ktorý definuje jedného alebo viacerých robotov a ich prostredie;
- súbor .wbt niekedy závisí od externých súborov (.proto) a textúr;
- jeden alebo viac programov pre radiče robotov (v C / C ++ / Java / Python / MATLAB);
- ďalší doplnok na simuláciu fyziky na úpravu štandardného fyzického správania Webotov (C / C ++).

**Programovací jazyk.** Program pre ovládač robota môžete napísť do Webots-u v jazykoch C ++, Java, Python alebo MATLAB. Je možné zvoliť si oblúbený programovací jazyk. Ďalej sme pre účely našej práce použili programovací jazyk - Python.

Vývojár môže meniť vlastnosti robotov a ich prostredie: fyzické vlastnosti, typy objektov atď. Tieto popisy sú usporiadane v hierarchickej podobe a na ich vytvorenie

sa používa interný jazyk DSL (Domain Specific Language). Používanie tohto jazyka prináša flexibilné výsledky - vývojárom by to malo robiť starosti ktoré jazyky alebo technológie sa používajú na zabezpečenie robotov ovládajúcich štadium opisu fyzických vlastností. Tento softvér poskytuje pomerne realistickú fyziku simulovaných svetov (pomocou fyzikálneho nástroja Open Dynamic Engine). V systéme je celkom dobrá dokumentácia Angličtina popisujúca všetky nuansy. K dispozícii sú verzie softvéru Webots-a pre Windows, MacOS a Ubuntu.

## 4.2 Ovládanie robota Koristi

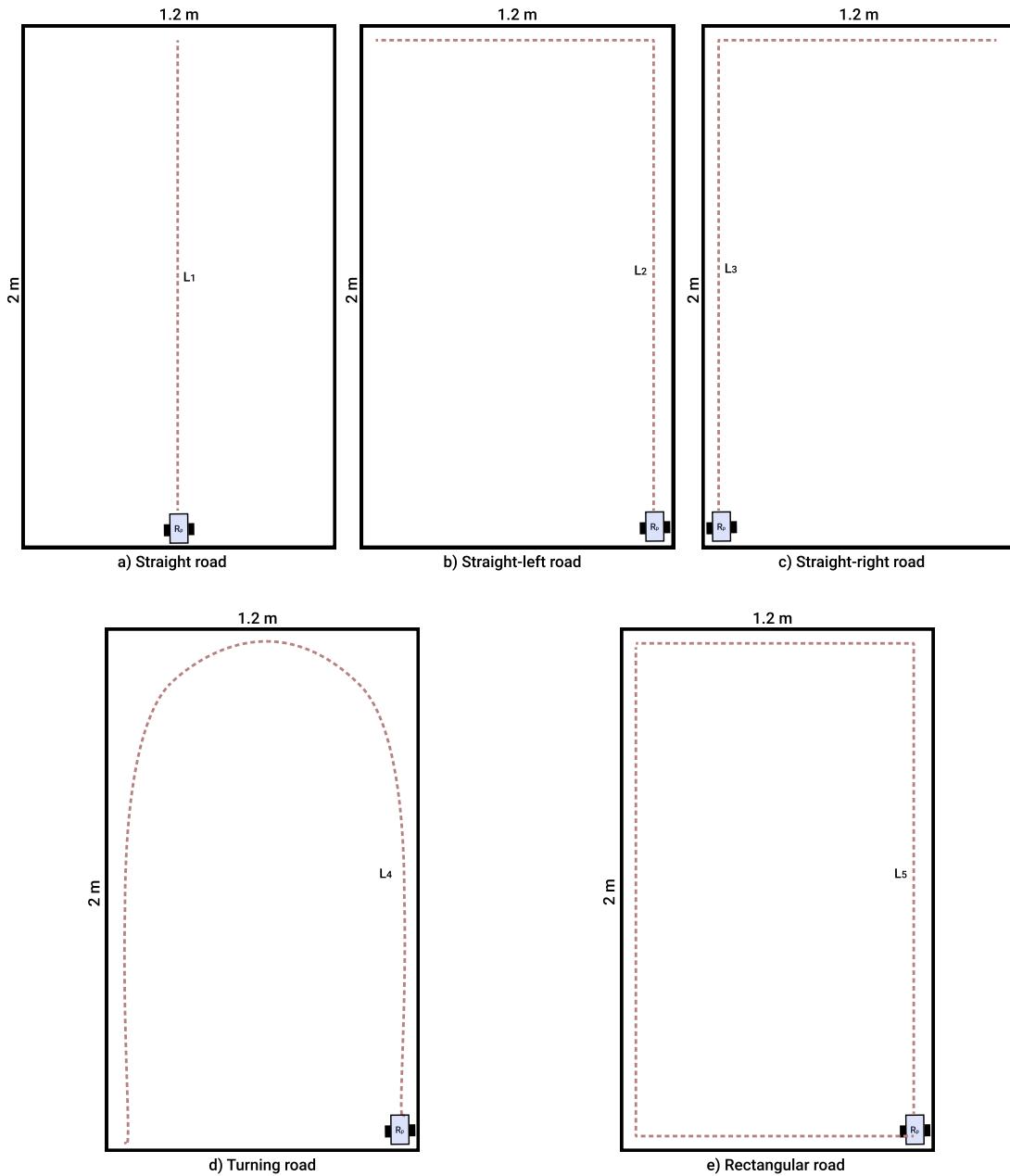
Pokus sa vykonáva na hladkej rovnej ploche. Testy sa uskutočňovali na simulátore Webots-e. Všetky testy sa uskutočňovali v rovnakej ploche. Vytvorili sme päť rôznych ciest, cez ktoré robot môže prechádzať: priamu, priamu-pravú, priamu-lavú, obrat a obdlžník. Na všetkých týchto cestách bolo pomocou klávesnice vedených 5 chodníkov. To nám dáva celkovo 25 dráh rôznych typov. Hodnoty zaznamenané počas každej z týchto trás sú uvedené v tabuľkách, aby sa získala efektivita kontroly. Tabuľka 4-1 ukazuje načasovanie cesty robota pomocou každého z 5 režimov.

