



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH
DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA
I TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

Algorytmy sterowania zespołem robotów mobilnych i ich implementacja

Dawid Knapik

Promotor rozprawy: **prof. dr hab. inż. Andrzej Turnau**

Praca wykonana:
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie,
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej,
Katedra Automatyki i Robotyki

Kraków, 2024



AGH

AGH UNIVERSITY OF KRAKOW

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE AUTOMATION, ELECTRONICS, ELECTRICAL
ENGINEERING AND SPACE TECHNOLOGIES

DOCTORAL THESIS

Algorithms and Implementation for Cooperative Control of Multi-robot Systems

Dawid Knapik

Supervisor: **prof. dr hab. inż. Andrzej Turnau**

Completed in:

**AGH University of Krakow, Faculty of Electrical Engineering,
Automatics, Computer Science and Biomedical Engineering,
Department of Automatic Control and Robotics**

Krakow, 2024

Streszczenie

Tematem pracy są rozproszone algorytmy sterowania dla zespołu współpracujących robotów mobilnych. Rozważane przypadki dotyczą sytuacji, w której roboty nie mają dostępu do globalnego układu współrzędnych, a ich pomiary i sterowanie mają charakter lokalny.

W pracy przeprowadzono analizę sterowania bazującego na wirtualnych potencjałach pomiędzy sąsiadującymi robotami. Zdefiniowano algorytmy sterowania dla zespołu pojazdów mobilnych wykonujących różne zadania, takie jak osiągnięcie formacji czy podążanie za liderem. Opis poszczególnych zadań został scharakteryzowany przez grafy definiujące sposób wymiany informacji pomiędzy robotami. W celu weryfikacji algorytmów przeprowadzono analizę stabilności układu zamkniętego oraz wykonano eksperymenty symulacyjne dla obiektów o uproszczonej dynamice (punkty kinematyczne).

W pracy zaproponowano i opisano adaptację proponowanych algorytmów dla robotów o złożonej kinematyce i dynamice. Wykorzystano w tym celu lokalne pętle sterowania oparte na regulatorach od stanu oraz przekształcenie sterowania obliczonego dla punktów kinematycznych na kinematykę nieholonomicznych pojazdów. Przedstawiono wyniki eksperymentalne i symulacyjne dla różnych typów robotów mobilnych w tym dla niestabilnych pojazdów dwukołowych. Opisano metodologię projektowania układu sterowania bazującą na modelowaniu matematycznym. Zawarto model niestabilnego pojazdu dwukołowego oraz przeprowadzono identyfikację jego parametrów fizycznych. Praca zawiera również opis platformy programowo-sprzętowej, która umożliwiła prowadzenie badań eksperymentalnych.

Uzyskane wyniki potwierdzają, że możliwe jest opracowanie rozproszonych algorytmów sterowania dla zespołu współpracujących robotów o złożonej kinematyce i dynamice (np. nieholonomiczne, niestabilne roboty mobilne). Praca wskazuje również na możliwe dalsze kierunki badań oraz charakteryzuje potencjalne zastosowania opracowanych algorytmów.

Abstract

The subject of the dissertation are distributed control algorithms for a team of cooperating mobile robots. The considered cases involve situations in which the robots do not have access to a global coordinate system and their measurements and control are local.

The study conducted an analysis of control based on virtual potentials between neighbouring robots. Control algorithms were defined for a team of mobile vehicles performing various tasks such as formation maintenance and leader following. The description of individual tasks was characterised by graphs that define the method of information exchange between robots. To verify the algorithms, an analysis of the closed system stability was performed, and simulation experiments were conducted for objects with simplified dynamics (kinematic points).

The dissertation proposes and describes the adaptation of the suggested algorithms for robots with complex kinematics and dynamics. This involved the use of local control loops based on state regulators and the transformation of control computed for kinematic points to the kinematics of nonholonomic vehicles. Experimental and simulation results were presented on various types of mobile robots, including unstable two-wheeled vehicles. The model based design methodology for obtaining the control system was described. The model of an unstable two-wheeled vehicle is included and identification of its physical parameters procedure is described. The dissertation also contains a description of the test platform that enabled experimental research.

The results confirmed the feasibility of developing distributed control algorithms for a team of cooperating robots with complex kinematics and dynamics (e.g., nonholonomic, unstable mobile robots). This work also indicates possible further directions for research and shows the potential applications of the developed algorithms.

Autor składa serdeczne podziękowania:

Panu profesorowi Andrzejowi Turnau, promotorowi rozprawy, za wsparcie na wszystkich etapach kariery naukowej, sugestie i poprawki dotyczące tej pracy oraz niewyczerpaną cierpliwość, którą wielokrotnie testowałem.

Żonie Joannie oraz rodzicom za nieustającą wiarę i wsparcie.

Kolegom, w szczególności Maciejowi Klemiato, Andrzejowi Tutajowi, Pawłowi Zagórskiemu, Krzysztofowi Kołkowi, Maciejowi Rosołowi, Pawłowi Rotterowi, Łukaszowi Kuśmierzowi, Piotrowi Bani oraz Tomkowi Knapikowi, za uwagi merytoryczne, które pozytywnie wpłynęły na kształt pracy oraz inspirujące dyskusje i okazaną pomoc.

Spis treści

Spis rysunków	vii
Spis tabel	viii
Spis skrótów	ix
Oznaczenia	x
1 Wstęp	1
1.1 Cele pracy	1
1.2 Przegląd literatury	1
1.2.1 Wprowadzenie	1
1.2.2 Początki opisu formalnego	3
1.2.3 Sterowanie formacją robotów	4
1.2.4 Zastosowania	8
1.2.5 Istotne, zbliżone obszary badawcze	10
1.3 Stosowana terminologia	11
1.4 Układ pracy	11
2 Informacje wprowadzające	13
2.1 Podstawowe pojęcia i oznaczenia	13
2.2 Teoria sterowania	13
2.3 Opis matematyczny zespołu robotów	19
3 Algorytmy sterowania zespołem robotów - przykłady wprowadzające	23
3.1 Przykłady wprowadzające	23
3.1.1 Przykład 1	23
3.1.2 Przykład 2, algorytm cyklicznej pogoni dla trzech pojazdów	29
3.1.3 Przykład 3, formacja bazująca na wzajemnym położeniu	40
3.2 Podsumowanie	43
4 Algorytmy sterowania zespołem robotów nieholonomicznych	44
4.1 Robotarium	44
4.2 Przekształcenie sterowania dla robota klasy $(2,0)$	45
4.3 Metodologia badań	47
4.4 Eksperymenty	48

4.4.1	Przykład wprowadzający – algorytm osiągnięcia zadanej odległości przez dwa roboty	48
4.4.2	Algorytm cyklicznej pogoni dla formacji trzech robotów	53
4.4.3	Algorytm śledzenia lidera dla trzech robotów	55
4.4.4	Algorytm śledzenie lidera, przypadek z niestacjonarnymi wagami	57
4.4.5	Algorytm sterowania formacją pięciu robotów z dwukierunkową wymianą informacji	60
4.5	Podsumowanie wyników	65
5	Algorytmy sterowania zespołem robotów niestabilnych	66
5.1	Dwukołowy, niestabilny robot mobilny	68
5.1.1	Platforma sprzętowa	69
5.1.2	Rozwiązania programowe	72
5.1.3	Model matematyczny	74
5.1.4	Identyfikacja parametrów fizycznych modelu	84
5.1.5	Weryfikacja modelu matematycznego	86
5.1.6	Odtwarzanie stanu	87
5.1.7	Regulator utrzymujący pionową postawę i śledzący sygnały referencyjne prędkości liniowej i kątowej	93
5.2	Ogólna postać algorytmu sterującego zespołem niestabilnych robotów mobilnych	98
5.3	Eksperymenty symulacyjne	101
5.3.1	Metodologia badań symulacyjnych	101
5.3.2	Przykład wprowadzający	102
5.3.3	Algorytm cyklicznej pogoni	105
5.3.4	Algorytm śledzenia stacjonarnego lidera	108
5.3.5	Algorytm śledzenia lidera poruszającego się wzdłuż prostej, przypadek niestacjonarnych, zadanych odległości	113
5.3.6	Algorytm śledzenia ruchomego lidera poruszającego się wzdłuż łuku, przypadek niestacjonarnych, zadanych odległości	117
5.3.7	Algorytm sterowania formacją dla pięciu robotów	119
5.3.8	Podsumowanie wyników	125
5.4	Eksperymenty z pojazdem fizycznym	126
5.4.1	Metodologia badań na rzeczywistym robocie	126
5.4.2	Algorytm śledzenia stacjonarnego lidera	127
5.4.3	Algorytm śledzenia lidera poruszającego się wzdłuż prostej	129
5.4.4	Podsumowanie wyników	132

6 Podsumowanie	133
6.1 Kierunki dalszych badań	134
6.2 Wnioski	135
Załączniki	136
Literatura	143

Spis rysunków

1.1	Przykłady kolektywnego zachowania w świecie zwierząt	2
1.2	Pokaz świetlny złożony z roju dronów (źródło: [Intel])	5
1.3	Zależność pomiędzy lokalnymi układami współrzędnych robotów a globalnym	6
1.4	Wizja satelitarnego systemu interferometrii - projekt Terrestrial Planet Finder (źródło: [NASA], domena publiczna)	9
2.1	Schemat blokowy układu regulacji z regulatorem od stanu	19
2.2	Przykład grafu (prostego)	19
2.3	Przykład grafu skierowanego	20
2.4	Przykład grafu ważonego	20
2.5	Strukturalna sztywność grafu	22
3.1	Zależność pomiędzy lokalnymi układami współrzędnych robotów a globalnym, zewnętrznym układem współrzędnych	24
3.2	Graf reprezentujący zadanie utrzymania stałej odległości przez dwa pojazdy.	25
3.3	Ruch robotów zaznaczony na płaszczyźnie XY	27
3.4	Przebieg położenia oraz wzajemnej odległości robotów 1 i 2	28
3.5	Ruch robotów zaznaczony na płaszczyźnie XY	28
3.6	Przebieg położenia oraz wzajemnej odległości robotów 1 i 2	29
3.7	Zależność pomiędzy lokalnymi układami współrzędnych robotów, a globalnym, zewnętrznym układem współrzędnych	30
3.8	Graf przedstawiający wymianę informacji w algorytmie cyklicznej pogoni	30
3.9	Ruch robotów zaznaczony na płaszczyźnie XY	36
3.10	Przebieg położenia oraz uchyb odległości robotów 1, 2 i 3	37
3.11	Ruch robotów zaznaczony na płaszczyźnie XY	38
3.12	Przebieg położenia oraz uchyb odległości robotów 1, 2 i 3	38
3.13	Ruch robotów zaznaczony na płaszczyźnie XY	39
3.14	Przebieg położenia oraz uchyb odległości robotów 1, 2 i 3	39
3.15	Zależność pomiędzy lokalnymi układami współrzędnych robotów, a globalnym, zewnętrznym układem współrzędnych	40
3.16	Ruch robotów zaznaczony na płaszczyźnie XY	42
3.17	Przebieg położenia robotów 1, 2 i 3	43
4.1	Robotarium - laboratorium robotów mobilnych o zdalnym dostępie	44
4.2	GRITSBot X - robot używany w laboratorium Robotarium	45

