

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

**Využitie prostriedkov internetu vozidiel v
navigácii mobilných robotov**

Diplomová práca

2023

Bohdan Tanasov

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

**Využitie prostriedkov internetu vozidiel v
navigácii mobilných robotov**

Diplomová práca

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

Košice 2023

Bohdan Tanasov

Abstrakt v SJ

Cieľom tejto práce bolo vyvinúť automatizovaný systém riadenia pre bezpilotné lietadlo, ktorý využíva zabudovanú kameru na navigáciu a zber údajov. Systém bol implementovaný pomocou programovacieho jazyka Python spolu s balíkom OpenCV pre algoritmy počítačového videnia a značkami ArUco pre merania, NodeJS a React. Pomocou systému môžeme manipulovať s viacerými dronmi a so samostatným dronom z webovej stránky a môžeme vidieť údaje dronov a súradnice dronov v priestore pomocou značiek aruco. Na záver možno konštatovať, že systém úspešne dosiahol svoj cieľ, ktorým je adaptácia autonómneho navigačného systému pre bezpilotné lietadlá v interiéri. Následne spracované súbory údajov boli dostatočne dobre filtrované, aby sa mohli považovať za adekvátne merania. Program by sa mohol v budúcnosti vylepšiť tak, aby podporoval viac príkazov pre skupiny dronov, napr. postaviť kolo alebo iné postavy. Takýto systém sa dá použiť aj pre priemyselné bezpilotné lietadlá, kde sú polohy AR markerov vopred známe.

Klúčové slová

UAV, autonómna navigácia, AR marker, určovanie polohy, merania

Abstrakt v AJ

The objective of this work was to develop an automated control system for an unmanned aircraft that uses an embedded camera for navigation and data collection. The system was implemented using the Python programming language along with the OpenCV package for computer vision algorithms and the ArUco tags for measurements, NodeJS and React. Using the system, we can manipulate multiple drones and a single drone from a web page and can see the drone data and drone coordinates in space using aruco tags. In conclusion, the system has successfully achieved its

goal of adapting an autonomous navigation system for drones indoors. The subsequently processed datasets were sufficiently well filtered to be considered as adequate measurements. The program could be improved in the future to support more commands for groups of drones, e.g., build a bike or other figures. Such a system could also be used for industrial drones where the positions of the AR markers are known in advance.

Kľúčové slová v AJ

UAV, autonomous navigation, AR marker, positioning, measurements

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

Z A D A N I E
D I P L O M O V E J P R Á C E

Študijný odbor: **Informatika**

Študijný program: **Inteligentné systémy**

Názov práce:

Využitie prostriedkov internetu vozidiel v navigácii mobilných robotov

Use of the Internet of Vehicles Means in the Navigation of Mobile Robots

Študent: **Bc. Bohdan Tanasov**

Školtiteľ: **doc. Dr. Ing. Ján Vaščák**

Školiace pracovisko: **Katedra kybernetiky a umelej inteligencie**

Konzultant práce: **Ing. Dušan Herich**

Pracovisko konzultanta: **Katedra kybernetiky a umelej inteligencie**

Pokyny na vypracovanie diplomovej práce:

1. Analýza existujúcich prístupov a technológií pre využitie infraštruktúry internetu vozidiel v navigácii mobilných robotov.
2. Návrh architektúry systému pre využitie prostriedkov internetu vozidiel v navigácii mobilných robotov.
3. Implementácia navrhnutého systému a jeho testovanie v rôznych prostrediacach a scenároch.
4. Overenie funkčnosti a spoločalivosti systému na lietajúcich vozidlách.
5. Porovnanie výkonu a efektivity navrhnutého systému s existujúcimi prístupmi a technológiami.
6. Vypracovať dokumentáciu podľa pokynov vedúceho záverečnej práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 21.04.2023

Dátum zadania diplomovej práce: 31.10.2022

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

Čestné vyhlásenie

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

Košice 21. 4. 2023

.....

Vlastnoručný podpis

Poděkovanie

Najskôr by som sa chcel poděkovat 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 pripúštal, aby tato práca bola mojou vlastnou, ale vždy, keď si myslel, že to potrebujem, ma nasmeroval správnym smerom.

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

Predhovor

Téma práce sa zaoberá využitím dronov Tello v skupinovej koordinovanej misii, kde bola navrhnutá a implementovaná webová aplikácia, ktorá umožňuje ovládanie viacerých dronov naraz. V práci je tiež popísaná implementácia systému, ktorý umožňuje sledovanie pozície dronov pomocou kamery a detekcie Aruco markerov. Cieľom tejto práce bolo ukázať, ako efektívne ovládať viacero dronov naraz pomocou jednoduchej webovej aplikácie a koordinovaného systému detekcie pozície dronov. V práci sa tiež zaoberáme riešeniami problémov pri použití viacerých dronov naraz a analyzujeme výsledky našich experimentov a testov.

Obsah

Úvod	1
1 Formulácia úlohy	2
2 Prehľad literatúry	4
2.1 Optický princíp senzorov	4
2.2 Vyhýbanie sa prekážkam a orientácia na základe obrazu z kamery	5
2.2.1 S vylepšenou neurónovou sietou	5
2.2.2 Princíp stereosnímania	7
2.2.3 Na základe pohyblivého obrazu kamery	9
2.2.4 Určovanie polohy pomocou značiek	9
2.3 WebSocket	9
3 Definícia problému prenasledovania koristi a metódy jeho realizácie	10
3.1 Analýza problému	10
3.2 Riadenie koristi	11
3.2.1 Popis samotných blokov	12
3.3 Režimy navigácie	13
3.3.1 Režim detekcie koristi	13
3.3.2 Režim prenasledovania a navigácie	17
3.3.3 Režim znehybnenia (chytenia) koristi	20
3.4 Výber vhodnej metódy robotickej kooperácie	21
3.4.1 Centralizované vs. decentralizované	22
3.4.2 Systém spätej väzby(Closed-Loop Control System)	22
3.4.3 Deliberatívne vs. reaktívne riadenie	23
4 Simulačné experimenty	24
4.1 Prehľad a príprava prostredia	24
4.2 Ovládanie robota Koristi	26

4.3	Detekcia značiek	28
4.4	Navigácia pre systém viacerých robotov	29
4.4.1	Variácia uhlovej rýchlosťi virtuálnej štruktúry	30
4.4.2	Výsledky experimentu: 3 roboty s dosiahnuteľnou virtuálnou štruktúrou	32
4.5	Proces prenasledovania	34
5	Záver	38

Zoznam obrázkov

2–1	Načítaný obraz (vľavo), skutočná mapa hĺbky na základe senzorov (vľavo uprostred) a napokon mapa hĺbky vytvorená pomocou Gaussovoho modelu (vpravo uprostred) a Laplaceovho modelu (vpravo)	6
2–2	Faktory zohľadňované pri monokulárnom vnímaní hĺbky	7
3–1	Architektúra systému riadenia robotov.	12
3–2	Ilustrácia geometrie problému sledovania a navigácie.	13
3–3	Schéma modelovania obrázkov so značkami ArUco.	15
3–4	Určenie polohy robota vzhľadom na značku.	17
3–5	Určenie globálnej polohy robota pomocou polohy značky.	17
3–6	Režim detekcie koristi - schéma.	18
3–7	Režim sledovania a navigácie - schéma.	20
3–8	Režim zatváranie koristi - schéma.	21
3–9	Hybridná architektúra riadenia.	22
3–10	Systém spätej väzby.	23
4–1	Illustrované cesty, použitie na testovanie.	27
4–2	Závislosť vplyvu velkosti oblasti značky na chybe pri zisťovaní jej polohy.	28
4–3	Pravdepodobnosť detekcie značky v závislosti od jej oblasti na obrázku pre 3 rôzne uhly naklonenia: 0° , 45° , 70°	29
4–4	Nie sú vynútené nežiaduce kmity dráhy robota prechodnej fázy.	30
4–5	Nežiaduce oscilácie dráhy robota, ak nie sú splnené predpokladané obmedzenia dynamického ciela (ak $\theta_T > P$).	31
4–6	Trajektória robotov.	33
4–7	Zmena vzdialenosťi d_{S_i} medzi robotom i a zvoleným cielom (i = 1..3).	33
4–8	Výsledky simulácie	35

Zoznam tabuliek

4 – 1 Zoznam časovaní robota na pokrytie rôznych typov tratí	26
4 – 2 Výsledkom simulácie na ploche 100m * 100m vs 60m * 60m	35
4 – 3 Výsledky simulácie 3 prenasledovateľov v oblasti 100 m * 100 m	36

Slovník termínov

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

R_p je zápis na diagramoch a obrázkoch robota Korist

R₁, R₂, R₃ 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 приблизiť 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

Bezpilotné lietadlá (UAV) alebo drony, ako sú častejšie známe, sú čoraz bežnejšie, pričom jedným z dôvodov je ich miniaturizácia a nová kategória, ktorú vytvorili: MAV (Micro Aerial Vehicle). Kvadrokoptéry, ktoré sa dajú používať v interiéri, sa vďaka svojim malým rozmerom môžu ľahko presúvať medzi rôznymi typmi zariadení. V širokom spektri aplikácií, ako je vizuálny dohľad, monitorovanie alebo dokonca nahrávanie videa, poskytujú malé rozmery lepšiu manévrovateľnosť a možnosť lietať a skúmať menšie, ťažko dostupné oblasti.