N	Priama	Priama-práva	Priama-láva	Obrat	Obdlžník
1	5.0	8.2	8.69	13.3	16.4
2	5.2	8.51	8.9	13.3	16.2
3	5.0	8.65	8.7	13.4	16.5
4	5.3	8.2	8.53	13.3	16.4
5	5.5	8.2	8.27	13.4	16.3
Stredné	5.13	8.24	8.618	13.34	16.36

**Tabuľka 4 – 1** Zoznam časovaní robota na pokrytie rôznych typov tratí

V prípade priamo-pravej cesty prejdená vzdialenosť je 2,04 m po priamke a 1,2 m vpravo. V prípade priamo-lavej cesty je prekonaná vzdialenosť 2 m v priamke a 1,2 m vľavo. Stĺpec 4 zobrazuje časovanie cesty otočenia o U. Prejdená vzdialenosť je

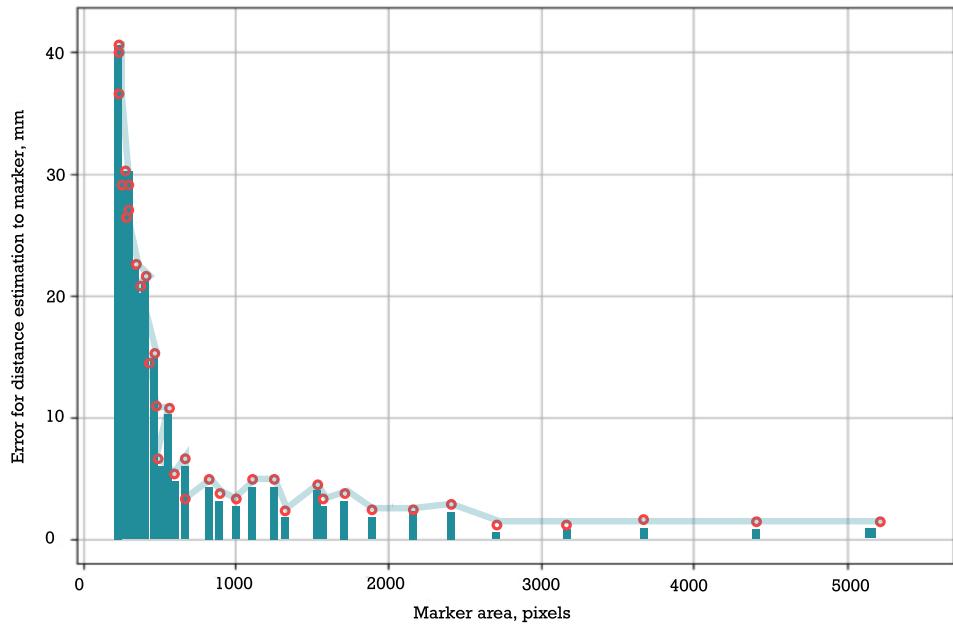
2 m priamo hore a dole a šírka je 1,02 m v prípade obdĺžnikového časovania cesty uvedeného v stĺpci 5



**Obrázok 4 – 1** Ilustrované cesty, použitie na testovanie.

### 4.3 Detekcia značiek

V tejto simulácii bola veľkosť značky Aruco 6 x 6 mm. Na detekciu značiek v obrazu použite štandardne algoritmy na rozpoznávanie značiek ArUco v OpenCV so štandardnými parametrami.