4.3	Oznaczenia opisujące położenie i sterowania robota klasy (2,0). Zadana prędkość w ruchu liniowym została oznaczona przez v , a zadana prędkość w ruchu obrotowym przez ω	46
4.4	Ruch robotów zaznaczony na płaszczyźnie XY	49
4.5	Przebieg położenia, uchybu odległości i orientacji robotów 1 i 2	50
4.6	Migawki z nagrania rzeczywistego eksperymentu dla początku eksperymentu (lewa górna), czwartej (prawa górna), ósmej (lewa dolna) i dwunastej sekundy (prawa dolna)	51
4.7	Ruch robotów zaznaczony na płaszczyźnie XY	51
4.8	Przebieg położenia, wzajemnej odległości i orientacji robotów 1 i 2 . .	52
4.9	Digraf opisujący wymianę informacji w przypadku trzech robotów ...	53
4.10	Ruch robotów zaznaczony na płaszczyźnie XY	54
4.11	Przebieg położenia, uchybu wzajemnej odległości i orientacji robotów 1, 2 i 3	55
4.12	Digraf opisujący wymianę informacji w przypadku trzech robotów ...	55
4.13	Ruch robotów zaznaczony na płaszczyźnie XY	56
4.14	Przebieg położenia, uchybu wzajemnej odległości i orientacji robotów 1, 2 i 3	57
4.15	Zależność pomiędzy kształtem formacji a prędkością lidera	58
4.16	Ruch robotów zaznaczony na płaszczyźnie XY	59
4.17	Przebieg położenia, uchybu wzajemnej odległości, zadanych odległości i orientacji robotów 1, 2 i 3	60
4.18	Graf opisujący wymianę informacji w przypadku pięciu robotów ...	61
4.19	Ruch robotów zaznaczony na płaszczyźnie XY	62
4.20	Osiągana formacja w przypadku warunków początkowych (4.15) . . .	63
4.21	Osiągana formacja w przypadku warunków początkowych (4.16) . . .	63
4.22	Przebieg położenia, uchybu wzajemnej odległości, sumy wartości bezwzględnej uchybów i orientacji robotów 1-5	64
4.23	Migawki z nagrania rzeczywistego eksperymentu	65
5.1	Autonomiczne pojazdy dwukołowe	67
5.2	Fotografia dwukołowego, niestabilnego robota mobilnego	68
5.3	Dwukołowy robot mobilny - schemat	70
5.4	Architektura karty RT-DAQ/Zynq (na podstawie [29])	71
5.5	Wydażność układu sterowania	72
5.6	Metodologia projektowania algorytmów na podstawie modelowania matematycznego	73

5.7	Diagram blokowy programu Simulink realizujący sterowanie pojedynczym robotem	74
5.8	Dwukołowy robot mobilny - oznaczenia. Po lewej: widok z boku, po prawej: widok z góry	75
5.9	Wyznaczenie środka ciężkości	85
5.10	Wyznaczenie momentu bezwładności	85
5.11	Układ współrzędnych czujnika bezwładnościowego	88
5.12	Wyznaczenie kąta pochylenia robota na podstawie pomiaru z akcelerometrów	89
5.13	Schemat blokowy filtra Kalmana	90
5.14	Porównanie odtwarzania pochylenia robota	92
5.15	Porównanie odtwarzania kąta wahadła	92
5.16	Porównanie odtwarzania prędkości kątowej wahadła	93
5.17	Schemat blokowy regulatora od rozszerzonego stanu	94
5.18	Przebieg wybranych elementów stanu robota w zadaniu stabilizacji .	96
5.19	Przebieg wybranych elementów stanu robota w zadaniu nadążaniu za sygnałem referencyjnym	97
5.20	Porównanie trajektorii ruchu modelu symulacyjnego robota oraz rzeczywistego przebiegu	98
5.21	Schemat blokowy algorytmu sterującego formacją trzech robotów . .	99
5.22	Schemat blokowy wewnętrznej pętli sprzężenia zwrotnego oraz modelu robota	100
5.23	Schemat blokowy modułu obliczającego położenie robota na płaszczyźnie	100
5.24	Schemat blokowy zewnętrznej pętli sprzężenia zwrotnego dla pojedynczego robota	100
5.25	Diagram stanu dla eksperymentu symulacyjnego opisującego przypadek sterownia formacją pięciu robotów	102
5.26	Formacja dwóch robotów	103
5.27	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	104
5.28	Migawki z wizualizacji eksperymentu symulacyjnego	105
5.29	Digraf opisujący wymianę informacji w przypadku trzech robotów ...	105
5.30	Formacja robotów w algorytmie cyklicznej pogoni	107
5.31	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	108
5.32	Digraf opisujący wymianę informacji w przypadku trzech robotów ...	109

5.33	Formacja robotów w algorytmie śledzenia lidera, przypadek nieruchomego lidera	110
5.34	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	111
5.35	Formacja robotów w algorytmie śledzenia lidera, przypadek nieruchomego lidera	112
5.36	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	113
5.37	Digraf opisujący wymianę informacji w przypadku trzech robotów ...	114
5.38	Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej	115
5.39	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	116
5.40	Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej	118
5.41	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	119
5.42	Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej	121
5.43	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	122
5.44	Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej	123
5.45	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	124
5.46	Migawki z wizualizacji eksperymentu symulacyjnego	125
5.47	Formacja robotów w algorytmie śledzenia lidera, przypadek nieruchomego lidera	128
5.48	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	129
5.49	Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej	131
5.50	Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów	132

Spis tabel

1.1	Zestawienie publikacji dotyczących utrzymania formacji robotów . . .	7
5.1	Zestawienie parametrów fizycznych dwukołowego robota	86
6.1	Wykaz nagrań przeprowadzonych eksperymentów na rzeczywistych robotach	141
6.2	Wykaz wizualizacji eksperymentów przeprowadzonych na modelach niestabilnych pojazdów mobilnych	142

Spis skrótów

AS — asymptotyczna stabilność,

ARE — *Algebraic Riccati Equation*, algebraiczne równanie Riccati’ego

DAC — *Data Acquisition and Control*, (karta) pomiarowo – sterująca

EXS — stabilność wykładnicza

FPGA — *Field – Programmable Gate Array*, programowalny układ logiczny

GAS — globalna asymptotyczna stabilność,

GNSS — *Global Navigation Satellite System*, system nawigacji satelitarnej

GPS — *Global Positioning System*, jeden z pięciu systemów nawigacji satelitarnej
(wg stanu z 2024 roku)

IMU — *Inertial Measurement Unit*, czujnik bezwładnościowy (akcelerometr + żyroskop)

MAS — *Multi-agent system*, system wieloagentowy

MBD — *Model Based Design*

MEMS — *Micro – ElectroMechanical System*, mikroukład elektromechaniczny

LQR — *Linear – Quadratic Regulator*, regulator liniowo – kwadratowy

PWM — *Pulse Width Modulation*, sygnał o modulowanym wypełnieniu

ROS — *Robot Operating System*, zestaw oprogramowania przeznaczony do robotyki

RT — *Real – Time*, czas rzeczywisty

UART — *Universal Asynchronous Receiver – Transmitter*, uniwersalny asynchroniczny nadajnik – odbiornik

UAV — *Unmanned Aerial Vehicle*, bezzałogowy statek powietrzny

V2X — *Vehicle-to-Everything*, klasa komunikacja w autonomicznych pojazdach

Oznaczenia

- \mathbb{R} — zbiór liczb rzeczywistych
- \mathbb{N} — zbiór liczb naturalnych
- \mathbb{Z} — zbiór liczb całkowitych
- \mathbb{C} — zbiór liczb zespolonych
- \mathbb{R}^n — przestrzeń euklidesowa o wymiarze n
- $\|\cdot\|$ — norma euklidesowa
- \mathbb{I}_n — macierz jednostkowa $n \times n$
- \mathbb{J} — jacobian
- M^T — macierz transponowana
- M^{-1} — macierz odwrotna
- $\text{rz } M$ — rząd macierzy
- $\det M$ — wyznacznik macierzy
- $\sigma(M)$ — widmo macierzy
- $M = M^T, \quad M > 0$ — macierz (symetryczna) dodatnio określona
- $M = M^T, \quad M \geq 0$ — macierz (symetryczna) półdodatnio określona
- $x \in \mathbb{R}^n$ — wektor n elementowy, zwykle wektor stanu
- $u \in \mathbb{R}^n$ — wektor sterowania
- $A \in \mathbb{R}^{n \times n}$ — macierz stanu systemu dynamicznego
- $B \in \mathbb{R}^{n \times m}$ — macierz wejść systemu dynamicznego
- $C \in \mathbb{R}^{p \times n}$ — macierz wyjść systemu dynamicznego
- q_i — położenie robota na płaszczyźnie w lokalnym układzie współrzędnych,
 $q_i \in \mathbb{R}^2$
- p_i — położenie robota na płaszczyźnie w globalnym układzie współrzędnych,
 $p_i \in \mathbb{R}^2$

- V — funkcja Lapunowa
- \mathcal{G} — graf nieskierowany
- \mathcal{D} — graf skierowany (digraf)
- $L(\mathcal{G})$ — laplasjan grafu zwykłego \mathcal{G}
- $\Delta(\mathcal{G})$ lub $\Delta(\mathcal{D})$ - macierz stopni grafu
- $A(\mathcal{G})$ — macierz sąsiedztwa grafu zwykłego
- $A(\mathcal{D})$ — macierz sąsiedztwa digrafu (wejściowa)
- $L(\mathcal{D})$ — laplasjan (wejściowy) digrafu \mathcal{D}

1 Wstęp

Rozproszone algorytmy sterowania zespołem współpracujących ze sobą robotów mobilnych stanowią główny temat badawczy dysertacji. Robotami steruje się lokalnie i mierzy się ich lokalne położenia, bez dostępu do globalnego układu współrzędnych, a więc bez użycia centralnej jednostki sterującej. W istocie bada się możliwości sterowania grupą robotów, a więc formacją, którą tworzy się, a następnie steruje jej ruchem budując różnorodne algorytmy sterowania. Ruch formacji jest badany zarówno symulacyjnie, jak i praktycznie, na rzeczywistych robotach mobilnych sterowanych w czasie rzeczywistym. Typowe zadania obejmują osiągnięcie formacji i jej sterowanie, przy jednoczesnym podążaniu za liderem oraz unikanie przeszkód.

Rozproszone sterowanie formacją robotów posiada unikalne właściwości, a mianowicie: skalowalność, odporność na awarie, zdolność do realizacji złożonych zadań przez użycie prostych, autonomicznych jednostek. W pracy analizowano sterowanie oparte na wirtualnych potencjałach między sąsiadującymi robotami oraz zastosowano teorię grafów do opisu wymiany informacji między jednostkami.

1.1 Cele pracy

Celem głównym pracy jest skonstruowanie algorytmów sterowania rozproszonego dla układów zbudowanych z wielu autonomicznych robotów mobilnych. Zadaniem zespołu robotów jest osiągnięcie zdefiniowanej formacji wyrażonej przez wzajemne odległości pomiędzy robotami.

Celem podrzędnym pracy jest budowa środowiska sprzętowo – programowego umożliwiającego testowanie algorytmów sterowania dla robotów mobilnych.

1.2 Przegląd literatury

1.2.1 Wprowadzenie

Na początku podjęto próbę osadzenia problemu sterowania zespołem współpracujących robotów mobilnych w szerszym kontekście i systematyzacji dostępnej literatury. Jako punkt wyjścia przyjęto rodzaj sterowania. Rozproszone sterowanie, w przeciwieństwie do centralnego sterowania, zakłada brak nadrzędnej jednostki wydającej polecenia podrzędnym wykonawcom, a osiągnięcie wspólnego celu jest realizowane we współpracy. Inspiracją takiego zachowania są systemy biologiczne, np. wspólny ruch klucza ptaków, ławicy ryb czy stada zwierząt (rysunek 1.1).

W 1987 roku, w pracy [74] został zaprezentowany model komputerowy symulu-

jący zachowanie stadne powszechnie znany pod nazwą Boids¹. Praca ta pokazała, że bardzo proste, stosowane lokalnie zasady pozwalają uzyskać skomplikowane zachowanie stadne. W modelu Reynoldsa występują wyłącznie trzy reguły:

1. unikanie kolizji,
2. wyrównanie prędkości i kierunku ruchu,
3. dążenie do pozostania w środku grupy.

Każda z nich jest stosowana tylko do ograniczonego sąsiedztwa, ponieważ percepcja „boidów” jest wyłącznie lokalna. Model ten pokazał, że zespół mobilnych obiektów może osiągnąć globalne zadanie bazując wyłącznie na lokalnych obserwacjach i zasadach.



(a)



(b)



(c)



(d)

Rysunek 1.1: Przykłady kolektywnego zachowania w świecie zwierząt a: chmara ptaków (autor: Gregory Hunt, źródło: <https://www.flickr.com/photos/eosgreg/6965121429/>), b: stado owiec (autor: Dariusz Paciorek, źródło: <https://www.istockphoto.com/pl/zdjęcie/gm155150469-18487497>), c: ławica ryb (autor: Robin Hughes, CC), d: wataha wilków (źródło: Park Narodowy Yellowstone, domena publiczna)

W 1995 roku została opublikowana praca [88], w której opisano model „cząstek”, którym nadana jest stała prędkość i losowy kierunek ruchu. Vicsek i in. wykazali, że prosta reguła, która uśrednia kierunek ruchu na podstawie obserwacji lokalnego

¹Boids od ang. *bird-oid*, *bird-like* – nowojorskie, slangowe określenie ptaków

sąsiedztwa prowadzi do synchronizacji ruchu całej grupy pomimo występowania zakłóceń.