Malá velkosť je však na úkor menšieho počtu snímačov, keďže hmotnosť je rozhodujúcim faktorom. Jediná zabudovaná kamera so správnym softvérom môže nahradiť mnohé snímače potrebné na vyhýbanie sa prekážkam, čo je dôležité pre autonómnu prevádzku. V súčasnosti ani palubné počítače kompaktných zariadení nemajú takýto výpočtový výkon, takže na spracovanie obrazu a odosielanie riadiacich údajov môžeme použiť externý počítač.

Cieľom tejto práce je ovládať a riadiť dron a viacero dronov pomocou webovej aplikácie a zverejňovať polohu dronu v priestore pomocou kamery a aruko tagov.

1 Formulácia úlohy

Analýza existujúcich prístupov a technológií pre využitie infraštruktúry internetu vozidiel v navigácii mobilných robotov. Táto časť bakalárskej práce bude obsahovať analýzu existujúcich prístupov a technológií pre využitie infraštruktúry internetu vozidiel v navigácii mobilných robotov. Tento bod sa bude venovať prehľadu existujúcich riešení a technológií, ktoré umožňujú mobilným robotom využívať internetové pripojenie vozidiel a ich infraštruktúru na navigáciu a pohyb.

Návrh architektúry systému pre využitie prostriedkov internetu vozidiel v navigácii mobilných robotov. V tejto časti sa budeme zaoberať návrhom architektúry systému pre využitie prostriedkov internetu vozidiel v navigácii mobilných robotov. V rámci tohto bodu sa bude riešiť návrh architektúry, ktorá umožní efektívne využívanie prostriedkov internetu vozidiel pre navigáciu mobilných robotov.

Implementácia navrhnutého systému a jeho testovanie v rôznych prostrediach a scenároch. Tato časť bude zahŕňať implementáciu navrhnutého systému a jeho testovanie v rôznych prostrediach a scenároch. Bude zahŕňať praktickú implementáciu navrhnutého systému a jeho otestovanie v rôznych reálnych prostrediach a podmienkach.

Overenie funkčnosti a spoľahlivosti systému na lietajúcich vozidlách. Táto kapitola sa bude zaoberať overením funkčnosti a spoľahlivosti systému na lietajúcich vozidlách. V rámci tohto bodu sa bude testovať funkčnosť a spoľahlivosť systému na lietajúcich vozidlách, aby sa zabezpečilo, že systém dokáže bezchybne fungovať aj v náročných podmienkach.

Porovnanie výkonu a efektivity navrhnutého systému s existujúcimi prístupmi a technológiami. Tento bod sa bude zaoberať porovnaním výkonu a efektivity navrhnutého systému s existujúcimi prístupmi a technológiami. Tento bod bude zahŕňať porovnanie navrhnutého systému s existujúcimi riešeniami a technológiami,

aby sa zistilo, ako dobre funguje navrhnutý systém v porovnaní s konkurenčnými riešeniami.

Vypracovať dokumentáciu podľa pokynov vedúceho záverečnej práce. Posledný bod sa bude zaoberať vypracovaním dokumentácie podľa pokynov vedúceho záverečnej práce. Tento bod bude zahŕňať vypracovanie dokumentácie, ktorá bude obsahovať kompletný popis navrhnutého systému, jeho implementáciu, testovanie a výsledky porovnania s existujúcimi riešeniami.

2 Prehľad literatúry

Pred výberom metódy optického polohovania a navigácie sme museli preskúmať rôzne možnosti. Pozreli sme sa na metódy, ktoré by mohli byť vhodné na navigáciu dronov. Skúmala sa ich použitelnosť pre mikrodrony.

2.1 Optický princíp senzorov

Snímače s optickým princípom sa bežne používajú v mobilnej robotike na navigáciu, lokalizáciu a mapovanie. Tieto senzory zachytávajú informácie z prostredia analýzou svetla a jeho interakcie s povrchmi, čím poskytujú bohaté údaje na presné určovanie polohy robota a plánovanie trajektórie. V tejto časti sa budeme zaoberať rôznymi typmi optických principiálnych snímačov a ich aplikáciami v mobilnej robotike.

Jedným z najčastejšie používaných optických snímačov je kamera, ktorá zachytáva obrázky a videá prostredia. Kamery poskytujú bohaté vizuálne informácie vrátane farby, textúry a tvaru, ktoré možno využiť na detekciu, sledovanie a rozpoznávanie objektov. V našej práci využívame kamery namontované na bezpilotných lietadlách Tello na zisťovanie polohy značiek Aruco a odhadovanie ich súradníc v 3D priestore.

Ďalším dôležitým optickým senzorom je senzor LIDAR (Light Detection and Ranging), ktorý vysiela laserové lúče a meria ich odraz na vytvorenie 3D mapy prostredia. Senzory LIDAR sa bežne používajú v autonómnych vozidlách na vyhýbanie sa prekážkam a mapovanie. Sú však drahé a vyžadujú vysoký výpočtový výkon, čo ich robí menej vhodnými pre malú mobilnú robotiku.

Okrem kamier a LIDAR-u patria medzi ďalšie typy snímačov na optickom princípe infračervené snímače, ultrazvukové snímače a snímače času letu. Infračervené senzory merajú odraz infračerveného svetla na detekciu objektov a meranie ich vzdialenosť, zatiaľ čo ultrazvukové senzory vysielajú vysokofrekvenčné zvukové vlny a merajú ich ozvenu na odhad vzdialenosť. Snímače času letu využívajú svetlo na

meranie času, za ktorý sa signál odrazí späť, čím poskytujú presné merania vzdialenosťi.

Senzory na optickom princípe celkovo zohrávajú kľúčovú úlohu v mobilnej robotike a poskytujú bohaté informácie na navigáciu a lokalizáciu robota. Pochopením rôznych typov senzorov a ich aplikácií môžeme navrhnúť a implementovať efektívne robotické systémy, ktoré môžu fungovať v rôznych prostrediach a scenároch.

2.2 Vyhýbanie sa prekážkam a orientácia na základe obrazu z kamery

Jedným z najdôležitejších úloh mobilných robotov je schopnosť vyhnúť sa prekážkam a správne sa orientovať v prostredí. K dispozícii sú rôzne senzory, ktoré umožňujú robotom rozpoznať a vyhnúť sa prekážkam. Okrem senzorov ako LIDAR, ultrazvukových a infračervených senzorov sa dá využiť aj obraz z kamery.

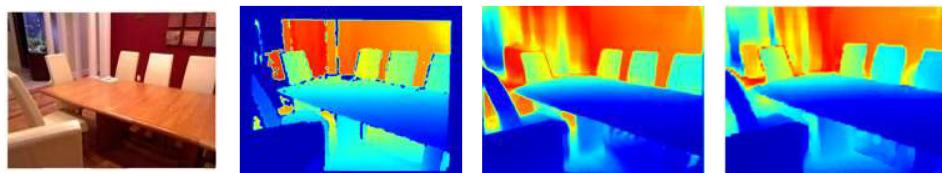
Kamera poskytuje bohaté vizuálne informácie o prostredí a umožňuje robotom získať informácie o prekážkach a teréne. Existujú rôzne spôsoby, ako využiť obraz z kamery pre navigáciu a vyhýbanie sa prekážkam.