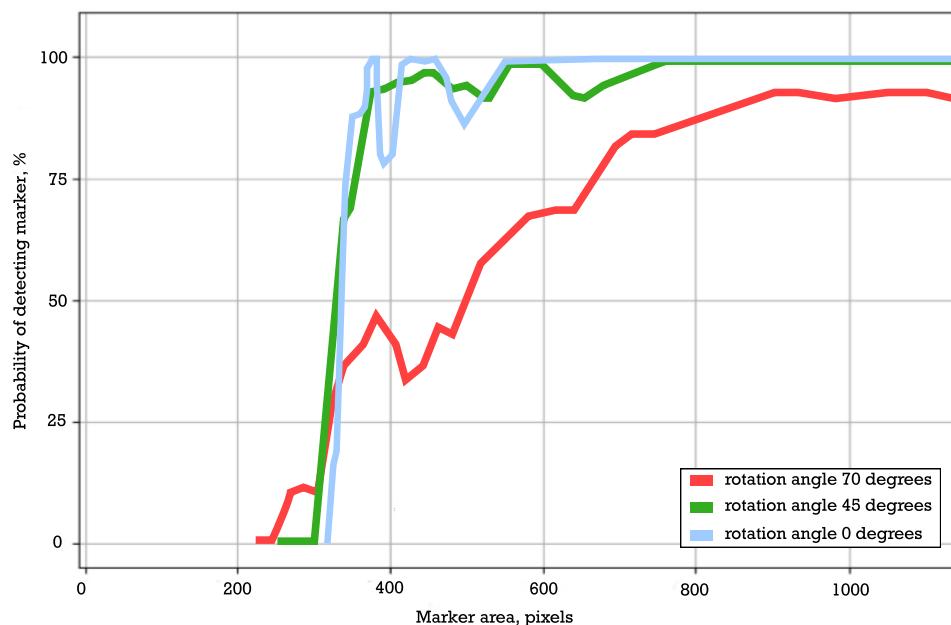


Obrázok 4 – 2 Závislosť vplyvu veľkosti oblasti značky na chybe pri zistovaní jej polohy.

Uvádzame dva príklady výsledkov získaných modelovaním. Jedným z dôležitých parametrov, ktoré môžu byť zaujímavé pri konštrukcii meracích systémov s referenčnými značkami, je presnosť zistenia ich polohy v priestore, a to v uhloch náklonu aj vo vzdialosti. Závislosť vplyvu oblasti značky na chybu ovplyvňujúcu jej polohu získanú pri modelovaní je uvedená na obr. 4-1. Závislosti ukazujú, že zmenšenie veľkosti značky na obrázku vedie k zvýšeniu chyby.

Obrázok 4-2 zobrazuje pravdepodobnosť nájdenia značky v závislosti od jej oblasti na obrázku pre 3 rôzne uhly naklonenia: 0, 45, 70. Zo závislosti je zrejmé, že značka pravdepodobnosť rozpoznania pre uhol náklonu 0 ° sa výrazne lísi od 100 v dvoch

oblastiach: s plochou 300 pixelov a s plochou v rozsahu 350-450 pixelov. Tento výsledok je dôsledne opakované pre rôzne sériu modelovania s rôznymi počiatočnými parametrami. Navyše pre značky s niekoľkými číslami v slovníku druhá oblasť (oblasť v rozsahu 350-450 pixelov) zmizne. Vysvetlenie tohto správania je jedna z úloh pre ďalšiu prácu.



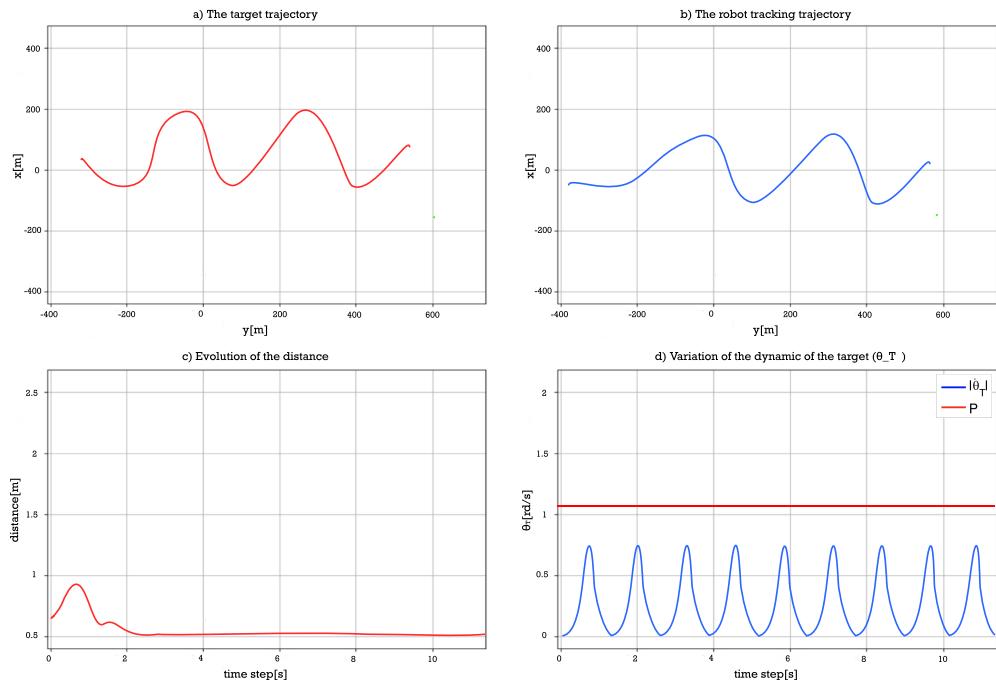
**Obrázok 4 – 3** Pravdepodobnosť detekcie značky v závislosti od jej oblasti na obrázku pre 3 rôzne uhly naklonenia: 0°, 45°, 70°.