Powyższe prace zawierały modele symulacyjne dobrze odzwierciedlające zjawiska występujące w przyrodzie, ale nie dostarczały precyzyjnego opisu i nie określały warunków, których spełnienie jest konieczne do osiągnięcia pożądanego zachowania grupy obiektów. W latach dwutysięcznych powstało wiele prac formalizujących i uogólniających rozważania dotyczące ww. zjawisk.

1.2.2 Początki opisu formalnego

W 2003 roku Jadbabaie i in. [42] dostarczyli formalnego opisu modelu Vicseka w oparciu o grafy definiujące sąsiedztwo „cząstek”. Ich rozważania mogą zostać podsumowane do następującej tezy: jeśli graf opisujący sąsiedztwo elementów pozostaje spójny dla każdej chwili czasu to osiągną one konsensus ich wektorów prędkości. Innymi słowy obiekt sterowania (złożony z poruszających się ze stałą prędkością cząstek), gdzie każda cząstka na podstawie lokalnej obserwacji (kierunku ruchu sąsiadów) i używając lokalnego sterowania (uśrednionej wartości sterowania) osiąga zadaną wartość czyli wspólny kierunek ruchu.

Wnioski płynące z powyższych obserwacji mogą znaleźć zastosowanie w różnych dziedzinach. Wśród nich zespół robotów mobilnych jest naturalnym kandydatem do zastosowania zdecentralizowanego sterowania. Istnieje wiele potencjalnych zalet zastąpienia jednego skomplikowanego urządzenia wieloma, ale prostszymi obiektami (w podrozdziale 1.2.4 podano kilka przykładów, dodatkowe są opisane w [44]). Jednocześnie zastosowanie zdecentralizowanego sterowania pozwala osiągnąć wiele pożądanых właściwości, takich jak:

- skalowalność – czyli zdolność do prawidłowego działania w warunkach rosnącej liczby robotów,
- niezawodność i dostępność – czyli odporność na pojedyncze awarie oraz łatwość zastąpienia wadliwych obiektów,
- uniwersalność i elastyczność – czyli możliwość zastosowania powtarzalnych, standaryzowanych komponentów.

Te zalety doprowadziły do szybkiego rozwoju dziedziny związanej ze zdecentralizowanym sterowaniem robotów, które współpracują w celu osiągnięcia globalnego celu. Pierwsze prace skupiały się na szczegółowej analizie matematycznej wykorzystując m.in. idee z algebraicznej teorii grafów [23, 27], teorię potencjałów [66, 61] oraz metody Lapunowa [60].

Dalsze prace doprowadziły do wyróżnienia kilku podstawowych problemów w zależności od definicji zadania. Najprostszym zagadnieniem jest doprowadzenie robotów do wspólnego punktu, czyli algorytm spotkania (ang. *rendezvous*) [56, 1]. Popularnym jego uogólnieniem jest algorytm konsensusu, w którym grupa robotów (agentów) osiąga zgodność co do jakiejś wartości (miejsca, prędkości, mierzonej wartości, w tym czasu (synchronizacja)). Odmienny problem polega na rozmieszczeniu robotów (lub mobilnych sensorów) na danym terenie w taki sposób, aby zapewnić pokrycie lub monitorowanie całej powierzchni (ang. *coverage control*) [54, 89]. Innym problemem, będącym istotą pracy, jest zadanie sterowania formacją robotów (ang. *formation control*) [7, 2, 53]. Zadanie to polega na koordynacji ruchu robotów w taki sposób, aby osiągnęły one zadane, wzajemne położenie (np. reprezentowane przez figurę geometryczną) korzystając z lokalnych informacji, np. pomiaru położenia sąsiada.

1.2.3 Sterowanie formacją robotów

W badaniach dotyczących sterowania formacją współpracujących robotów mobilnych istotną kwestią jest sposób wymiany informacji oraz oddziaływanie robotów pomiędzy sobą. Ponieważ sterowaną wartością jest względne położenie robotów to przez informację rozumie się położenie sąsiadów. Wymiana tej informacji może być oparta na komunikacji pomiędzy robotami lub być pozyskiwana za pomocą czujników, np. kamer.

W zależności od rodzaju informacji posiadanej przez roboty rozważane są różne typy sterowania formacją robotów. Wyróżnia się cztery główne kategorie.

1. Bazujące na pozycji sąsiada (ang. *position-based*). Metody te zakładają istnienie globalnego układu współrzędnych oraz znajomości bezwzględnego położenia sąsiadów. Przykładowym czujnikiem, który może dostarczyć takich informacji są moduły nawigacji bazujące na systemie GPS. W pracach [73, 87, 25, 50] przedstawiono propozycje sterowania, które pozwala śledzić zadaną trajektorię przez zespół robotów utrzymując zadaną formację.

Na rysunku 1.2 przedstawiono przykładową aplikację, tego typu sterowania - rój dronów realizujący pokaz świetlny.



Rysunek 1.2: Pokaz świetlny złożony z roju dronów (źródło: [Intel])

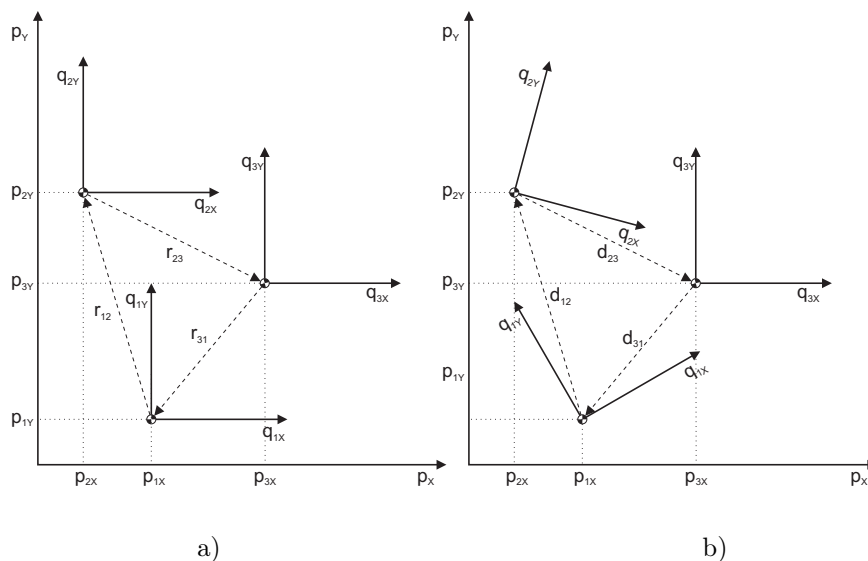
Istnienie globalnego układu współrzędnych nie wyklucza projektowania sterowania w systemie rozproszonym. Jednocześnie ten typ sterowania wymaga najbardziej zaawansowanego systemu czujników oraz nie zapewnia niezawodności układów bazujących wyłącznie na lokalnej informacji [63]. W przypadku awarii komponentu odpowiedzialnego za globalny układ współrzędnych sterowanie nie może zostać zrealizowane.

2. Bazujące na wzajemnym położeniu (ang. *displacement-based*). Metody te zakładają istnienie lokalnych układów współrzędnych przypisanych do każdego robota o orientacji wyrównanej z globalnym układem współrzędnych, czyli skierowanych w tę samą stronę (rysunek 1.3 a).

W tej konfiguracji cel sterowania to osiągnięcie zdefiniowanych pozycji pomiędzy robotami $p_i - p_j = r_{ij}^*$, $i, j \in V$ gdzie wektor r_{ij}^* określa zadane przesunięcie pomiędzy robotami i, j , a V określa zbiór wierzchołków w grafie opisującym wymianę informacji.

Zakłada się, że roboty znają względne położenie swoich sąsiadów czyli potrafią zmierzyć ich kierunek oraz względną odległość, a dodatkowo znają orientację globalnego układu współrzędnych (zmierzoną np. za pomocą kompasu) [43, 26]. Wartości odległości mogą zostać zmierzone przez robota za pomocą czujników, takich jak radar, lidar czy sonar. Ponieważ układy współrzędnych mają jednakową orientację to pomiar wzajemnych położenia w lokalnym i globalnym układzie współrzędnych są jednakowe.

Głównym narzędziem w rozwiązywaniu zadań w tej konfiguracji są grafy, a podstawową metodą gwarantującą osiągnięcie zadania jest zapewnienie spójności grafu [63, 43, 65].



Rysunek 1.3: Zależność pomiędzy lokalnymi układami współrzędnych robotów, a globalnym układem współrzędnych, a) lokalne układy współrzędnych o orientacji wyrównanej z układem globalnym, oznaczone wzajemne położenia, b) lokalne układy współrzędnych o dowolnej orientacji, oznaczone wzajemne odległości

3. Bazujące na odległości pomiędzy robotami (ang. *distance-based*). Metody te zakładają istnienie lokalnych układów współrzędnych, które są niezależne od globalnego układu współrzędnych.

W tej konfiguracji cel sterowania to osiągnięcie zdefiniowanych odległości pomiędzy robotami $\|p_i - p_j\| = d_{ij}^*$, $i, j \in V$ gdzie d_{ij}^* określa zadaną odległość pomiędzy robotami i, j .

W tych metodach zakłada się, że roboty mierzą odległość do swoich sąsiadów w swoich lokalnych układach współrzędnych (rysunek 1.3 b). Notacja globalnego układu współrzędnych może być pomocna w analizie sterowania, ale informacja o jego istnieniu jest niedostępna dla poszczególnych robotów. Wartości odległości mogą zostać zmierzone przez robota za pomocą czujników, takich jak radar, lidar czy sonar.

Algorytmy sterowania w tej kategorii wymagają strukturalnej sztywności grafu, a sterowania są w postaci gradientu odległości [67, 24, 62].

4. Bazujące na kierunku pomiędzy robotami (ang. *bearing-based*). Metody te zakładają istnienie lokalnych układów współrzędnych, które są niezależne od globalnego układu współrzędnych, a ponadto jedyną mierzoną wielkością jest kierunek, w którym znajduje się sąsiad bez znajomości wzajemnej odległości.

Algorytmy sterowania w tej kategorii wymagają strukturalnej sztywności grafu

oraz wykrywaniu kąta pomiędzy sąsiadami [21, 19, 18, 81]. Taka informacja może zostać uzyskana przez zastosowanie dookolnej kamery.

Kolejność powyższych kategorii jest uporządkowana w zależności od wymagań dotyczących mierzonych wartości. Algorytmy sterujące pozycją wymagają najbardziej zaawansowanego systemu czujników, a każda następna kategoria pozwala na uproszczenie systemu pomiarowego [71].

W pracy rozważane są algorytmy sterowania formacją bazujące na wzajemnych odległości pomiędzy robotami. Do opisu formacji robotów, zachodzących pomiędzy nimi interakcji (sąsiedztwa) oraz definicji zadanych odległości wykorzystuje się grafy. Zależności te zostaną przedstawione w kolejnych rozdziałach pracy. **W związku z odległością pomiędzy robotami, będącą nieliniową funkcją, sterowanie jest nieliniowe nawet, gdy poszczególne roboty są opisane przez liniowe równania różniczkowe.** Analiza właściwości, w szczególności określenie stabilności, formacji robotów mobilnych stanowi duże wyzwanie. Do najpopularniejszych metod używanych w analizie ww. układów należą funkcjonały Lapunowa.