2.2.1 S vylepšenou neurónovou sieťou

Jednou zo skúmaných metód na vyhýbanie sa prekážkam a orientáciu pomocou kamery bolo strojové učenie pomocou neurónových sietí [?]. Nakoniec však bola zavrhnutá kvôli časovo a hardvérovo náročným procesom učenia. Táto metóda zahŕňa použitie naučenej neurónovej siete na priradenie približných hodnôt hĺbky každému pixelu na základe polohy objektov z jedného obrazu. Vzory sa dajú naučiť z dvojice LIDAR a digitálnych kamier umiestnených vedľa seba. Zadaním dostatočného počtu týchto naučených dvojíc môže neurónová siet vytvoriť vzťah medzi pixelmi na obrázkoch a hodnotami hĺbky získanými z LIDAR-u alebo stereokamery.

Siet sa môže učiť pomocou učenia založeného na LIDAR-e s dohľadom a učenia založeného na stereopároch bez dohľadu na základe pravdepodobnostného princípu. Učenie pod dohľadom je však často príliš prísne a učenie bez dohľadu poskytuje nepresné výsledky. Preto sa odporúča používať poloprevádzkové učenie [?] alebo porovnávať výsledky získané pomocou siete s kalibrovaným systémom detekcie hĺbky počas prevádzky [?].

Táto metóda získava hodnoty hĺbky z okolia a umiestnenia objektov (množiny pixelov) a vytvára mapu hĺbky podobnú tej, ktorá je znázornená na obrázku 1. Jej presnosť je zatiaľ experimentálna a takýto systém je stále vystavený mnohým chybám. Napriek tomu bol použitý v aplikáciách na riadenie dronov [?]. Hĺbkové mapy na obrázku 1 ilustrujú rozdiel medzi rôznymi metódami a presnosť tejto metódy.



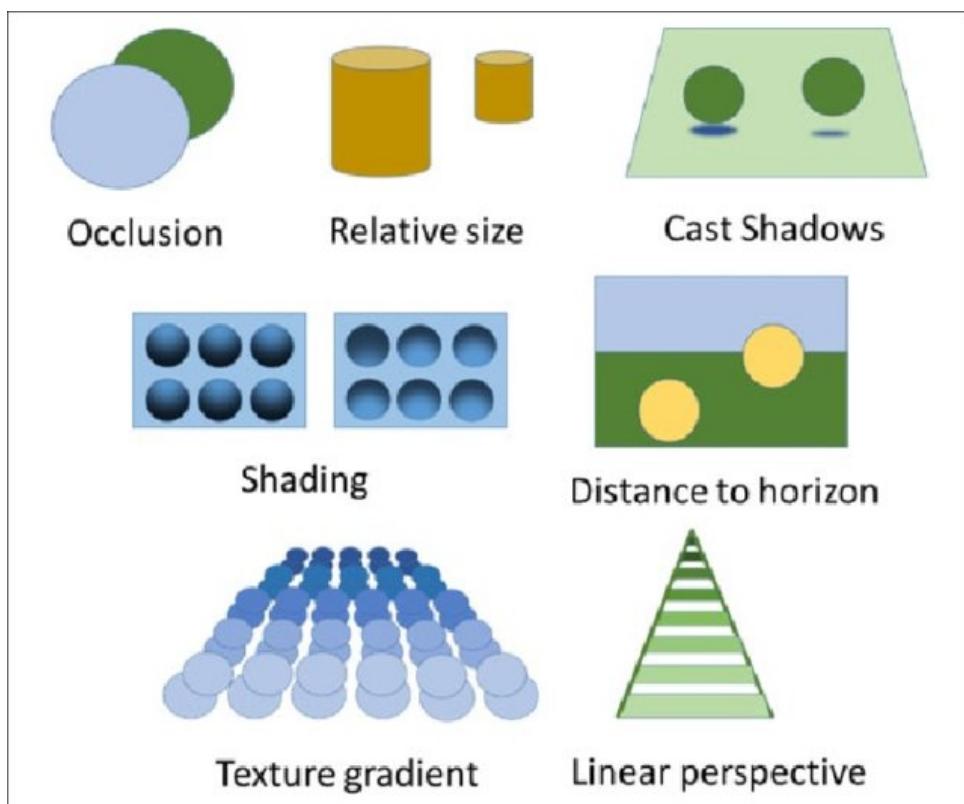
Obrázok 2–1 Načítaný obraz (vľavo), skutočná mapa hĺbky na základe senzorov (vľavo uprostred) a napokon mapa hĺbky vytvorená pomocou Gaussovoho modelu (vpravo uprostred) a Laplaceovho modelu (vpravo)

Neurónová siet berie do úvahy niekoľko faktorov:

- **oklúzia**
- **relatívna veľkosť**
- **vrhaný tieň**
- **tieňovanie**
- **vzdialenosť od horizontu**
- **gradient textúry**

- lineárna perspektíva

Učenie siete je časovo a zdrojovo náročné. Po vyškolení je výzvou zabezpečiť, aby sieť fungovala v reálnom čase na riadenie dronov bez toho, aby sa stratilo príliš veľa údajov v dôsledku zrýchlenia kalibrácie, čo môže viest ku kolíziám. Počas pomalého letu sa siet stále dokáže vyhýbať prekážkam, ale považuje ich len za súbor bodov s relatívnou polohou, nie za presné obrysy.



Obrázok 2 – 2 Faktory zohľadňované pri monokulárnom vnímaní hĺbky

2.2.2 Princíp stereosnímania

Princíp stereosnímania je široko používaná metóda vnímania hĺbky v počítačovom videní, robotike a príbuzných oblastiach. Zahŕňa použitie dvoch alebo viacerých kamier na snímanie obrazov tej istej scény z rôznych uhlov pohľadu a následné použitie rozdielov v obrazoch na výpočet vzdialostí objektov na scéne. Tento princíp sa

uplatňuje pri rôznych úlohách v robotike vrátane rozpoznávania objektov, navigácie a vyhýbania sa prekážkam.

V kontexte bezpilotných lietadiel môže byť princíp stereočočky užitočným nástrojom na navigáciu a vyhýbanie sa prekážkam. Populárny dron DJI Tello má však napríklad len jednu kameru. V takomto prípade možno obraz z kamery rozdeliť na dve virtuálne kamery, ktoré sa potom môžu použiť na simuláciu stereovidenia. Tento prístup má však svoje obmedzenia a výzvy.

Jednou z hlavných výziev je, že rozlíšenie virtuálnych kamier je nižšie ako rozlíšenie skutočnej kamery. Toto zníženie rozlíšenia môže ovplyvniť presnosť vnímania hĺbky, najmä v prípade vzdialených objektov. Ďalšou výzvou je, že základná vzdialenosť medzi dvoma virtuálnymi kamerami je pevná a nedá sa upraviť, čo môže obmedziť rozsah vzdialenosťí, ktoré možno presne odhadnúť.

Okrem toho výkonnosť princípu StereoLensing môžu ovplyvniť rôzne faktory, ako sú svetelné podmienky, kalibrácia kamery a oklúzie. Tieto faktory môžu mať za následok chyby vo vnímaní hĺbky, čo môže viesť ku kolíziám alebo nepresnej navigácii.

Vzhľadom na tieto problémy je pochopiteľné, prečo princíp StereoLensing nemusí byť vždy najlepším riešením pre navigáciu dronov a vyhýbanie sa prekážkam. Namiesto toho môžu byť na tieto úlohy vhodnejšie iné techniky, ako napríklad LiDAR, ultrazvukové senzory a vizuálno-inerciálna odometria, najmä v prostredí s náročnými svetelnými podmienkami alebo zložitými štruktúrami.

Na záver možno konštatovať, že hoci je princíp stereosnímania výkonnou metódou na vnímanie hĺbky a úspešne sa uplatňuje v rôznych oblastiach, jeho použitie v kontexte dronov, ako je Tello, je obmedzené z dôvodu použitia iba jednej kamery. V dôsledku toho môže byť potrebné použiť iné techniky snímania v spojení s týmto princípom, aby sa dosiahla presná a spoloahlivá navigácia dronov a vyhýbanie sa prekážkam.