#### 4.4 Navigácia pre systém viacerých robotov

V tejto časti sa najskôr simuláciou sleduje správanie robota vzhľadom na jeho dynamický pohyb ciela. Ďalej sa vykoná experiment s tromi robotmi na dosiahnutie a udržanie trojuholníkového útvaru.

#### 4.4.1 Variácia uhlovej rýchlosťi virtuálnej štruktúry

Táto časť ukazuje dôležitosť ohraničenia uhlovej rýchlosťi virtuálnej štruktúry  $\theta$  podľa kinematických obmedzení robotov. Preto sa simuluje mobilný robot, ktorý dosiahne virtuálny cieľ. Maximálna uhlová rýchlosť robota je  $\omega_{max} = 3./s$ . Vyberieme  $k = 0,6s^{-1}$ .



**Obrázok 4–4** Nie sú vynútené nežiaduce kmity dráhy robota prechodnej fázy.

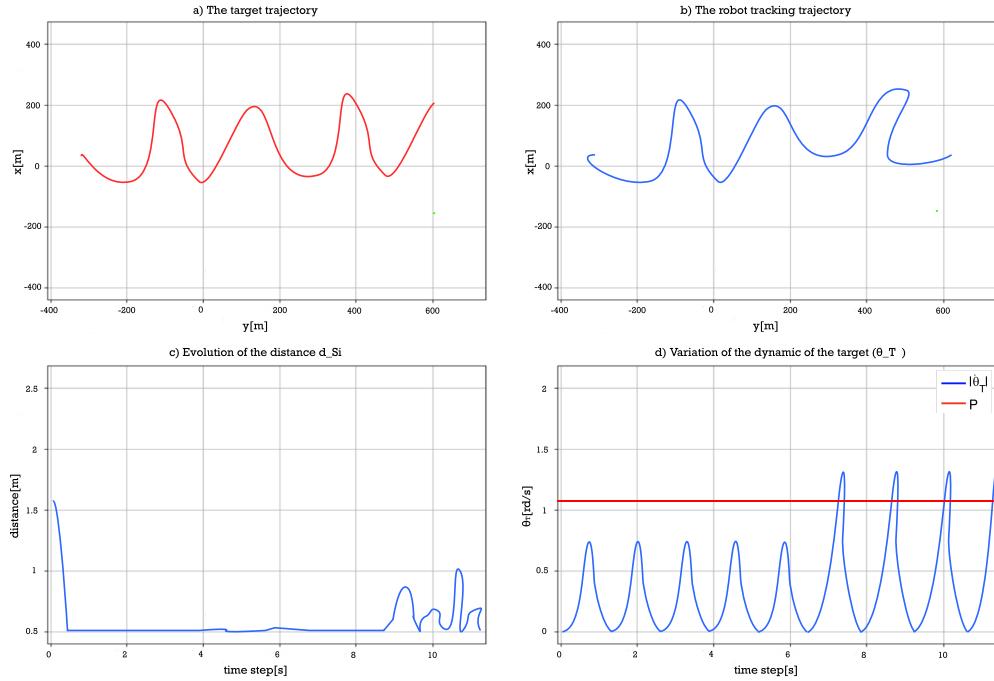
Podľa vzťahu a kvôli zjednodušeniu zápisu na číslach navrhujeme zaznamenať  $P = \omega_{max} - k\pi$ . Na základe zvolených hodnôt  $\omega_{max}$  a  $k$  nájdeme  $P = 1,1$ . Najskôr sa navrhuje ukázať dôležitosť prechodnej fázy, pri ktorej musí byť variácia  $\theta_T$  nastavená na 0. Preto teda na obrázku 4-4(d) vidíme, že  $\theta_T$  sa zvyšuje na začiatku simulácie (od 0,1 s) a cieľová trajektória nasleduje okamžite po významnej krivke (porovnajte obrázok 4-4(a-b))). Následne sledujeme oscilácie na trajektórii robota. Robot správne dosiahne cieľ iba v prípade, že tento má priamu trajektóriu ( $\theta_T = 0$ ). Obrázok 4-4(d)

to potvrdzuje. V skutočnosti, aj keď  $\theta_T$  uspokojí podmienku opísanú v rovnici

$$|\theta_T| \leq \omega_{max} - k\pi \quad (4.1)$$

, môžu sa objaviť oscilácie, ak nie je zavedená prechodná fáza. Prirodzene, že vzdialenosť  $d_S$  v tomto prípade osciluje (porovnajte obrázok 4-4(c)).

Obrázok 4-5 zobrazuje dôležitosť splnenia stavu opísaného vo vzťahu 4.1 po prechodnej fáze. Akonáhle je cieľ dosiahnutý ( $\theta_T = 0$  do okamihu 0,5 s), je splnená aj podmienka 4.1. Vidíme, že robot ide smerom k cieľu. Aj keď sa zvyšuje, variácia P je taká, že  $\theta_T < P$  (porovnajte obrázok 4-5(d)). V tomto intervale robot správne sleduje svoj cieľ (porovnajte obrázok 4-5(a-b)). Vzdialenosť  $d_{S_i}$ , ktorá ich oddeluje, je  $d_{S_i} = 0$  (porovnajte obrázok 4-5(c)). Po 9,5 s odstránime obmedzenie (4.1) tak, že  $\theta_T$  môže byť  $\theta_T > P$ . Vidíme, že robot nemôže sledovať cieľ.