W tabeli 1.1 zostały zebrane istotne prace dotyczące tematyki sterowania formacją opartą na odległościach. W drugiej kolumnie tabeli zawarto informacje o sposobie wymiany informacji (lub zachodzących pomiarach) pomiędzy robotami. Jeśli wymiana informacji jest symetryczna to do opisu używany jest graf prosty (nieskierowany). Jeśli dopuszcza się jednokierunkową wymianę informacji pomiędzy robotami (np. w algorytmie śledzenie lidera) to do opisu używa się grafów skierowanych. W trzeciej kolumnie zaznaczono jakie obiekty były przedmiotem badań w cytowanych publikacjach. Najczęstszym obiektem badań są punkty kinematyczne oraz masy punktowe. Takie uproszczone modele robotów znajdują zastosowanie, ponieważ zwykle obiektem zainteresowania jest cała formacja a nie poszczególne roboty. W ostatniej kolumnie opisano uzyskany w cytowanej pracy rezultat analizy stabilności.

Tabela 1.1: Zestawienie publikacji dotyczących utrzymania formacji robotów

Odnosnik	Opis interakcji	Opis obiektu	Rezultat
[79, 53, 26]	graf prosty	n punktów kinematycznych	AS
[90]	graf prosty	trzy punkty kinematyczne	GAS
[14, 59]	digraf bez cyklu	trzy punkty kinematyczne	EXS
[2, 15, 16]	digraf z cyklem	trzy punkty kinematyczne	EXS
[67, 62]	graf	n mas punktowych	AS

Do obszaru zainteresowań, poza analizą stabilności, należy również szereg innych

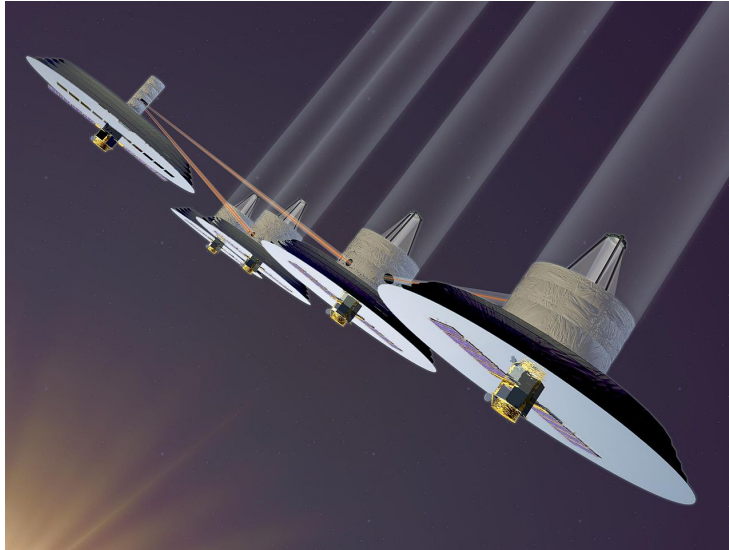
zagadnień. Na przykład w pracach [11, 31] został przeanalizowany przypadek stabilizacji formacji robotów, gdy wzajemnie mierzone odległości są zaburzone. Rozwiązanie prezentowane w [31] polega na konstrukcji rozproszonych obserwatorów, które odtwarzają poprawną wartość odległości. W pracach [24, 71] badano sterowanie zapewniające jednoczesne unikanie kolizji pomiędzy robotami. W pracach [71, 76] analizowano sterowanie umożliwiające jednoczesne sterowania formacją i omijanie przeszkód. Dynamiczna zmiana zadanych odległości pomiędzy robotami (skalowanie formacji) została opisana np. w [69, 20, 71]. Aspekty związane z zależnościami czasowymi przy wymianie informacji pomiędzy robotami zostały opisane w [32].

1.2.4 Zastosowania

Do szczególnie istotnych zastosowań sterowania rozproszonego opartego na wzajemnych odległościach można zaliczyć:

- Formacje satelitarne (ang. *satellite formation flight*) to zespół złożony z więcej niż jednego satelity, w którym ich stan (np. orientacja) jest sprzężony za pomocą wspólnego algorytmu sterującego [77]. Koncepcje zastosowania formacji satelitarnych rozważane są od lat 70. ze względu na związane z tym zalety. Przykładem może być projekt Terrestrial Planet Finder składający się z zespołu satelitów używających interferometrii do poszukiwania pozasłonecznych planet [57, 41, s. 6]. Koncepcyjna grafika tego projektu została przedstawiona na rysunku 1.4. Rozproszenie czujników pomiarowych na znaczne odległości i ich precyzyjne pozycjonowanie pozwala uzyskać zdolności pomiarowe przekraczające możliwości pojedynczych sensorów (interferometria wielkobazowa, ang. *Very Long Baseline Interferometry*).

Koszt wysłania wielu małych satelitów może być mniejszy niż wysłanie jednego dużego satelity, a potencjalne zyski są znaczne [72, 102]. Jednak centralne sterowanie formacją rodzi wyzwania wynikające np. z opóźnień czasowych podczas komunikacji. Potencjalne zastosowanie rozproszonych metod bazujących na lokalnej wymianie informacji otwiera nowe możliwości w tej dziedzinie [77].



Rysunek 1.4: Wizja satelitarnego systemu interferometrii - projekt Terrestrial Planet Finder (źródło: [NASA], domena publiczna)

- Loty formacyjne (ang. *formation flight*) to rodzaj lotu, gdzie dwie lub więcej maszyn, takich jak samoloty, drony czy inne pojazdy latające, poruszają się razem, utrzymując ściśle określoną konfigurację w przestrzeni. Tego typu loty mogą być wykorzystywane zarówno w celach militarnych, jak i cywilnych.

W związku z rosnącą ilością bezzałogowych statków powietrznych (ang. *unmanned aerial vehicle*, UAV), implementacja algorytmów sterowania rozproszonego stanowi obszar aktywnych badań (np. [13]). Ważne zastosowania cywilne dotyczą monitorowania obszarów (wykrywanie obiektów/zagrożeń, misje poszukiwawczo-ratownicze, badania zanieczyszczeń, tworzenie map) [57, 84, 107]. Osobną gałąź stanowią zastosowania wojskowe. W obydwu przypadkach metody sterowania oparte na wzajemnych odległościach pomiędzy robotami i lokalne sterowanie, stanowią ważne zastosowanie w przypadku braku zewnętrznego źródła informacji o położeniu, np. gdy sygnał nawigacji satelitarnej ulega zakłóceniom.

- Autonomiczny transport, w szczególności autonomiczne pojazdy, to technologie pozwalające poruszać się ze zmniejszonym nadzorem człowieka (lub jego brakiem). Współczesne pojazdy są wyposażone w niezbędne czujniki pozwalające widzieć i interpretować swoje otoczenie. Wśród nich są lidary, radary i kamery, które dostarczają informacji o odległości i pozycji sąsiadujących pojazdów [106]. Sterowanie rozproszone stanowi potencjalne uzupełnienie zaawansowanych systemów wspomagania kierowcy oraz, wraz z technologiami komunikacji pomiędzy pojazdami i infrastrukturą (ang. *vehicle-to-everything*, V2X), może przyczynić się do rozwoju inteligentnej mobilności [80, 55, 106].

Do innych zastosowań związanych ze sterowaniem rozproszonym, które są pośrednio związane z tą pracą, można zaliczyć koordynację robotów logistycznych ([97]), zarządzanie i bilansowanie sieci energetycznych (ang. *smart grid*) [45] oraz pomiary przy pomocy bezprzewodowych sieci sensorycznych (ang. *wireless sensor networks*) [46].

1.2.5 Istotne, zbliżone obszary badawcze

Tematem pracy jest sterowanie rozproszone fizycznych obiektów (robotów) współpracujących w celu osiągnięcia postawionego zadania. Bardzo ważnym obszarem, który uogólnia te rozważania, są systemy wieloagentowe (ang. *Multi-agent system*, MAS). Generalizacja dotyczy w szczególności rozważania dowolnych obiektów, niekoniecznie mających fizyczną reprezentację (np. programów komputerowych). Ponadto w systemach wieloagentowych założenie o współpracy (kooperacji) w celu osiągnięcia globalnego celu jest jedną z opcji, a nie założeniem.

Podstawowym elementem MAS jest agent - abstrakcyjna jednostka, umieszczona w zdefiniowanym środowisku, która może podejmować decyzje w celu realizacji postawionych zadań (przykładowy opis np. w [17]).

Jeśli jako agent zostanie potraktowany robot mobilny, jego środowiskiem będzie płaszczyzna, po której może się poruszać, a zadaniem utrzymanie formacji, to taki system wieloagentowy będzie podobny do opisywanego w tej pracy. Z drugiej strony agentem może być algorytm komputerowy, środowiskiem zbiór danych, a zadaniem optymalizacja wskaźnika jakości.

Główne kierunki badań w obszarze systemów wieloagentowych dotyczą m.in. formalizmu opisu, koordynacji i negocjacji celów, komunikacji pomiędzy agentami czy odporności na błędy [91, 78]. Zagadnienia te nie są obszarem zainteresowania tej pracy.

Innym istotnym obszarem badawczym jest robotyka bazująca na zachowaniach (ang. *behavior-based robotics*). Jest to dziedzina zainspirowana systemami biologicznymi. W sterowaniu behawioralnym definiuje się szereg zachowań robota, które są reakcją na otoczenie [4]. Do takich zachowań można zaliczyć zadania związane ze sterowaniem formacją, np. dostosuj prędkość, unikaj kolizji, pozostań w grupie.

Zaletą metod behawioralnych jest prostota konstrukcji sterowania w oparciu się na definicji reguł będących odpowiedzią na zmieniające się środowisko. Metody te pozwalają na samoorganizację zespołu robotów i osiągnięcie inteligencji stadnej. Ich wadą jest mniej precyzyjny zapis niż w metodach opisywanych w tej pracy. Może to utrudnić lub uniemożliwić charakteryzację układów regulacji, np. stabilność systemu zamkniętego.

1.3 Stosowana terminologia

W pracy starano się zastosować polskie formy i nazwy do opisu zjawisk i definicji wszędzie tam, gdzie występuje utarta terminologia.

Niektóre zjawiska i wyrażenia nie mają polskich odpowiedników, które zostały powszechnie zaakceptowane i weszły do ogólnego użycia. Przykładem może być słowo *jitter*, które nie ma polskiego odpowiednika i musi być zastąpione opisowym wyjaśnieniem. Jest to zjawisko, związane z obliczeniami lub przetwarzaniem w czasie rzeczywistym, w którym rzeczywisty okres próbkowania jest różny od ustalonego. Różnica ta nazywana jest właśnie angielskim terminem *jitter*.

Występują również przypadki, w których podano angielskie terminy obok ich polskich odpowiedników w celu wykluczenia nieporozumień wynikających z wieloznaczności tłumaczenia.

We wszystkich miejscach w pracy, w których występuje zwrot „formacja robotów” autor ma na myśli ustaloną strukturę geometryczną o zdefiniowanych odległościach pomiędzy robotami. Jest to bezpośrednio tłumaczenie zwrotu *multi – robot formation* występujące w literaturze anglojęzycznej.

Graf strukturalnie sztywny opisany w definicji 2.20. to bezpośrednio tłumaczenie angielskiego terminu *rigid graph*.

W opisie obiektów będących przedmiotem sterowania - robotów mobilnych często wymiennie stosuje się określenie pojazd. Innym często spotykanym w literaturze określeniem jest słowo agent. Występuje ono zarówno w polskiej (np. [17]) jak i angielskiej literaturze (np. [63]). Ponieważ często jest ono używane w szerszym kontekście, w tej pracy unika się tego określenia.

1.4 Układ pracy

W rozdziale 2 przedstawiono zbiór definicji, które są wykorzystywane do opisu zespołu robotów, analizy ich zachowania i konstrukcji algorytmów sterujących. W rozdziale 3 zostały przedstawione przykłady wprowadzające, które służą jako punkt wyjścia do dalszych rozważań. W rozdziale 4 przedstawiono analizę wybranych algorytmów sterujących grupą robotów nieholonomicznych. W rozdziale tym zamieszczono również wyniki symulacyjne i eksperymentalne przedstawiające działanie proponowanych algorytmów. W rozdziale 5 przedstawiono platformę składającą się z niestabilnego robota mobilnego służącego do transportu i rozwiązań programowo – sprzętowych umożliwiających jego programowanie i sterowanie. Następnie przedstawiono wyniki symulacyjne i eksperymentalne dotyczące implementacji algorytmów dla opisywanego robota. Rozdział 6 stanowi podsumowanie pracy. Zawarto

w nim omówienie otrzymanych wyników na tle innych prac. Ponadto przeanalizowano osiągnięte cele i nakreślono możliwe przyszłe kierunki badań. W dodatkach A i B przedstawiono implementacje algorytmów opisanych w przykładach zawartych w rozdziale 3. W dodatku C zamieszczono odnośniki do nagrań i wizualizacji z przeprowadzonych eksperymentów.