2.2.3 Na základe pohyblivého obrazu kamery

2.2.4 Určovanie polohy pomocou značiek

2.3 WebSocket

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

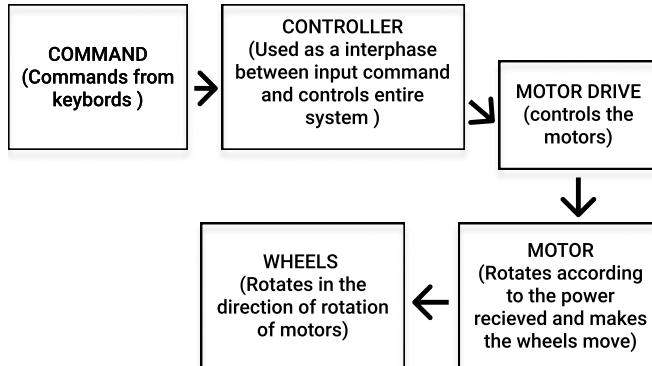
1. Každý robot má senzorický systém, ktorý umožňuje zisťovať prekážky a odhadnúť polohu koristi podľa robota.
2. Rýchlosťi robotov vyhovujú $v_i > v_p$, pre $i = 1, \dots, N$. Kde v_i - rýchlosť lovca a v_p - rýchlosť koristi.
3. Roboti a korist sa pohybujú konštantnou rýchlosťou.
4. Cesta, ktorú korist 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, dolava, doprava a stop. Tlačidlá hore a dole sa používajú na príkaz robota, aby sa pohyboval dopredu a dozadu. Ľavé tlačidlo slúži na ovládanie robota pri odbočovaní dolava a pravé tlačidlo pri odbočovaní doprava. Na zastavenie robota je možné použiť tlačidlo Stop.

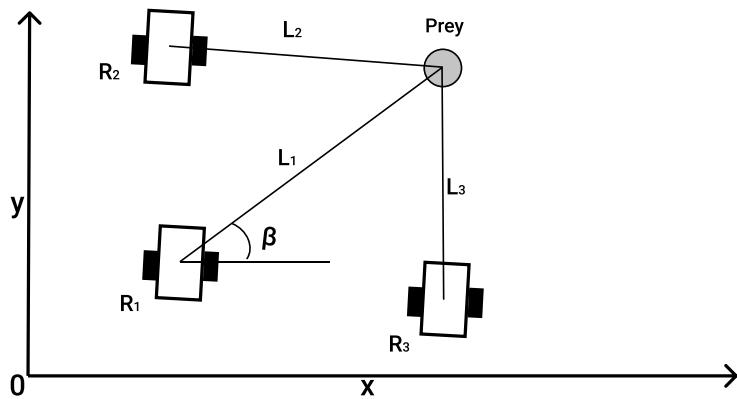
Blok radiča. Blok radiča je prepojený s rozhraním, dekóduje príkazy a vydáva riadiace signály do príslušných modulov, aby sa robot mohol pohybovať. Udržuje tiež časovanie a postupnosť generovania riadiacich signálov. Prijíma rozhodnutia na základe príkazov na riadenie pohybu robota v ľubovoľnom smere.

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 navaigačných režimoch nasledovne

1. Režim detekcie koristi.
2. Režim prenasledovania a navigácie: Počas tohto režimu je robot riadený na sledovanie koristi.
3. Režim znehýbnenia(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 navaigačné metódy alebo matematicky komplikovaná pravdepodobnostná metóda.

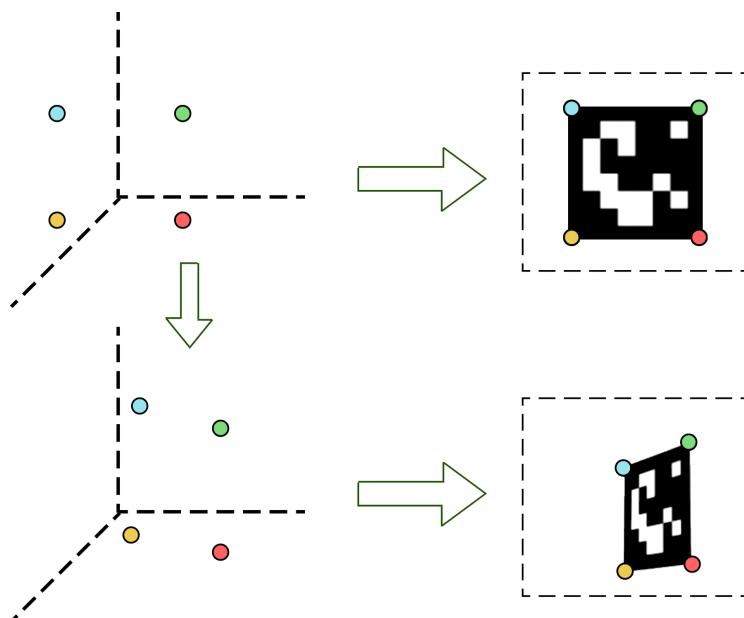
V súčasnosti má takmer každý novo vyvinutý mobilný robot zabudovaný kamerový systém. Preto je vhodné ho použiť na určenie polohy robota v prostredí. Vizuálna lokalizácia patrí do skupiny trigonometrických metód. Vizuálny systém môže obsahovať jednu alebo dve kamery a mal by zisťovať niektoré značky v prostredí pomocou nástrojov počítačového videnia. Tieto značky môžu byť umelé alebo prirodzené (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 počiaatočných súradníc rotačnej a translačnej matice.
6. Projekcia trojrozmerných bodov rohov značky do dvojrozmerných bodov na modelovanom obrázku pomocou parametrov kamery.
7. Výpočet funkcie transformácie perspektívnej 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 odhadovej polohy značky s vopred určenou polohou v kroku 4.
11. Opakovanie krokov 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 [1]. 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 funkciemi 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 θ je uhol medzi fotoaparátom a značkou (obrázok 3-4).

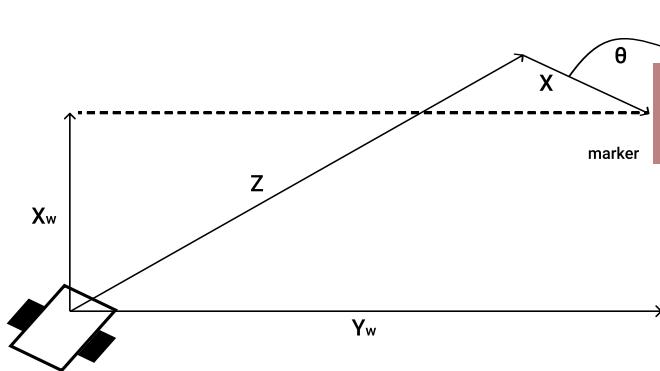
Globálna poloha robota x_w , y_w a θ_w podľa obrázku 3-5 je potom:

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

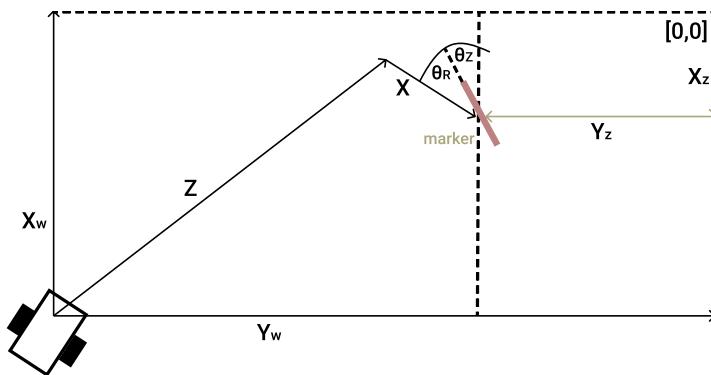
$$y_w = z.\cos(\theta_R + \theta_Z) - x.\sin(\theta_R + \theta_Z) + y_Z; \quad (3.6)$$

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

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



Obrázok 3 – 4 Určenie polohy robota vzhľadom na značku.

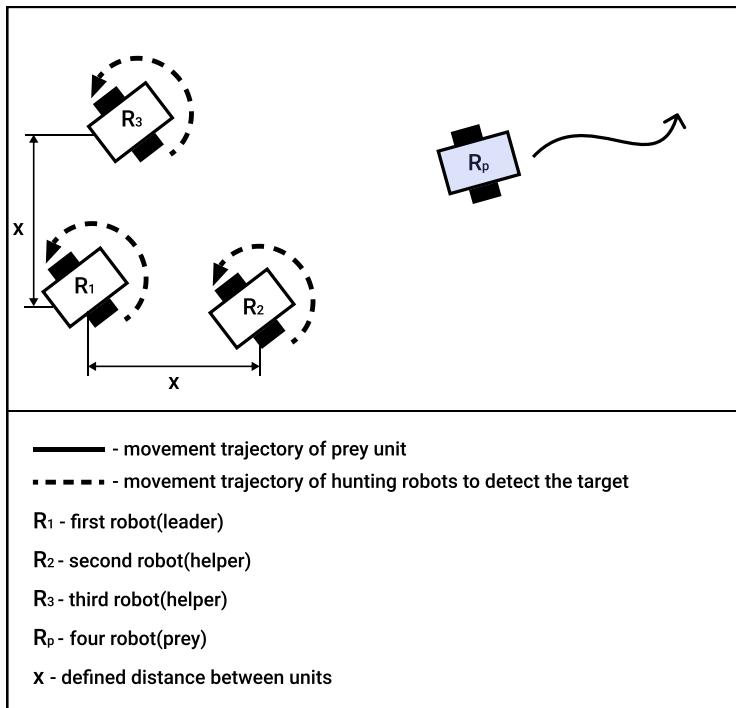


Obrázok 3 – 5 Určenie globálnej polohy robota pomocou polohy značky.