**Obrázok 4 – 5** Nežiaduce oscilácie dráhy robota, ak nie sú splnené predpokladané obmedzenia dynamického cieľa (ak  $\theta_T > P$ ).

#### 4.4.2 Výsledky experimentu: 3 roboty s dosiahnuteľnou virtuálnou štruktúrou

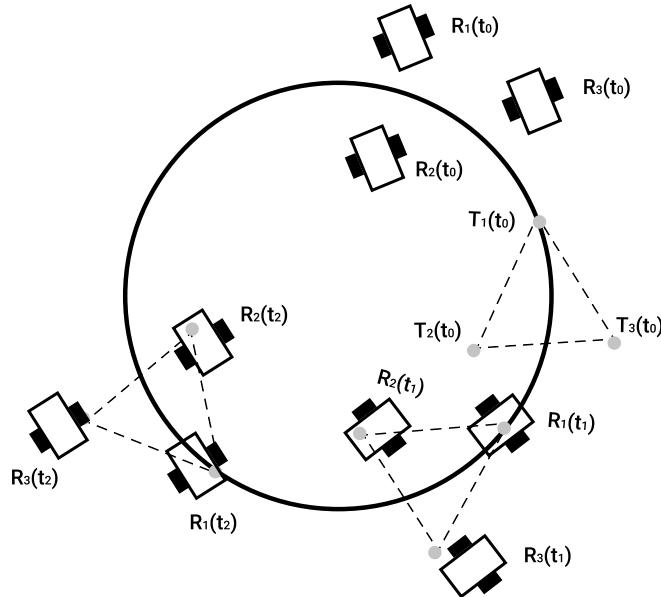
Pre tento experiment boli použité 3 roboty. Každý z nich vie o polohe a orientácii ostatných, tieto informácie boli zhromaždené pomocou odometrie. Pre tento experiment bol použitý centralizovaný systém. Experimentom je pohyb robotov v kruhu v smere hodinových ručičiek.

Virtuálna štruktúra má priamu trajektóriu. Tu sa navrhuje rozšíriť na kruhový pohyb tak, aby všetky ciele napriek ich kinematickým obmedzeniam zostali dosiahnuteľné všetkými robotmi. S vedomím, že dynamika virtuálnej štruktúry musí sledovať vzťah, bol polomer  $R_{vs}$  kruhového pohybu formovaný ako:

$$R_{vs} = \frac{v_T}{\theta_T} > \frac{v_T}{\omega_{max} - k\pi}; \quad (4.2)$$

kde  $v_T$  je konštant a  $v_T \ll v_{max}$

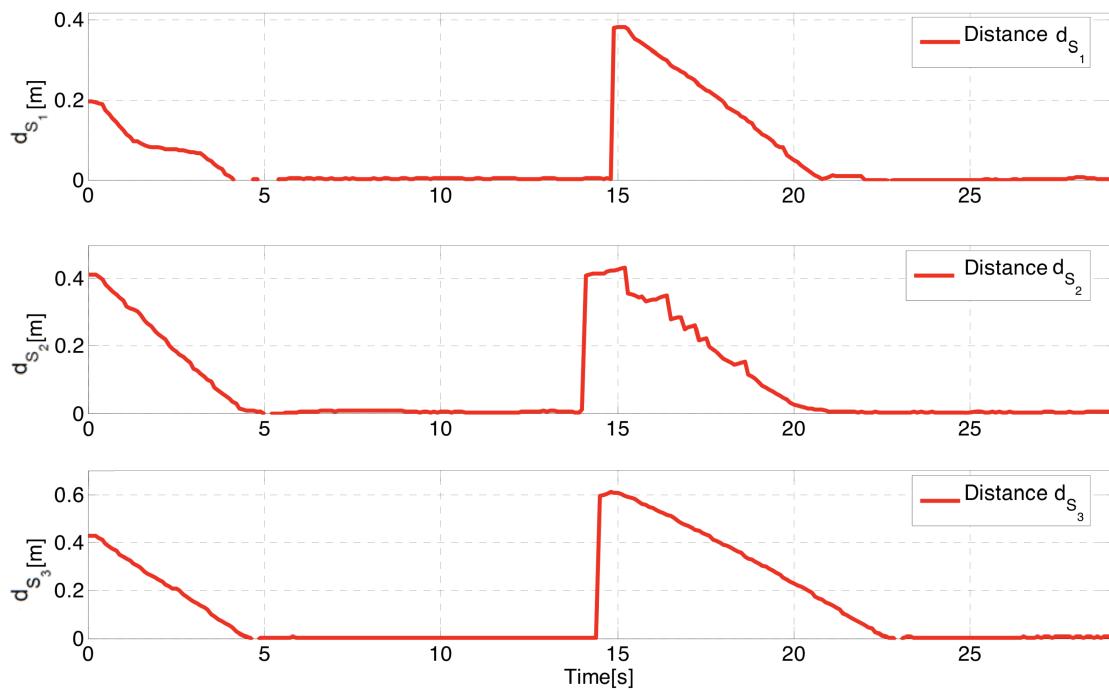
Uvažujeme o pohybe v smere hodinových ručičiek (obrázok 4-6)



Obrázok 4 – 6 Trajektória robotov.