Elektroniczna wersja pracy zawiera odnośniki do elementów numerowanych (spis treści, rysunki, wzory, tabele, bibliografia). Kliknięcie w odnośnik przenosi do odpowiedniego elementu (niektóre programy, np. Adobe Reader, umożliwiają powrót do wcześniejszego miejsca w tekście poprzez skrót **Alt** + **←**). Większość osadzonych ilustracji, w tym wszystkie wykresy, mają postać grafiki wektorowej. Dzięki temu możliwe jest ich znaczne powiększenie.

2 Informacje wprowadzające

W tym rozdziale zostały zebrane podstawowe pojęcia związane z analizą matematyczną, teorią sterowania oraz teorią grafów, które są używane w dalszych etapach pracy. Mogą okazać się pomocnymi przy czytaniu pracy.

2.1 Podstawowe pojęcia i oznaczenia

Rozważania dalszych rozdziałów skupione są wokół formacji robotów mobilnych na płaszczyźnie. Podstawową informacją potrzebną do sterowania wzajemnym położeniem będzie odległość pomiędzy sąsiadującymi robotami.

Definicja 2.1. (Metryka) Metryka to funkcja określająca odległość. Zakładając, że w zbiorze \mathcal{X} istnieją punkty $a, b \in \mathcal{X}$, można dla nich zdefiniować odległość $d(a, b) \in [0, +\infty)$ spełniającą następujące postulaty:

- $d(a, b) = 0 \Leftrightarrow a = b$,
- $d(a, b) = d(b, a)$,
- $d(a, c) \leq d(a, b) + d(b, c)$.

Parę (\mathcal{X}, d) nazywamy przestrzenią metryczną, a funkcję d – metryką.

Definicja 2.2. (Metryka Euklidesowa) W przestrzeni \mathbb{R}^n metryka euklidesowa jest zdefiniowana jako:

$$d_e(a, b) = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2}.$$

Metryka euklidesowa spełnia postulaty metryki (identyczność nierozróżnialnych, symetria, nierówność trójkąta). W pracy przyjęto zapis metryki euklidesowej w postaci

$$d_e(a, b) = \|a, b\|.$$

2.2 Teoria sterowania

Do opisu różnych procesów i zjawisk zachodzących w czasie często używa się równań różniczkowych. W tej pracy równania różniczkowe służą do opisu oraz analizy dynamiki zarówno zespołu robotów jak i poszczególnych robotów.

Definicja 2.3. (Równanie różniczkowe zwyczajne) Równanie o postaci:

$$\dot{x}(t) = f(x(t)), \tag{2.1}$$

gdzie f jest funkcją różniczkowalną, odwzorowującą $\mathbb{R}^n \rightarrow \mathbb{R}^n$, a $x(t) \in \mathbb{R}^n$ jest wektorem stanu, nazywa się stacjonarnym układem równań różniczkowych zwyczajnych lub w uproszczeniu, gdy nie wprowadza niejasności, równaniem różniczkowym. Ogólna postać równania różniczkowego została zdefiniowana np. w [58, s. 107], [38, s. 39].

Równanie takie opisuje zmienność procesu. Miarą zmienności jest pochodna stanu względem czasu, a przyszły stan procesu $x(t)$ zależy wyłącznie od stanu wcześniejszego $x(t_0)$, gdzie $t > t_0$. Zmienna t jest interpretowana jako czas.

W układach automatyki równanie (2.1) często jest uogólnione do postaci

$$\dot{x}(t) = f(x(t), u(t)), \quad (2.2)$$

gdzie f jest funkcją odwzorowującą $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, $x(t) \in \mathbb{R}^n$ jest stanem, a $u(t) \in \mathbb{R}^m$ jest sterowaniem.

Przykładowe modele matematyczne zjawisk fizycznych opisane przy pomocy równań różniczkowych są zamieszczone np. w [47, s. 1-56], [38, s. 39-108], [103, s. 1-47].

Definicja 2.4. (Punkt równowagi) Punktem równowagi równania różniczkowego (2.1) nazywa się taki stan $\bar{x} \in \mathbb{R}^n$, dla którego zachodzi

$$f(\bar{x}) = 0. \quad (2.3)$$

Oznacza to, że punktowi równowagi odpowiada stałe w czasie rozwiązanie [10, s. 33].

W przypadku równania (2.2) punkt równowagi jest parą (\bar{x}, \bar{u}) oraz zachodzi $f(\bar{x}, \bar{u}) = 0$.

Punkty równowagi są ważną cechą systemów dynamicznych, ponieważ definiują one stany odpowiadające stałym warunkom pracy [5, s. 158].

Definicja 2.5. (Liniowe równanie różniczkowe zwyczajne) Równanie postaci

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2.4)$$

gdzie $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, nazywane jest liniowym równaniem różniczkowym. W przypadku równania bez sterowania przyjmuje ono postać

$$\dot{x}(t) = Ax(t) \quad (2.5)$$

i jest nazywane autonomicznym, liniowym równaniem różniczkowym.

Pierwiastki wielomianu charakterystycznego macierzy A (widmo macierzy) $\sigma(A)$ określają istotne własności dynamiczne układu (2.5) [10, s. 9], [58, s. 118].

Definicja 2.6. (Linearyzacja równania różniczkowego) Linearyzacja polega na przybliżeniu nieliniowego równania różniczkowego jego liniowym odpowiednikiem. Przy spełnieniu założeń opisanych w twierdzenie Hartmana – Grobmana, zachowanie zlinearyzowanego modelu w otoczeniu punktu równowagi jest podobne do zachowania oryginalnego modelu [40], [35, s. 30].

Dla równania różniczkowego (2.1) można zdefiniować odchyłkę od punktu równowagi \bar{x} jako:

$$\Delta x(t) := x(t) - \bar{x}. \quad (2.6)$$

Równanie ruchu dla nowej zmiennej ma postać:

$$\Delta \dot{x}(t) = f(\bar{x} + \Delta x(t)). \quad (2.7)$$

Powyższe równanie, po rozwinięciu w szereg Taylora oraz pominięciu wyrazów wyższych (od pierwszego) rzędów, przyjmuje postać:

$$\Delta \dot{x}(t) \approx f(\bar{x}) + \left. \frac{\partial f}{\partial x} \right|_{x=\bar{x}} \Delta x(t). \quad (2.8)$$

Ponieważ \bar{x} jest punktem równowagi systemu (2.1) i zachodzi $f(\bar{x}) = 0$, to równanie (2.8) upraszcza się do

$$\Delta \dot{x}(t) \approx \left. \frac{\partial f}{\partial x} \right|_{x=\bar{x}} \Delta x(t). \quad (2.9)$$

Powszechnie stosuje się definicję:

$$A := \left. \frac{\partial f}{\partial x} \right|_{x=\bar{x}} \in \mathbb{R}^{n \times n}, \quad (2.10)$$

gdzie macierz A nazywana jest macierzą stanu, a równanie (2.9) zapisuje się w postaci (2.5).

Macierz pochodnych cząstkowych funkcji f w równaniu (2.8) nazywa się macierzą Jacobiego i definiuje w następujący sposób:

$$\mathbb{J} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}. \quad (2.11)$$

Definicja 2.7. (Stabilność układu dynamicznego [5, s. 160]) Jeśli trajektoria $x(t, a)$ jest rozwiązaniem równania różniczkowego (2.1) z warunkiem początkowym a , to takie rozwiązanie nazywamy stabilnym w sensie Lapunowa, jeśli rozwiązania zaczynające się z podobnym warunkiem początkowym pozostają w pobliżu trajektorii $x(t, a)$, czyli dla każdej stałej $\epsilon > 0$ istnieje taka stała $\gamma > 0$, dla których jest spełnione:

$$\|b - a\| < \gamma \quad \Rightarrow \quad \|x(t, b) - x(t, a)\| < \epsilon \quad \forall t > 0.$$

Szczególnie istotnym przypadkiem jest sytuacja, w której trajektoria $x(t, a)$ jest punktem równowagi układu (2.1). Jeśli $x(0)$ oznacza warunek początkowy trajektorii $x(t)$, a punkt \bar{x} spełnia warunek

$$\|x(0) - \bar{x}\| < \gamma \quad \Rightarrow \quad \|x(t) - \bar{x}\| < \epsilon \quad \forall t > 0,$$

to taki punkt równowagi nazywa się stabilnym.

Definicja 2.8. (Asymptotyczna stabilność układu dynamicznego) Rozwiązanie $x(t, a)$ jest asymptotycznie stabilne jeśli jest stabilne i dodatkowo trajektorie zaczynające się w jej pobliżu zbiegają do trajektorii $x(t, a)$, gdy $t \rightarrow \infty$ ([5, s. 161]). Oznacza to, że dla każdej stałej $\epsilon > 0$ istnieje taka stała $\gamma > 0$, że zachodzi następująca relacja:

$$\|b - a\| < \gamma \quad \Rightarrow \quad \|x(t, b) - x(t, a)\| < \epsilon \quad \wedge \quad \lim_{x \rightarrow \infty} \|x(t, b) - x(t, a)\| = 0.$$

Jeśli punkt równowagi układu jest lokalnie asymptotycznie stabilny, to stosuje się zapis **AS**. W szczególnym przypadku, gdy obszar atrakcji jest równy przestrzeni stanów, stosuje się zapis **GAS**, oznaczający globalną asymptotyczną stabilność.

Definicja 2.9. (Wykładnicza stabilność układu dynamicznego) Jeśli rozwiązanie $x(t, a)$ jest asymptotycznie stabilne i dodatkowo istnieją takie stałe $\alpha > 0$, $\beta > 0$, $\gamma > 0$, że zachodzi:

$$\|b - a\| < \gamma \quad \Rightarrow \quad \|x(t, b) - x(t, a)\| \leq \alpha \|x(0, b) - x(0, a)\| e^{-\beta t} \quad \forall t > 0,$$

to punkt równowagi układu jest wykładniczo stabilny. Wykładnicza stabilność dostarcza informacji o tym, że tempo zbieżności badanego rozwiązania układu (2.1) jest nie mniejsze niż $\alpha \|x(0, b) - x(0, a)\| e^{-\beta t}$.

Przyjęto skrótowy zapis wykładniczej stabilności w postaci **EXS**.

Definicja 2.10. (I metoda Lapunowa) Pierwsza metoda Lapunowa pozwala uzyskać informacje odnośnie stabilności układu (2.1) wokół punktów równowagi. Idea metody polega na przeprowadzeniu linearyzacji układu (2.1) wokół punktu równowagi oraz określeniu charakteru układu zlinearyzowanego poprzez badanie widma macierzy stanu. Jeśli wszystkie wartości własne macierzy A mają ujemne części rzeczywiste, to punkt równowagi układu (2.1) jest asymptotycznie stabilny [10].

Definicja 2.11. (II metoda Lapunowa) Druga metoda Lapunowa, nazywana również metodą funkcjonałów Lapunowa, pozwala zbadać stabilność punktów równowagi \bar{x} układu (2.1) na podstawie badania energetycznego funkcjonału V , bez konieczności rozwiązywania równań stanu [35]. Do testu wykorzystuje się odpowiednio dobraną funkcję $V : \mathbb{R}^n \supset \Omega \rightarrow \mathbb{R}$, posiadającą ciągłe pochodne cząstkowe, o własności:

$$V(x) > 0 \quad \forall x \in \Omega \setminus \{\bar{x}\}, \quad V(\bar{x}) = 0, \quad (2.12)$$

gdzie Ω jest pewnym otoczeniem badanego punktu równowagi. Jeśli

$$\dot{V}(x) \leq 0 \quad \forall x \in \Omega \setminus \{\bar{x}\}, \quad (2.13)$$

to punkt równowagi jest stabilny. Funkcję V spełniającą powyższe warunki nazywa się funkcjonałem Lapunowa. Funkcja \dot{V} jest pochodną po czasie funkcjonału V liczoną na trajektoriach układu (2.1). Jeśli spełniony jest warunek

$$\dot{V}(x) < 0 \quad \forall x \in \Omega \setminus \{\bar{x}\}, \quad (2.14)$$

to punkt równowagi \bar{x} jest asymptotycznie stabilny. Jeśli dodatkowo $\Omega = \mathbb{R}^n$, to układ jest globalnie asymptotycznie stabilny (**GAS**).