3.3.2 Režim prenasledovania a navigácie

Tento režim je pravdepodobne najdôležitejším režimom v rámci problému s lovom. Roboti začnú koristiť sledovať zo svojich počiatočných pozícii. Výpočet relatívnej vzdialenosť od cieľa 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 cieľu. Aby sme zaručili, že prenasledovateľ nenarazí na korist, zavedieme bezpečnú vzdialenosť r ($0 < r < R$), takže keď je vzdialenosť medzi prenasledovateľom a koristou menšia ako r , prenasledovateľ by mal spomaliť o δ , aby sa zväčšil ich



Obrázok 3 – 6 Režim detekcie koristi - schéma.

vzdialenosť rýchlo. Rýchlosť sa potom nastavuje odlišne podľa nasledujúcich troch podmienok:

- $d(P, A) > R$. V tejto situácii je vzdialosť od prenasledovateľa po korist veľmi veľká, čo sa často stáva v počiatočnej fáze sledovania, prenasledovateľ musí prenasledovať korist zrýchlením. Rýchlosť prenasledovateľa sa v každom kroku prudko zvýší o to, aby sa dosiahlo rýchle zmenšenie vzdialenosťi. Z dôvodu obmedzenia maximálnej zmeny rýchlosťi 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 korist 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 δ m/s, aby získal väčšiu

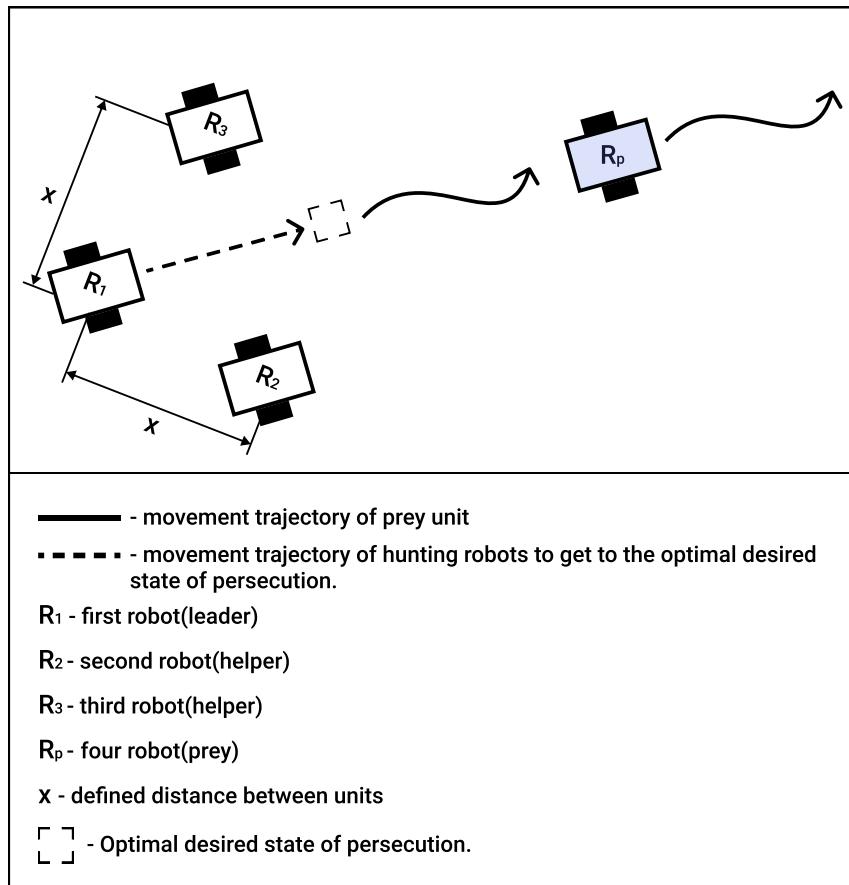
manévrovateľnosť z hľadiska bezpečnosti. Algoritmus 1 sumarizuje navrhovanú schému sledovania koristi-predátora.

V prírode môže lovčovi občas chýbať snímanie koristi v dôsledku oklúzie. Pri robotickom 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 ciel 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 ciel bude pokračovať v ceste rovnakým smerom a rovnakou rýchlosťou, keď nebudú môcť ciel vidieť. Tento predpoklad je ale užitočný iba v situáciach, keď pozorovanie chýba veľmi krátko. Ďalšou výhodou tejto metódy je, že nie je po-

trebné žiadne učenie, takže je výpočtovo efektívnejšia ako ostatné dve predikčné metódy.



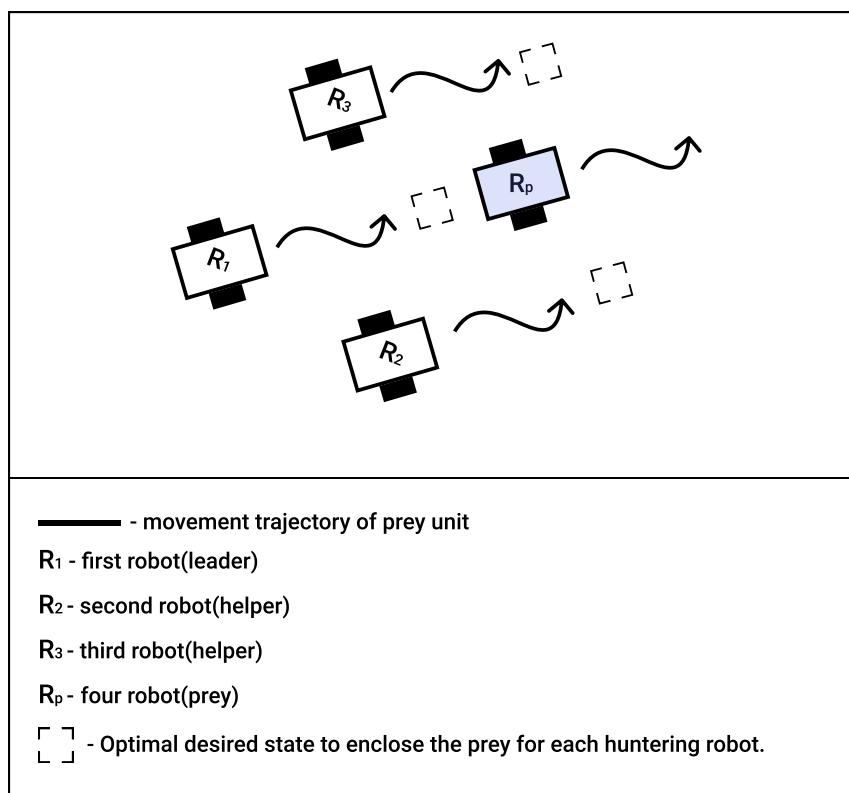
Obrázok 3 – 7 Režim sledovania a navigácia - schéma.

3.3.3 Režim znehybnenia (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 vzdialenosť l_0 sa počíta neskoro l_1 , vzdialenosť medzi dvoma po sebe nasledujúcimi robotmi vo formácii.

2. Keď roboti dosiahnu bod so vzdialenosťou l_0 od korisť, 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 korisť obklopená robotmi.