Ako vidíme na obrázku 4-6, skupina robotov začala v polohe udalosti do požadovanej

polohy, ale nie v správnej orientácii. Ako východiskový bod experimentu si vezmeme bod  $T$ , v ktorom už roboty vytvárajú požadovanú štruktúru. Ako vidíme v čase  $t_1$ , ktorý je prvým kontrolným bodom, práce takmer sledujú ideálnu štruktúru, ale stále existujú menšie chyby. Spájam to s nedokonalým výkonom snímača a nedokonalou komunikáciou. Na nasledujúcom kontrolnom bode, v čase  $t_2$ , vidíme o niečo lepsí obraz o dodržiavaní ideálnej štruktúry, ale stále s malými chybami. Ak to vezmeme ako celok, môžeme s istotou povedať, že skupina robotov sa pri pohybe v kruhu drží štruktúry, a drobné chyby je nevýznamný pri plnení požadovaného typu úlohy, a to prenasledovania koristi. Vzdialosti medzi robotmi a ich cieľmi sú uvedené na obrázku 4-7. Zmenšili sa oni do 0, čo potvrdzuje, že tvar bol dosiahnutý a udržiavaný. Keď sa zmenila dynamika virtuálnej štruktúry, roboty boli daleko od svojich cieľov, čo vysvetluje pozorované skoky.



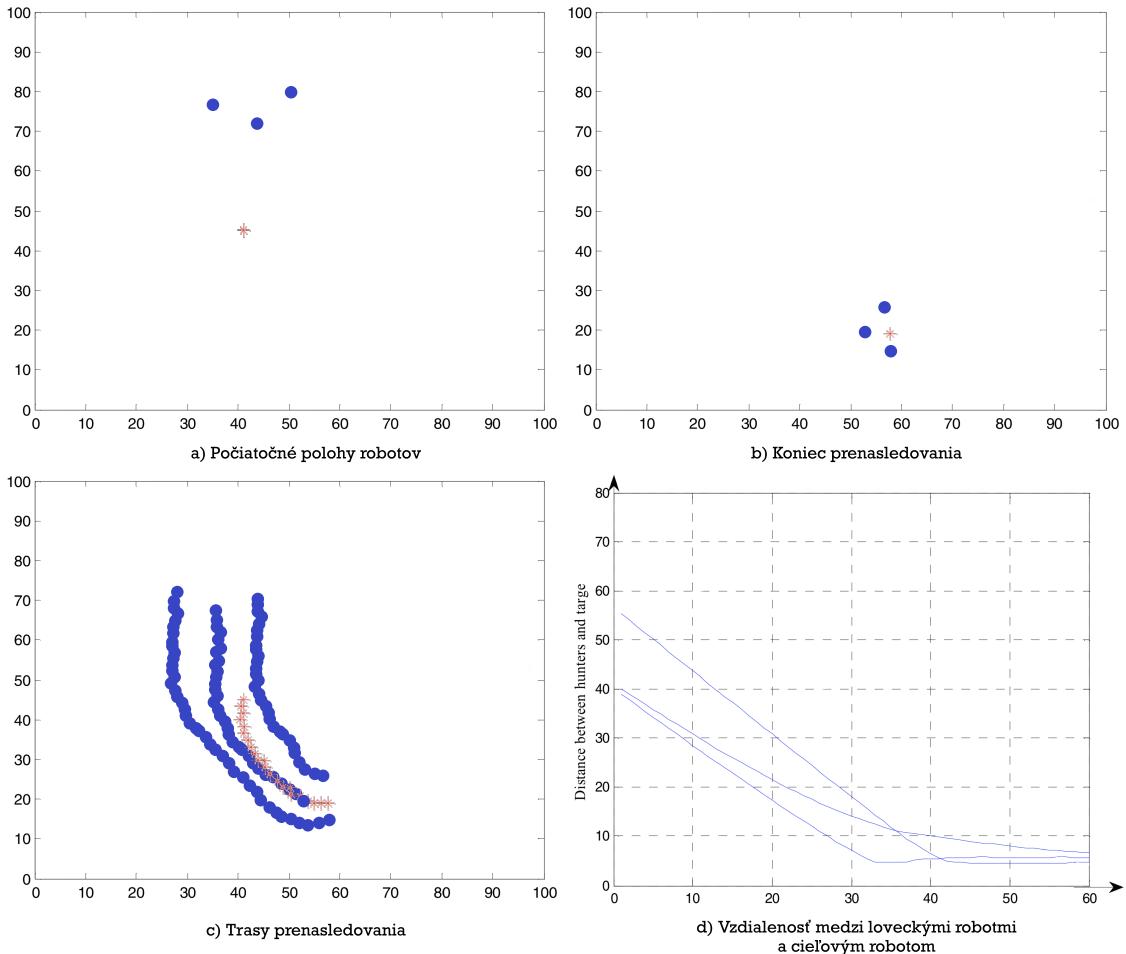
Obrázok 4 – 7 Zmena vzdialenosťi  $d_{S_i}$  medzi robotom i a zvoleným cieľom ( $i = 1..3$ ).