Definicja 2.12. (Stabilność układów liniowych) Badanie stabilności układów liniowych (2.5) sprowadza się do badania wartości własnych macierzy A . Do ich wyznaczenia należy obliczyć pierwiastki wielomianu charakterystycznego macierzy A :

$$\sigma(A) = \{\lambda \in \mathbb{C} : \det(\lambda \mathbb{I} - A) = 0\}. \quad (2.15)$$

Jeśli λ_i jest wartością własną macierzy A , to istnieje taki wektor $v_i \in \mathbb{R}^n$, dla którego spełnione jest równanie $Av_i = v_i \lambda_i$. Wektor taki nazywamy wektorem własnym odpowiadającym danej wartości własnej.

Jeśli wszystkie wartości własne macierzy A mają części rzeczywiste mniejsze od 0, to układ jest asymptotycznie stabilny.

Jeżeli wszystkie wartości własne macierzy A mają części rzeczywiste mniejsze od

0 lub równe 0 i z każdą wartością własną o zerowej części rzeczywistej λ_i o krotności m jest związanych m liniowo niezależnych wektorów własnych, to układ jest stabilny.

Definicja 2.13. (Problem liniowo – kwadratowy z nieskończonym horyzontem czasowym)

Dla układu liniowego:

$$\dot{x} = Ax + Bu, \quad (2.16)$$

gdzie $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, problem liniowo-kwadratowy polega na poszukiwaniu takiego sterowania u , które minimalizuje wskaźnik jakości

$$J = \int_0^\infty (x(t)^\top Q x(t) + u(t)^\top R u(t)) dt, \quad (2.17)$$

gdzie:

- $Q = Q^\top \geq 0$ jest macierzą wag związaną ze stanem, symetryczną, półdefinitnie określoną,
- $R = R^\top > 0$ jest macierzą wag związaną ze sterowaniem, symetryczną, dodatnio określoną.

Składnik $x^\top Q x$ obrazuje straty związane z niedotrzymaniem stanu zadanego, a składnik $u^\top R u$ koszty sterowania [85, s. 140–143].

Jeśli para (A, B) jest stabilizowalna, a para (Q, A) wykrywalna, to dla każdego $x(0)$ istnieje dokładnie jedno sterowanie minimalizujące wskaźnik jakości (2.17) [58, s. 554] i jest ono postaci:

$$u = Kx, \quad (2.18)$$

gdzie macierz K wynosi:

$$K = -R^{-1}B^\top P, \quad (2.19)$$

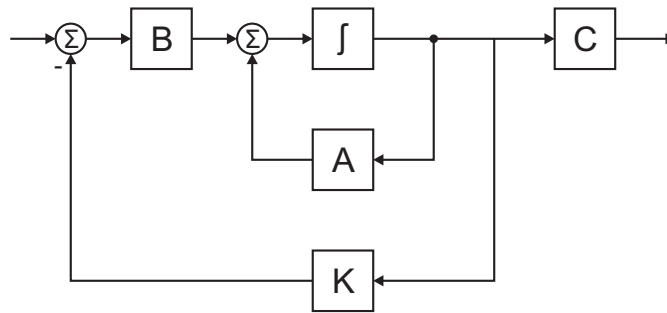
a macierz $P^\top = P \geq 0$ jest rozwiązaniem algebraicznego równania Riccatiego (ang. *Algebraic Riccati Equation* (ARE))

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0. \quad (2.20)$$

Przy ww. założeniach istnieje dokładnie jedno, półdefinitnie określone, rozwiązanie algebraicznego równania Riccatiego, czyli jedno P .

Regulator liniowo – kwadratowy jest więc optymalnym w sensie wskaźnika jakości (2.17) regulatorem od stanu.

Schemat układu regulacji z regulatorem od stanu został przedstawiony na rysunku 2.1.



Rysunek 2.1: Schemat blokowy układu regulacji z regulatorem od stanu

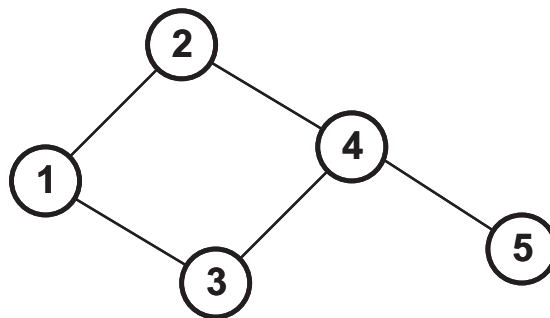
2.3 Opis matematyczny zespołu robotów

Do matematycznego opisu interakcji w zespole robotów mobilnych wykorzystuje się pojęcia związane z teorią grafów.

Definicja 2.14. (Graf) Graf \mathcal{G} to para złożona ze zbioru wierzchołków \mathbf{V} i zbioru \mathbf{E} krawędzi łączących wierzchołki. Zbiór wierzchołków \mathbf{V} jest niepusty i skończony. Zbiór krawędzi \mathbf{E} tworzą pary elementów ze zbioru wierzchołków $\mathbf{E} \subseteq \{\{u, v\} : u, v \in \mathbf{V}\}$ [93, s. 19].

Jeśli para krawędzi jest nieuporządkowana, to graf nazywa się prostym (nieskierowanym).

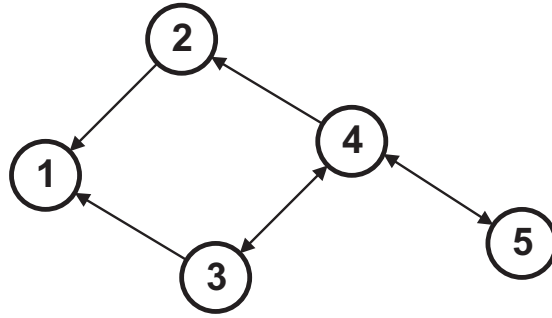
Na rysunku 2.2 przedstawiony przykładowy graf składający się z 5 wierzchołków i 5 krawędzi.



Rysunek 2.2: Przykład grafu (prostego)

Definicja 2.15. (Graf skierowany) Grafem skierowanym \mathcal{D} nazywa się parę złożoną ze zbioru wierzchołków \mathbf{V} i zbioru łączących je krawędzi \mathbf{E} , stanowiących uporządkowane pary. Często graf skierowany nazywany jest **digrafem** (od ang. *directed graph*) [93, s. 135].

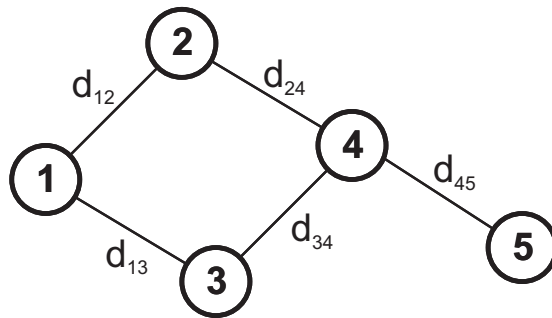
Na rysunku 2.3 przedstawiony przykładowy graf składający się z 5 wierzchołków i 7 krawędzi.



Rysunek 2.3: Przykład grafu skierowanego

Definicja 2.16. (Graf ważony) Graf ważony to graf (skierowany lub prosty), którego krawędziom nadano wagi. Jest to więc trójka $(\mathbf{V}, \mathbf{E}, \mathbf{w})$, gdzie (\mathbf{V}, \mathbf{E}) jest grafem, a $\mathbf{w} : \mathbf{E} \rightarrow \mathbf{V}$ jest funkcją określającą wagi krawędzi.

Na rysunku 2.4 przedstawiony przykładowy graf składający się z 5 wierzchołków i 5 krawędzi z wagami d_{ij} , gdzie (i, j) oznacza krawędź, do której jest przypisana dana waga.



Rysunek 2.4: Przykład grafu ważonego

Definicja 2.17. (Macierz sąsiedztwa) Macierz sąsiedztwa to kwadratowa macierz o rozmiarze równym liczbie wierzchołków grafu, której elementy oznaczają czy wierzchołki grafu sąsiadują ze sobą, tj. czy istnieje łącząca je krawędź.

Dla grafu przedstawionego na rysunku 2.2 macierz sąsiedztwa ma następującą postać:

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.21)$$

Dla grafów prostych (nieskierowanych) macierz sąsiedztwa jest macierzą symetryczną.

Definicja 2.18. (Macierz stopni) Macierz stopni to diagonalna macierz o rozmiarze równym liczbie wierzchołków grafu, której elementy oznaczają stopień wierzchołka grafu. Stopień wierzchołka jest zdefiniowany jako liczba krawędzi incydujących do wierzchołka (takich, które mają w tym wierzchołku swój koniec lub początek).

Dla grafu przedstawionego na rysunku 2.2 macierz stopni ma następującą postać:

$$\Delta(\mathcal{G}) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.22)$$

Dla grafów skierowanych definiuje się macierz stopni wejściowych i macierz stopni wyjściowych biorąc pod uwagę liczbę krawędzi mających odpowiednio koniec lub początek w danym wierzchołku. W pracy przyjęto, że $\Delta(\mathcal{D})$ oznacza macierz stopni wejściowych.

Definicja 2.19. (Laplasjan grafu) Laplasjan grafu to kwadratowa macierz o rozmiarze równym liczbie wierzchołków grafu, zdefiniowana jako:

$$L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G}). \quad (2.23)$$

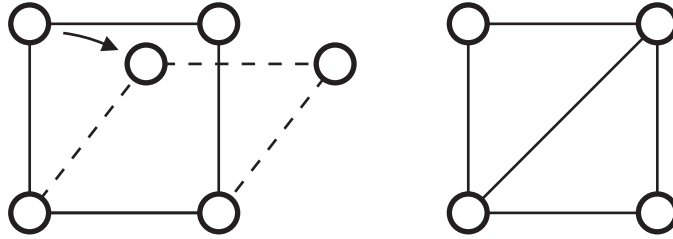
Dla grafu przedstawionego na rysunku 2.2 laplasjan grafu ma następującą postać:

$$L(\mathcal{G}) = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}. \quad (2.24)$$

Dla grafów skierowanych definiuje się laplasjan wejściowy i wyjściowy – analogicznie do macierzy stopni. Konsekwentnie z przyjętą definicją macierzy stopni dla digrafów (definicja 2.18.), przez laplasjan digrafu $L(\mathcal{D})$ oznacza się laplasjan wejściowy.

Definicja 2.20. (Strukturalna sztywność grafu) Strukturalna sztywność grafu (ang. *graph rigidity*) jest koncepcją przeniesioną z analizy struktur i zastosowaną w algebraicznym opisie grafu. W poniższym opisie, dla uproszczenia, rozważane są struktury na płaszczyźnie \mathbb{R}^2 . Definicja została opisana m.in w [75].

Idea polega na geometrycznej interpretacji grafu. Jeśli krawędzie grafu zostaną potraktowane jako poprzeczki łączące sąsiadujące wierzchołki, to sztywność może zostać zinterpretowana jako odporność powyższej struktury na deformację. Inaczej mówiąc, jeśli nie jest możliwe ciągle przemieszczenie wierzchołków grafu, z zachowaniem odległości pomiędzy wierzchołkami, bez poruszania całej struktury, to taki graf nazywamy strukturalnie sztywnym. Sytuacja ta została zobrazowana na rysunku 2.5. Wierzchołki grafu po lewej stronie mogą zostać przesunięte bez zmiany wzajemnych odległości. Taki graf nie jest strukturalnie sztywny. W przypadku grafu po prawej stronie, każde przemieszczenie, które zachowuje odległości pomiędzy wierzchołkami, przemieszcza cały graf. Do takich transformacji można zaliczyć translację i obrót.