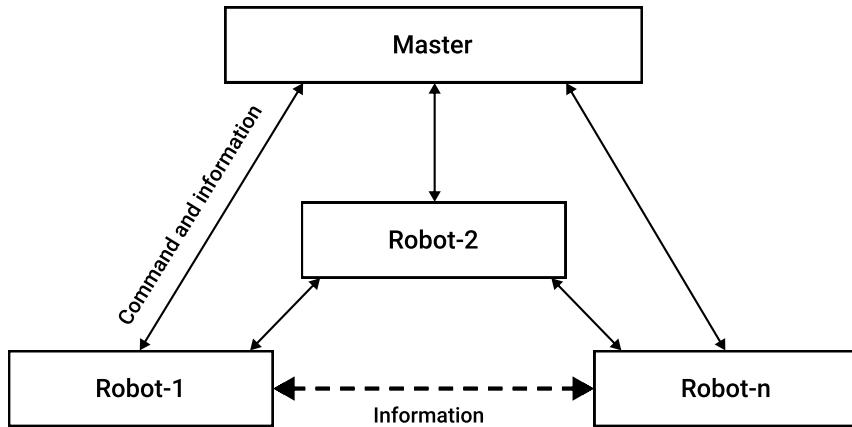
Obrázok 3 – 8 Režim zatváranie koristi - schéma.

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 pomocou 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 nastáť 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.

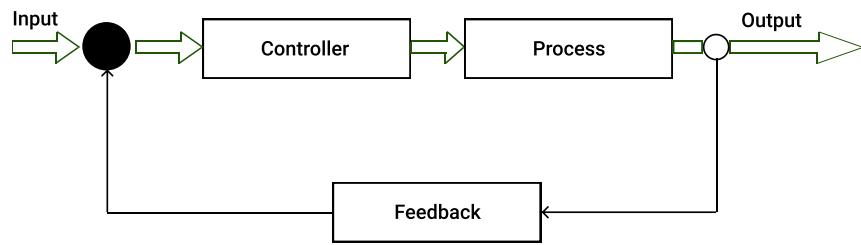


Obrázok 3 – 9 Hybridná architektúra riadenia.

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 spoľahlivosť výstupu,

protože 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 rýchlosťi 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 [2]. Webots je simulátor robotov s otvoreným zdrojovým kódom, ktorý poskytuje kompletné 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, le-

tectvo 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 fyzikálnych 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

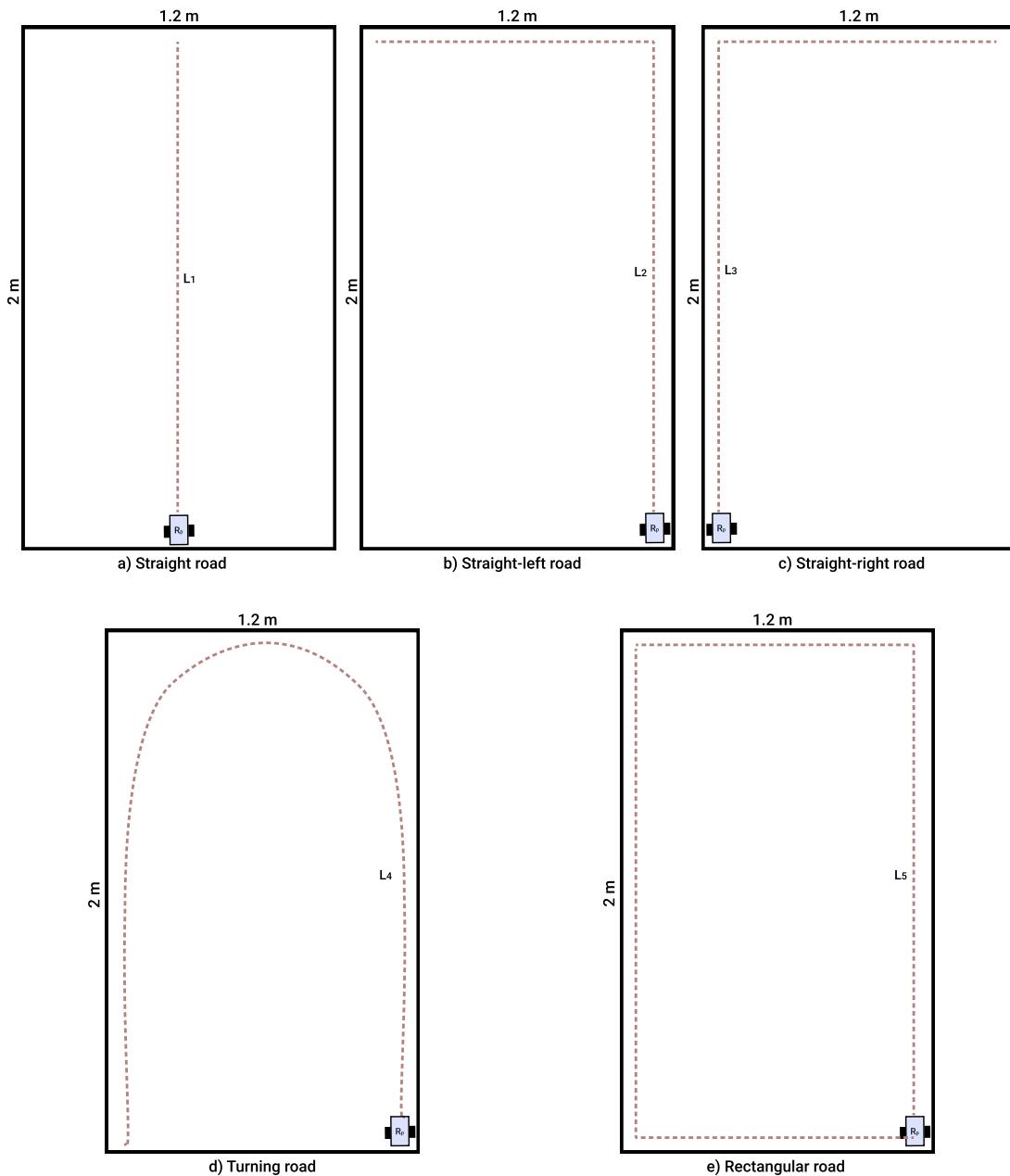
Pokus sa vykonáva na hladkej rovnej ploche. Testy sa uskutočňovali na simulátore Webots-e. Všetky testy sa uskutočňovali v rovnakej ploche. Vytvorili sme päť rôznych ciest, cez ktoré robot môže prechádzat: priamu, priamu-pravú, priamu-lavú, obrat a obdlžník. Na všetkých týchto cestách bolo pomocou klávesnice 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. V prí-

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

pade priamo-pravej cesty prejdená vzdialenosť je 2,04 m po priamke a 1,2 m vpravo.

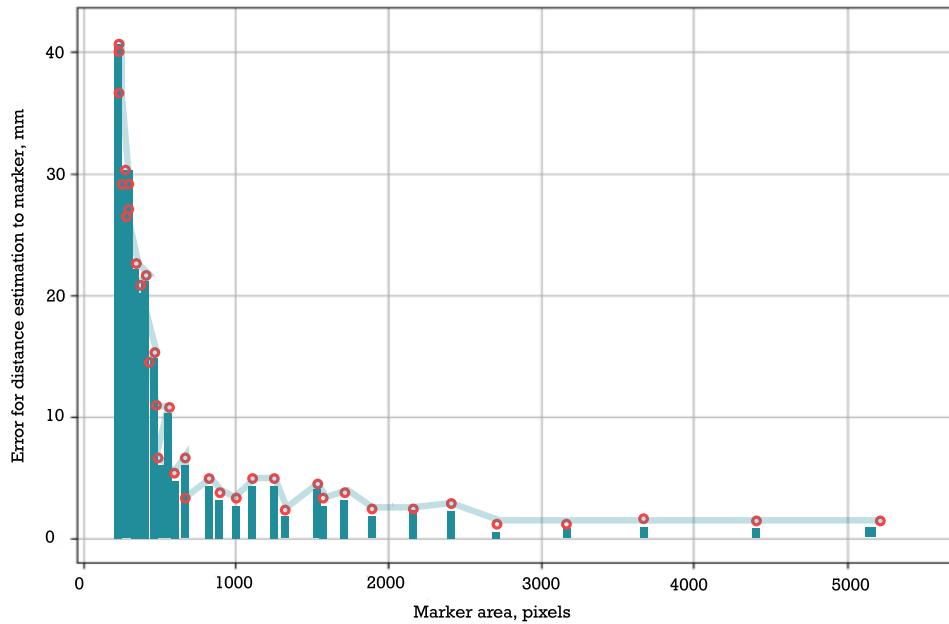
V prípade priamo-ľavej cesty je prekonaná vzdialenosť 2 m v priamke a 1,2 m vľavo. Stĺpec 4 zobrazuje časovanie cesty otočenia o U. Prejdená vzdialenosť je 2 m priamo hore a dole a šírka je 1,02 m v prípade obdlžníkového časovania cesty uvedeného v stĺpci 5



Obrázok 4 – 1 Ilustrované cesty, použitie na testovanie.