## 4.5 Proces prenasledovania

Táto časť používa na simuláciu v experimentoch grafické rozhranie Rqt a RVIZ. V oblasti štvorca  $100\text{m} * 100\text{m}$  nie je žiadna prekážka. Modrá bodka je v mene polovníckeho robota, červená hviezda predstavuje cieľového robota, ako je to znázornené na obrázku 4-8(a). Počiatočné polohy polovníckych robotov a cieľového robota sú náhodne rozložené v rámci štvorcovej oblasti, čo viac odráža platnosť simulácia. Po začiatku úlohy prenasledovania môže každý robot získať informácie o polohe všetkých robotov v prostredí v reálnom čase. Pri úspešnom dosiahnutí, roboti chytia cieľ a znemožnia mu pohyb, čo sa nazýva koniec prenasledovania. V prvej sade experimentov sú tri roboty prenasledovateľa, ktorých počiatočné informácie o polohe sú  $(38, 77)$ ,  $(44, 73)$  a  $(52, 81)$ . Informácie o počiatočnej polohe cieľového robota sú  $(40, 45)$ , ako je znázornené na obrázku 4-8(a).

Konečný výsledok prenasledovania je uvedený na obrázku 4-8(b). Tri prenasledovatelia sa rovnomerne rozložia okolo cieľového robota, aby si cieľ nemohol zvoliť ďalší pohyb, čo znamená úspešný lov. Z trajektórie procesu prenasledovania, ako je znázornené na obrázku 4-8(c), vidíme, že trajektória prenasledovacieho robota je veľmi hladká. Keď štyri prenasledovacie roboty dorazia do potrebnej oblasti, začnú upravovať svoje polohy tak, aby obkolesili cieľ. Červená krivka predstavuje trajektóriu ciela a vidíme, že korist tiež neustále uniká prenasledovateľov. Ako je znázornené na obrázku 4-8(d), os  $X$  predstavuje počet opakovania herného procesu, os  $Y$  predstavuje vzdialenosť medzi prenasledovacím robotom a cieľom a všetky tri modré plné čiary predstavujú krivku zmeny vzdialenosťi medzi tromi robotmi a cieľom resp. Ukazuje to, že roboty sa postupom času neustále približujú k cieľu. Pri hľadaní potrebnej polohy sa roboti začnú skôr pohybovať s cieľom, ako by sa mali približovať, čo dokazujú sklony kriviek plných modrých čiar. Keď posledný prenasledovací robot dosiahne svoju pozíciu, cieľ sa považuje za úspešne chytený a systém sa zastaví.

Tabuľka 4-2 zobrazuje priemerný počet krokov a úspešnosť v loveckej úlohe s podmienkou vyššej rýchlosťi prenasledovateľov a náhodným generovaním informácií o



Obrázok 4 – 8 Výsledky simulácie

Veľkosť oblasti	Priemerný počet krokov	Úspešnosť
100m * 100m	90.26	78%
60m * 60m	62.14	76%
40m * 40m	47.23	69%

Tabuľka 4 – 2 Výsledkom simulácie na ploche 100m \* 100m vs 60m \* 60m

ich polohe. Ako je uvedené v tabuľke, pri veľkosti oblasti 100 m x 100 m bude úspešnosť 78%. Medzitým je priemerný počet krokov k lov 90,26 krokov. Pre ďalší variant bol experimentálny priestor zmenený na plochu 60m \* 60m. Miesta loveckých robotov sú generované náhodne a maximálna rýchlosť loveckých robotov je stále vyššia

Rýchlosť prenasledovateľov	Priemerný počet krokov	Úspešnosť
0.05	*	0%
0.07	99.26	20%
0.1	77.48	100%
0.2	50.04	100%
0.4	34.28	100%

**Tabuľka 4–3** Výsledky simulácie 3 prenasledovateľov v oblasti 100 m \* 100 m

ako cieľový robot. V súčasnosti majú tieto tri lovy takmer rovnakú úspešnosť ako v predchádzajúcej oblasti, ale priemerný počet krokov je znížený na 62,14 krokov. Teda údaje v tabuľke 4-2 ukazujú, že počiatočné polohy prenasledujúcich robotov majú vplyv na poľovnícku úlohu. Tabuľka 4-3 ukazuje výsledky loveckých experimentov, v ktorých traja roboti prenasledovatelia náhodne vygenerovaní v oblasti 100 m \* 100 m a pri rôznych rýchlosťach dokončili 20 experimentov. Ako ukazuje tabuľka, pri rýchlosťi 0,05 m/krok, ktorá je rovnaká ako u robota koristi, nie je lovecký robot schopný dokončiť úlohu. Keď sa rýchlosť prenasledovateľov zvýši o 0,02 m/krok, úspešnosť lovú sa zvýši tiež na 20% a priemerný počet krokov k úspešnému lovú je 99 krokov. Keď sa rýchlosť zvýši na 0,1 m/krok, úspešnosť je až 100% a priemerný počet krokov je 77,48 kroku. Ak sa rýchlosť zvýši na 0,2 m/krok a 0,4 m/krok, priemerný počet krokov sa zodpovedne zníži. Ukazuje, že relatívna rýchlosť medzi loveckými robotmi a cieľovým robotom je veľmi znepokojujúca z hľadiska efektívnosti kooperatívneho prenasledovania s viacerými robotmi v tabuľke 4-3.