Rysunek 2.5: Strukturalna sztywność grafu. Po lewej: graf strukturalnie giętki, po prawej: graf strukturalnie sztywny (na podstawie [75])

Formalny opis matematyczny sztywności grafu polega na definicji struktury, złożonej z pary $(\mathcal{G} = (V, E), p)$, gdzie \mathcal{G} jest grafem, a p funkcją przyporządkowującą każdemu wierzchołkowi współrzędne na płaszczyźnie, czyli $p : V \rightarrow \mathbb{R}^2$.

Struktury (\mathcal{G}, p_1) i (\mathcal{G}, p_2) są przystające (ang. *congruent*) jeśli zachodzi:

$$\|p_1(v_i) - p_1(v_j)\| = \|p_2(v_i) - p_2(v_j)\| \quad \forall v_i, v_j \in V. \quad (2.25)$$

Struktury (\mathcal{G}, p_1) i (\mathcal{G}, p_2) są równoważne (ang. *equivalent*) jeśli zachodzi:

$$\|p_1(v_i) - p_1(v_j)\| = \|p_2(v_i) - p_2(v_j)\| \quad \forall \{v_i, v_j\} \in E. \quad (2.26)$$

Struktura (\mathcal{G}, p_1) jest globalnie sztywna jeśli wszystkie struktury, które są równoważne (\mathcal{G}, p_1) są jednocześnie przystające do (\mathcal{G}, p_1) [104]. Graf \mathcal{G} będący elementem takiej struktury jest nazywany strukturalnie sztywnym.

Strukturalna sztywność jest istotną właściwością formacji robotów mobilnych bazujących na wzajemnych odległościach, ze względu na wynikającą z niej jednoznaczność definicji formacji.

3 Algorytmy sterowania zespołem robotów - przykłady wprowadzające

Rozdział stanowi wprowadzenie do głównego obszaru zainteresowania rozprawy tj. algorytmów sterowania formacją dla grupy robotów mobilnych. Jako formację robotów mobilnych należy rozumieć grupę n robotów, których wzajemne położenie jest sterowane w sposób zapewniający utrzymanie zadanych odległości pomiędzy robotami. Wygodnym narzędziem do reprezentacji i opisu zadanych odległości oraz relacji pomiędzy robotami są grafy. Taki sposób opisu wprowadzi przykład 3.1.1, ale matematyczne zalety tego opisu zostaną przedstawione w przykładzie 3.1.2. W tej reprezentacji roboty są wierzchołkami grafu, a krawędzie służą do definicji wzajemnych odległości. W przypadku użycia grafów skierowanych przepływ informacji pomiędzy robotami może następować w jednym kierunku. Oznacza to, że może zachodzić sytuacja, w której robot A utrzymuje (mierzy) odległość od robota B, ale robot B nie utrzymuje (mierzy) odległości względem robota A. W przypadku grafów nieskierowanych uznaje się, że informacja przepływa w obydwu kierunkach.

Celem przykładu 3.1.1 jest demonstracja uproszczonego modelu i użycie podstawowych pojęć opisanych w rozdziale 2 do jego analizy, w szczególności do badania stabilności opisywanego układu. W przykładzie 3.1.2 zostanie uzupełniony aparat matematyczny dając podstawy do rozważań w kolejnych rozdziałach.

3.1 Przykłady wprowadzające

3.1.1 Przykład 1

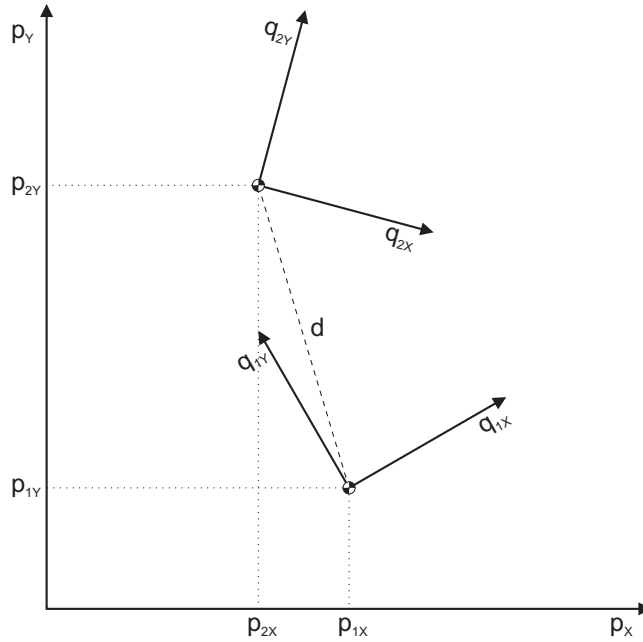
W przykładzie zostanie rozważony przypadek dwóch pojazdów (robotów) mobilnych poruszających się na płaszczyźnie. Dla uproszczenia pojazd zostanie opisany równaniem kinematycznym

$$\dot{q}_i = u_i, \quad i = 1, 2, \quad (3.1)$$

gdzie $q_i \in \mathbb{R}^2$ jest współrzędną robota w lokalnym układzie współrzędnych, a $u_i \in \mathbb{R}^2$ jest sterowaniem, które może zostać zinterpretowane jako zadana prędkość robota. Związek pomiędzy położeniem robota j w lokalnym a globalnym układem współrzędnych robota i opisany jest przez równanie

$$q_{ij} = R_i p_j + \tau_i,$$

gdzie R_i jest macierzą obrotu, τ_i wektorem przesunięcia (translacji), a $p_j \in \mathbb{R}^2$ jest położeniem robota (Rys. 3.1).



Rysunek 3.1: Zależność pomiędzy lokalnymi układami współrzędnych robotów a globalnym, zewnętrznym układem współrzędnych

Równanie ruchu robota w globalnym układzie jest opisane w następujący sposób

$$\dot{p}_i = R_i^{-1} \dot{q}_{ii} = R_i^{-1} u_i. \quad (3.2)$$

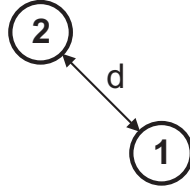
Zadaniem robotów jest utrzymanie stałej odległości pomiędzy sobą wykorzystując jedynie informację o wzajemnym położeniu sąsiada w lokalnym układzie współrzędnych każdego robota. Takie zadanie jest określane terminem utrzymania formacji. Można zauważyć, że informacja, którą pozyskują roboty nie wymaga komunikacji pomiędzy robotami, ponadto roboty nie mają wiedzy o globalnym układzie współrzędnych. Mierzone przez pojazdy względne położenie można zapisać jako

$$\begin{aligned} r_1 &= q_{11} - q_{12} = R_1(p_1 - p_2) \\ r_2 &= q_{22} - q_{21} = R_2(p_2 - p_1). \end{aligned} \quad (3.3)$$

Jeśli pożądana odległość pomiędzy robotami zostanie oznaczona jako d to różnica pomiędzy aktualną odległością, a odległością zadaną (uchyby) ma postać

$$\begin{aligned} e_1 &= \|R_1 r_1\|^2 - d^2 \\ e_2 &= \|R_2 r_2\|^2 - d^2. \end{aligned} \quad (3.4)$$

Graf, który opisuje powyższą zależność przedstawiono na rys. 3.2. W definicji uchybu użyto normy i odległości podniesionych do kwadratu ze względu na istnienie pochodnej tego wyrażenia w \mathbb{R}^2 .



Rysunek 3.2: Graf reprezentujący zadanie utrzymania stałej odległości przez dwa pojazdy. Strzałki symbolizują dwustronną wymianę informacji, ale należy zauważyć, że przedstawiony graf jest nieskierowany

Ze względu na ortogonalność macierzy obrotu oraz fakt, że $r_1 = -r_2$ i wynikającą z tego zależność $\|r_1\| = \|-r_2\| = \|r\|$ równania (3.4) mogą zostać uproszczone do

$$e = \|r\|^2 - d^2. \quad (3.5)$$

Naturalnymi kandydatami na sterowanie dla poszczególnych robotów są

$$\begin{aligned} u_1 &= -r_1 e \\ u_2 &= -r_2 e. \end{aligned} \quad (3.6)$$

Taka postać sterowania ma następującą interpretację. Pierwszy człon określa kierunek ruchu związany z sąsiednim robotem. Drugi człon jest uchybem sterowania - przyjmuje wartość ujemną, gdy pojazdy są zbyt blisko siebie, wartość dodatnią, gdy są zbyt daleko, oraz wartość 0, gdy pojazdy są w żądanej odległości od siebie.

Po podstawieniu równania (3.6) do równania ruchu otrzymuje się następujące równanie układu zamkniętego

$$\begin{aligned} \dot{p}_1 &= -r_1 e = -(p_1 - p_2)(\|p_1 - p_2\|^2 - d^2) \\ \dot{p}_2 &= -r_2 e = -(p_2 - p_1)(\|p_2 - p_1\|^2 - d^2). \end{aligned} \quad (3.7)$$

Otrzymany system dynamiczny jest stacjonarny i nieliniowy. Mimo lokalnych układów współrzędnych, które mają różne orientacje, badanie układu może zostać przeprowadzone w globalnym układzie współrzędnych. Przedmiotem dalszej analizy jest stabilność systemu (3.7). W celu wykazania asymptotycznej stabilności układu rozważmy system związany z (3.7), opisujący dynamikę względnych położenia pojazdów.

$$\begin{aligned} \dot{r}_1 &= -r_1 e + r_2 e \\ \dot{r}_2 &= -r_2 e + r_1 e. \end{aligned} \quad (3.8)$$

Wykorzystując zależność $r = r_1 = -r_2$ równanie (3.8) można uprościć do

$$\dot{r} = -2re. \quad (3.9)$$

Punkty równowagi systemu (3.8) spełniają zależność

$$r_1e - r_2e = 0.$$

Wynika z tego, że układ (3.8) posiada dwa rozłączne zbiory punktów równowagi. Pierwszy:

$$\mathcal{R} = \{p : r_i = 0, \quad \forall i\}, \quad (3.10)$$

opisujący sytuację, w której położenie robotów jest sobie równe (roboty zajmują to samo miejsce na płaszczyźnie). Oraz drugi:

$$\mathcal{E} = \{p : e = 0\}, \quad (3.11)$$

przedstawiający położenie robotów w zadanej odległości, czyli cel zadania. W kolejnych krokach zostanie pokazane, że zbiór \mathcal{R} jest zbiorem niestabilnych punktów równowagi, a zbiór \mathcal{E} zbiorem asymptotycznie stabilnych punktów równowagi. Ponadto pokazane zostanie, że obszar atrakcji zbioru \mathcal{E} to $\mathbb{R}^4 \setminus \mathcal{E}$. W celu badania stabilności punktów równowagi należących do zbioru \mathcal{E} można wykorzystać drugą metodę Lapunowa i zaproponować funkcję $V : \mathbb{R}^2 \rightarrow \mathbb{R}$ postaci

$$V = \frac{1}{2}e^2. \quad (3.12)$$

Pochodna funkcjonału na trajektoriach systemu (3.8) wynosi

$$\dot{V} = e\dot{e} = -4\|re\|^2, \quad (3.13)$$

ponieważ, korzystając z definicji (3.8) oraz zależności $r_1 = -r_2$ otrzymuje się:

$$\dot{e} = \dot{e}_1 = \frac{d}{dt}\|r_1\|^2 = \dot{r}_1^T r_1 + r_1^T \dot{r}_1 = -2r_1^T r_1 e + 2r_1^T r_2 e = -4\|r_1\|^2 e. \quad (3.14)$$

Funkcjonał (3.12) jest ciągłą, nieujemną funkcją zdefiniowaną w otoczeniu punktów równowagi systemu (3.8). Pochodna funkcjonału (3.13) jest ujemna dla wszystkich $r \notin \mathcal{R} \cup \mathcal{E}$, czyli dla wszystkich punktów poza punktami równowagi systemu. Pod-

sumowując można zauważyć, że:

$$\begin{aligned} V &> 0 && \forall r_i \in \mathbb{R}^2 \setminus \mathcal{E}, \quad i = 1, 2 \\ V &= 0 && \forall r_i \in \mathcal{E} \\ \dot{V} &< 0 && \forall r_i \in \mathbb{R}^2 \setminus \mathcal{R} \cup \mathcal{E}, \quad i = 1, 2. \end{aligned}$$