4.3 Detekcia značiek

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

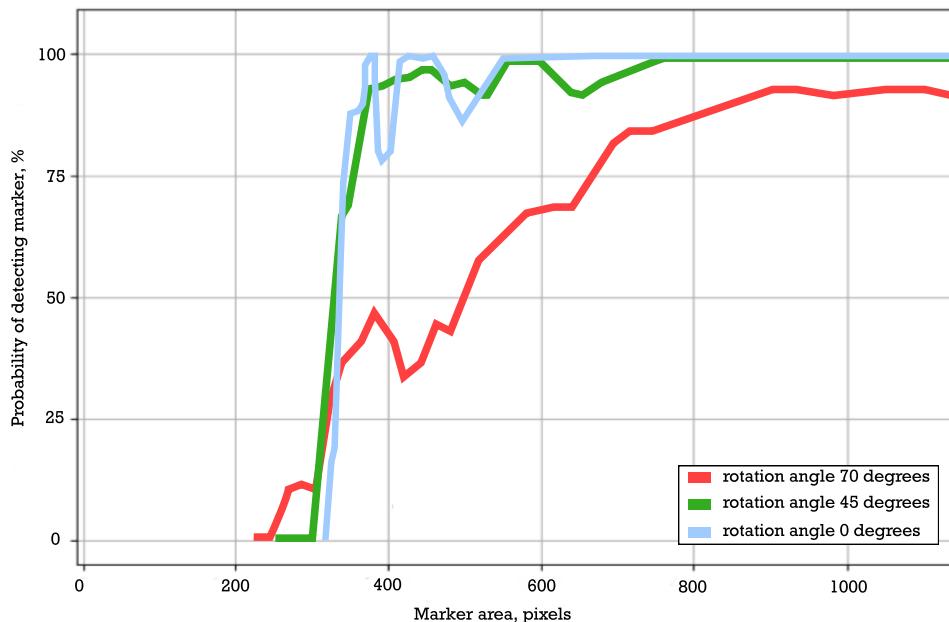


Obrázok 4 – 2 Závislosť vplyvu veľkosti oblasti značky na chybe pri zistovaní jej polohy.

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 líši od 100 v dvoch

oblastiach: s plochou 300 pixelov a s plochou v rozsahu 350-450 pixelov. Tento výsledok je dôsledne opakované pre rôzne série 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.



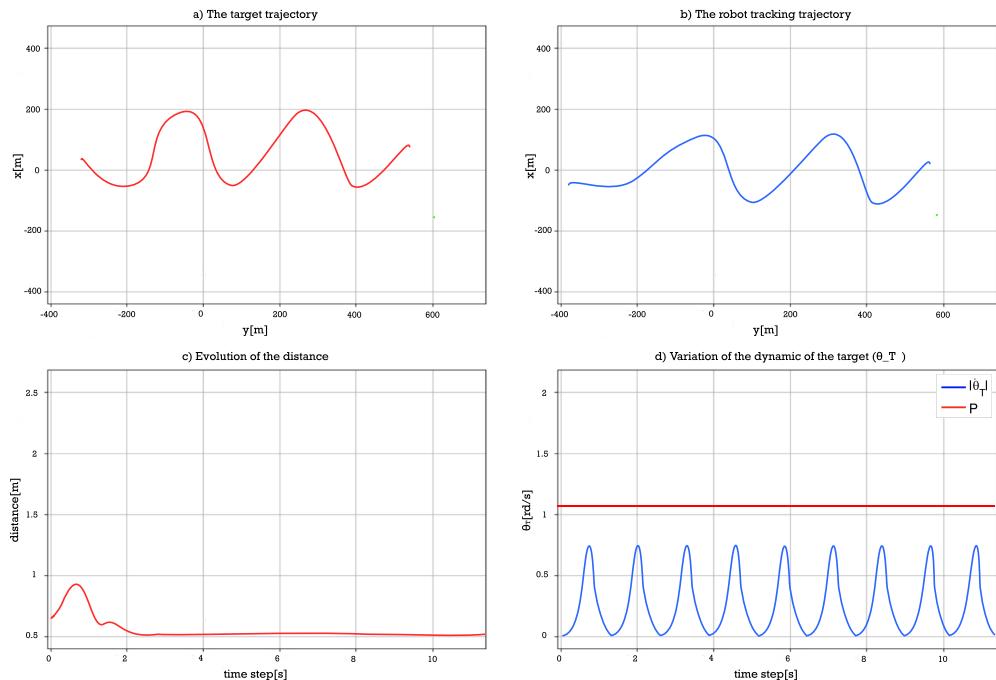
Obrázok 4 – 3 Pravdepodobnosť detekcie značky v závislosti od jej oblasti na obrázku pre 3 rôzne uhly naklonenia: 0°, 45°, 70°.

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 ohrianičenia uhlovej rýchlosťi virtuálnej štruktúry θ 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.

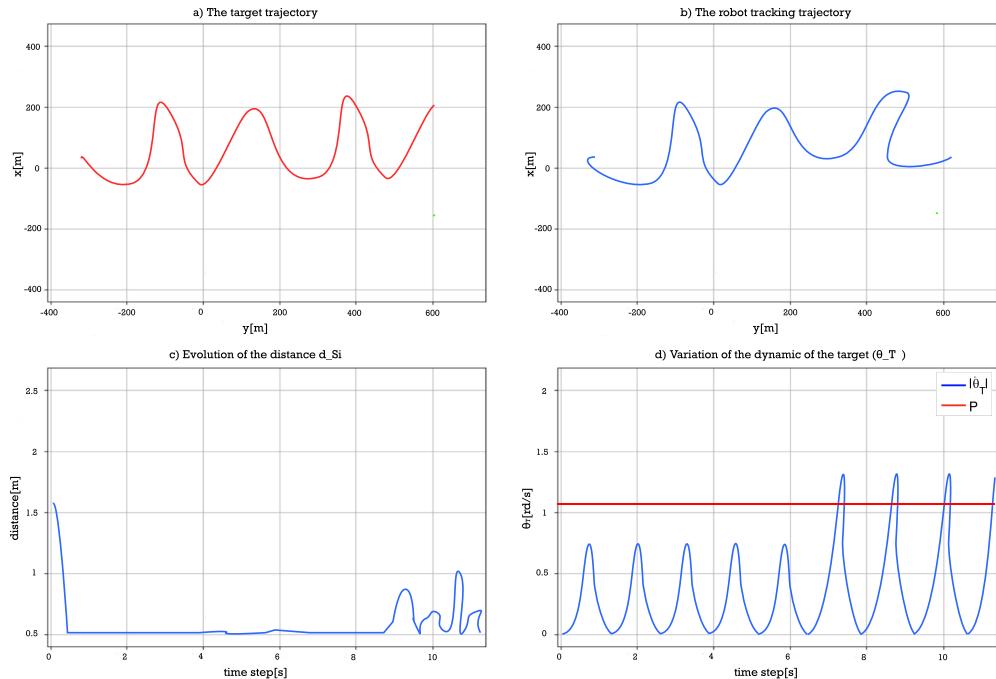
Podľa vzťahu a kvôli zjednodušeniu zápisu na číslach navrhujeme zaznamenať $P = \omega_{max} - k\pi$. Na základe zvolených hodnôt ω_{max} a k nájdeme $P = 1,1$. Najskôr sa navrhuje ukázať dôležitosť prechodnej fázy, pri ktorej musí byť variácia θ_T nastavená na 0. Preto teda na obrázku 4-4(d) vidíme, že θ_T sa zvyšuje na začiatku simulácie (od 0,1 s) a cieľová trajektória nasleduje okamžite po významnej krvke (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 θ_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 cielu. 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 θ_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 ciela (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.