Stručne povedané, navrhovaný algoritmus môže efektívne dokončiť úlohy prenasledovania s viacerými robotmi. Pri úlohe kooperatívneho prenasledovania s viacerými robotmi sú dôležitými faktormi je počet prenasledujúcich robotov, počiatočná poloha robotov a maximálna rýchlosť robotov. Keď je počet prenasledujúcich robotov veľký a ich polohy sú bližšie k robotu Koristi a robot prenasledovateľ je rýchlejší ako robot Korist, bude úloha dokončená ešte lepšie.

## 5 Záver

Táto bakalárska práca sa zameriava na výskum v oblasti interaktívnych robotických skupín a návrh kooperatívnych úloh vykonávaných rôznymi typmi robotov. Výskum je vedený v oblasti metód a algoritmov riadenia skupiny robotov a problémy, ktoré je potrebné pri ich vývoji vyriešiť, aby sa dosiahli konkrétné ciele. Bakalárska práca obsahuje prehľad úloh, ktoré je možné vyriešiť spoluprácou robotov, a popis rojovej robotiky - relatívne novej oblasti robotiky, založenej na myšlienke simultánneho riadenia veľkého počtu robotov. Táto bakalárska práca bola vypracovaná podľa pokynov vedúceho. Podmienečne je možné túto prácu rozdeliť na 4 časti (úlohy), pretože boli vyvinuté výsledky a pre každý režim boli urobené simulácie pre dokonalosť odhadu. Prvá simulácia bola vykonaná pre riadenie koristového robota pomocou klávesnice. Boli vytvorené rôzne stopy a odhadovaná efektivita riadenia. Druhý experiment bol vykonaný pre režim detekčných značiek, ktorý sa používal na detekciu robota. Výsledkom je efektivita detekcie tejto značky v rôznych podmienkach a pod rôznymi uhlami. Tretia simulácia bola zameraná na formovanie kooperatívnej navigácie. Výsledkom je, že sme presvedčení o dobrej koordinácii medzi robotmi. A posledná simulácia bola pre proces prenasledovania. Táto bakalárska práca a uskutočnenie experimentov sú relevantné, pretože pre spoločnosť je dôležitý aktívny vývoj systémov robotickej spolupráce a potreba skúmania algoritmov a metód riešenia rôznych problémov. Vďaka tomu, že sa ľudia snažia zlepšiť svoju životnú úroveň, bude sa robotika aktívne rozvíjať. V blízkej budúcnosti budú vedci riešiť problémy komunikácie medzi robotmi na veľké vzdialenosťi, vytvoria nové spôsoby pohybu a vzájomnej spolupráce pri dosahovaní lepších výsledkov. Tieto experimenty je možné rozšíriť a skomplikovať pridaním nových systémov a typov robotov v mnohých smeroch. Existujú možnosti na vývoj schém a riešení pre nové typy úloh vykonávaných rôznymi typmi robotov alebo na hlboké preskúmanie a vývoj metód pre multi-robotiku.

## Literatúra

- [1] P. Aparício and P. Lima, “Autonomous distributed control of a population of cooperative robots,” 08 1999.
- [2] J. McLurkin, *Using Cooperative Robots for Explosive Ordnance Disposal*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, 1996.
- [3] K. Xu, *Integrating Centralized and Decentralized Approaches for Multi-Robot Coordination*. Rutgers University, 2010.
- [4] J. Vaščák, *Autonomous Decision Systems Handbook*. BEN, 2011. ISBN 978-80-7300-415-6.
- [5] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [6] P. Ogren, “Improved predictability of reactive robot control using control lyapunov functions,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1274–1279, 2008.
- [7] R. C. Arkin, “Cooperation without communication: Multiagent schema-based robot navigation,” *Journal of Robotic Systems*, vol. 9, no. 3, pp. 351–364, 1992.
- [8] M. J. Matarić, *Interaction and Intelligent Behavior*. Massachusetts Institute of Technology, 1994.
- [9] J. Wang, “On sign-board based inter-robot communication in distributed robotic systems,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 1045–1050 vol.2, 1994.

- [10] M. Peshkin, J. Colgate, W. Wannasuphoprasit, C. Moore, R. Gillespie, and P. Akella, “Cobot architecture,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.
- [11] B. M. I. Badano, *A Multi-agent Architecture with Distributed Coordination for an Autonomous Robot*. Campus Montilivi - Edifici PIV, E- 17071 Girona, 2008.
- [12] P. Caloud, W. Choi, J.-C. Latombe, C. L. Pape, and M. Yin., *Indoor automation with many mobile robots*. IEEE/RSJ IROS, 1990.
- [13] L. Parker, “Alliance: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 2, pp. 776–783 vol.2, 1994.
- [14] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [15] “Webots documentation: introduction to webots.” <https://cyberbotics.com/doc/guide/introduction-to-webots>, 2021. Accessed: 2021-05-21.