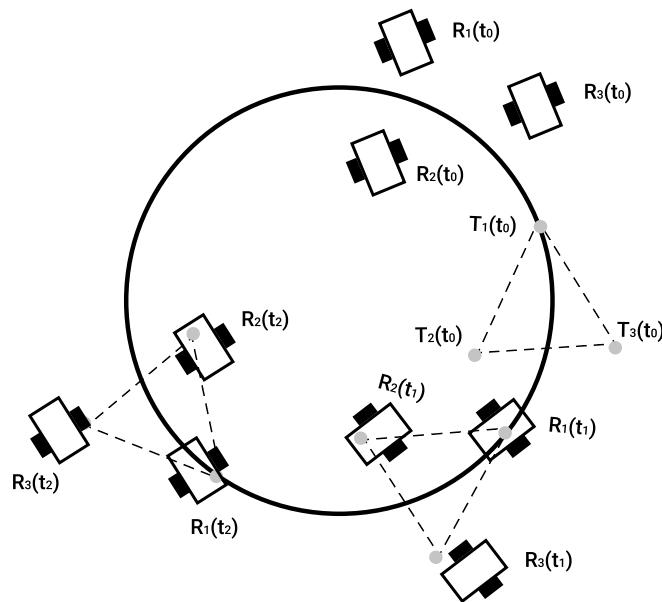
Virtuálna štruktúra má priamu trajektóriu. Tu sa navrhuje rozšíriť na kruhový pohyb tak, aby všetky ciele napriek ich kinematickým obmedzeniam zostali dosiahnuteľné všetkými robotmi. S vedomím, že dynamika virtuálnej štruktúry musí sledovať vzťah, bol polomer R_{vs} kruhového pohybu formovaný ako:

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

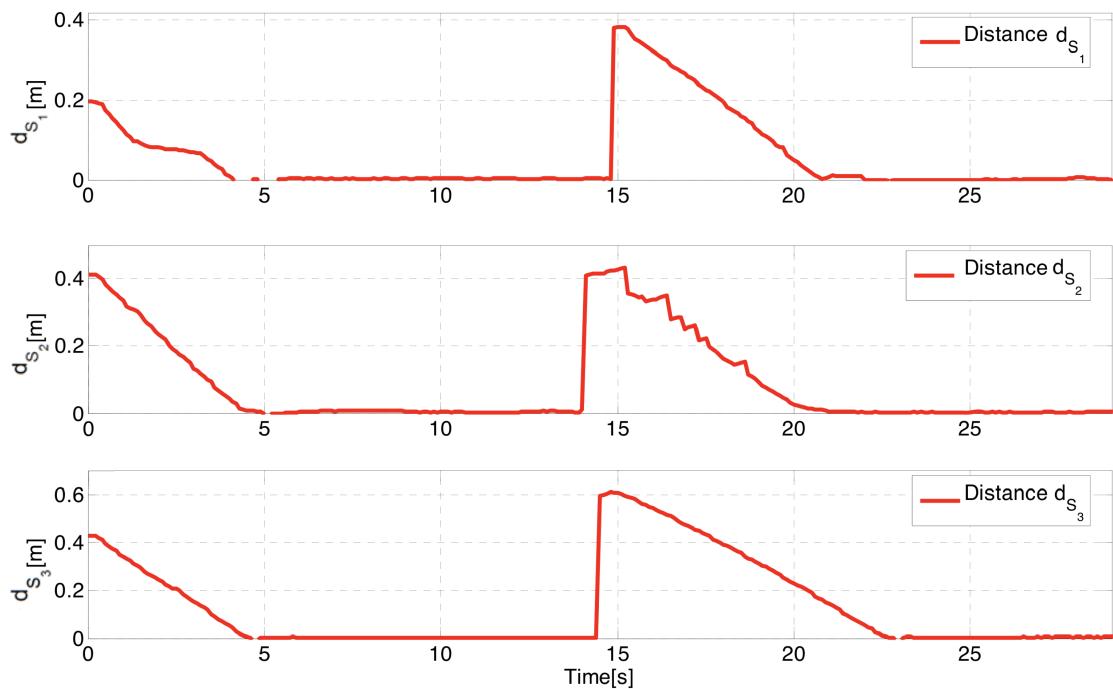
kde v_T je konštanta a $v_T \ll v_{max}$

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

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 lepší 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. Vzdialenosť medzi robotmi a ich cielmi 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 cielov, čo vysvetluje pozorované skoky.



Obrázok 4–6 Trajektória robotov.



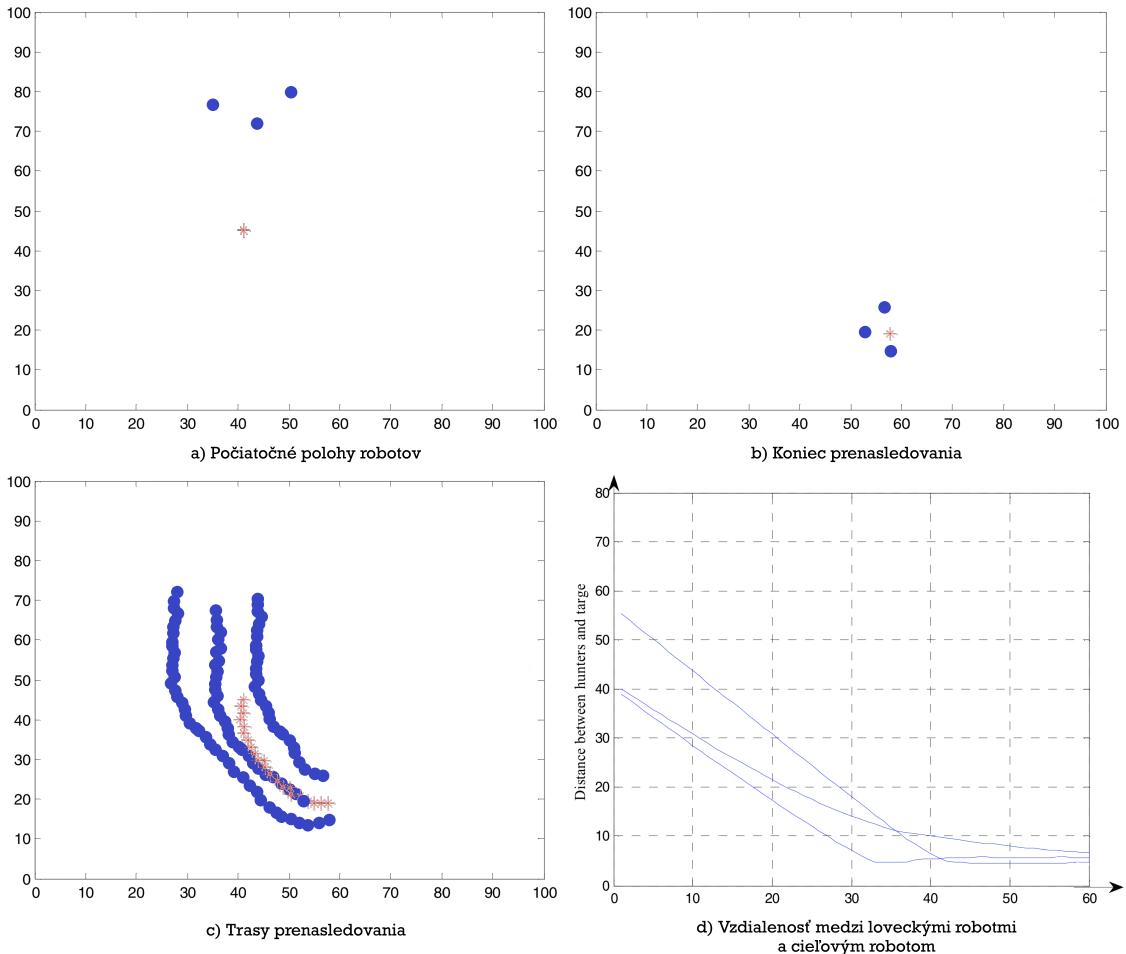
Obrázok 4–7 Zmena vzdialenosťi d_{S_i} medzi robotom i a zvoleným cieľom ($i = 1..3$).

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 opakovaní herného procesu, os Y predstavuje vzdialenosť medzi prenasledovacím robotom a cielom 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 cielu. Pri hľadaní potrebnej polohy sa roboti začnú skôr pohybovať s cielom, 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 pod-



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

mienkou vyššej rýchlosťi prenasledovateľov a náhodným generovaním informácií o 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 robo-

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

tov sú generované náhodne a maximálna rýchlosť loveckých robotov je stále vyššia 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 zvýší o 0,02 m/krok, úspešnosť lovú sa zvýší tiež na 20% a priemerný počet krokov k úspešnému lovú je 99 krokov. Keď sa rýchlosť zvýší na 0,1 m/krok, úspešnosť je až 100% a priemerný počet krokov je 77,48 kroku. Ak sa rýchlosť zvýší 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] 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.
- [2] “Webots documentation: introduction to webots.” <https://cyberbotics.com/doc/guide/introduction-to-webots>, 2021. Accessed: 2021-05-21.