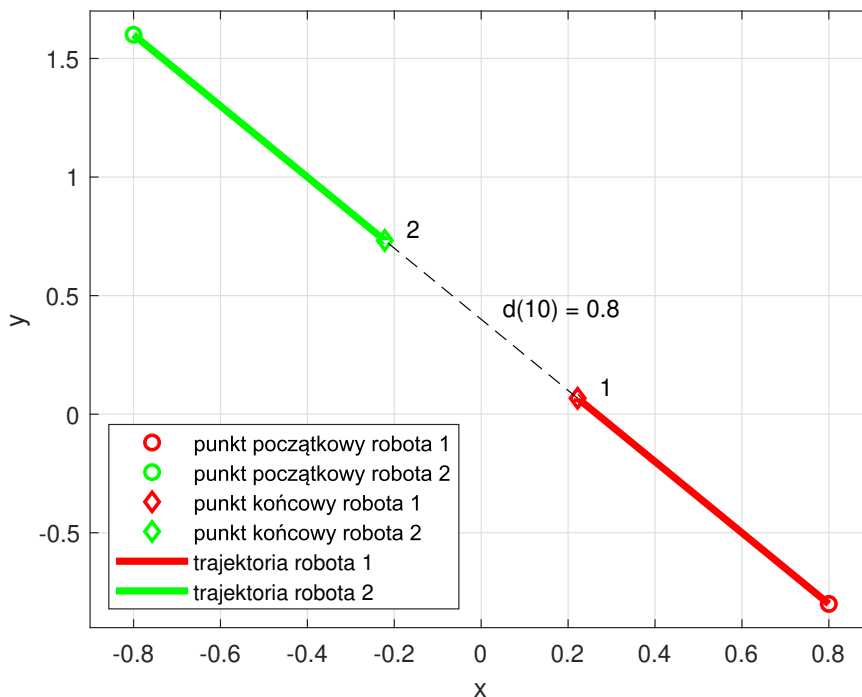
Oznacza to, że system rozszerzony (3.8) jest asymptotycznie stabilny w otoczeniu punktów równowagi ze zbioru \mathcal{E} , a trajektorie systemu zdyżają do zbioru $\mathcal{R} \cup \mathcal{E}$, gdy $t \rightarrow \infty$. Z definicji tego układu względem systemu wyjściowego (3.7) wynika, że jeśli stan początkowy robotów jest inny niż zbieżne położenie to odległość pomiędzy robotami osiąga stałą, zadaną wartość czyli $\|p_2 - p_1\| \rightarrow d$ gdy $t \rightarrow \infty$.

Powyższa analiza dowodzi, że dwa roboty opisane jako punkty kinematyczne, posiadając tylko i wyłącznie informację o swoim wzajemnym położeniu w lokalnym układzie współrzędnych mogą osiągnąć zadaną formację przez odpowiedni dobór algorytmu sterującego.

Na wykresach przedstawiono wyniki symulacyjne dla dwóch przypadków:

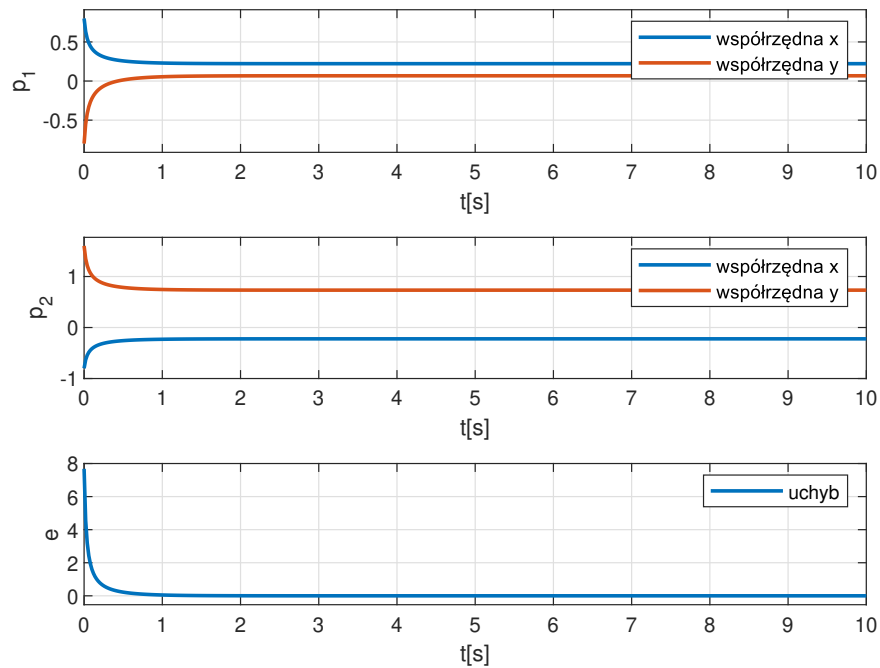
- pozycja początkowa pojazdów większa niż zadana,
- pozycja początkowa pojazdów mniejsza niż zadana.

Na rysunku 3.3 oraz 3.4 przedstawiono ruch robotów na płaszczyźnie XY oraz trajektorie ruchu.



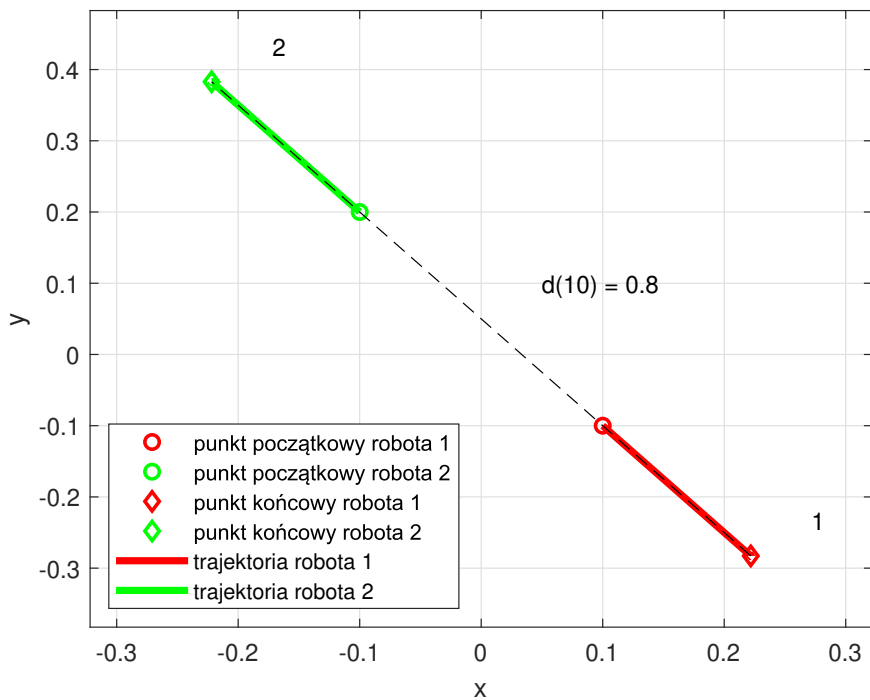
Rysunek 3.3: Ruch robotów zaznaczony na płaszczyźnie XY

Przy początkowym położeniu pojazdów skutkującej większą niż zadana odległość, pojazdy zbliżają się do siebie na zadaną odległość. Uchyb e asymptotycznie zmierza do zera.

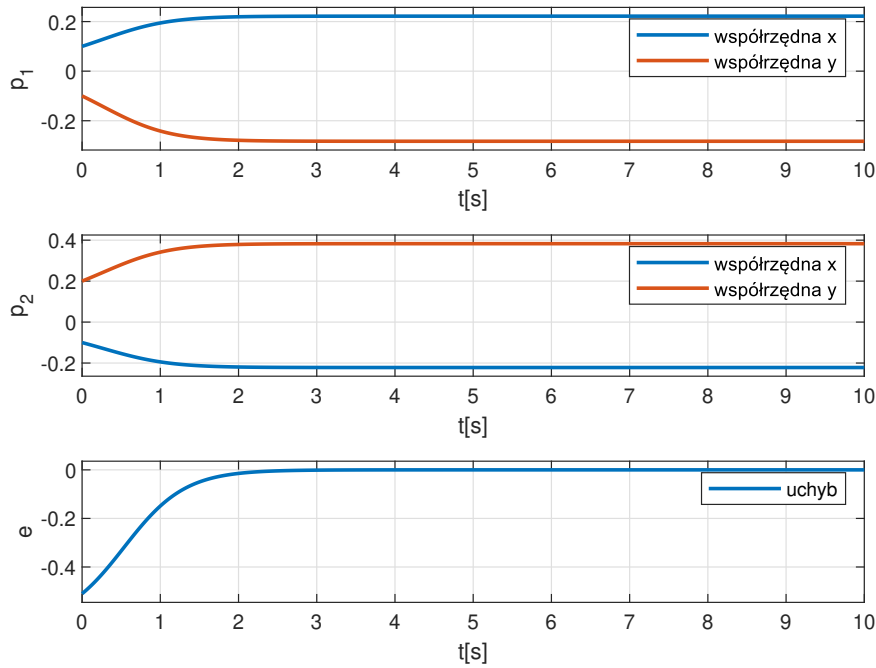


Rysunek 3.4: Przebieg położenia oraz wzajemnej odległości robotów 1 i 2

Na rysunku 3.5 oraz 3.6 przedstawiono ruch robotów na płaszczyźnie XY oraz trajektorie ruchu, gdy z początkowych pozycji wynika odległość mniejsza niż zadana odległość pomiędzy robotami.



Rysunek 3.5: Ruch robotów zaznaczony na płaszczyźnie XY



Rysunek 3.6: Przebieg położenia oraz wzajemnej odległości robotów 1 i 2

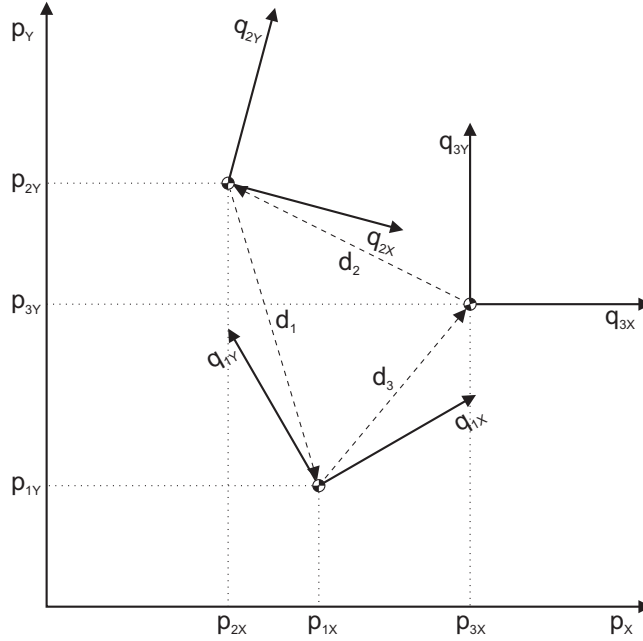
Przy początkowym położeniu pojazdów skutkującej mniejszą niż zadana odległość, pojazdy oddalają się od siebie na zadaną odległość. Tak jak w poprzednim przykładzie uchyb e asymptotycznie zmierza do zera.

Powyższe eksperymenty są zgodne z przeprowadzoną analizą.

Skrypt programu MATLAB, który pozwala na przeprowadzenie powyższych eksperymentów zamieszczono w załączniku 6.2.

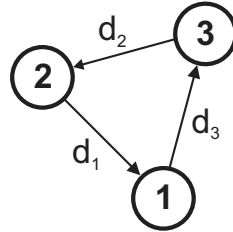
3.1.2 Przykład 2, algorytm cyklicznej pogoni dla trzech pojazdów

W przykładzie rozważona zostanie trójkątna formacja robotów, w której każdy robot śledzi położenie kolejnego sąsiada. Taki algorytm nosi nazwę cyklicznej pogoni. Do opisu tego przypadku można posłużyć się grafem skierowanym o trzech wierzchołkach i trzech krawędziach. Na rysunku 3.7 przedstawiono lokalne układy współrzędnych robotów z używanymi oznaczeniami. Zaznaczono również globalny układ współrzędnych, który jest wykorzystywany wyłącznie w celu analizy właściwości formacji robotów.



Rysunek 3.7: Zależność pomiędzy lokalnymi układami współrzędnych robotów, a globalnym, zewnętrznym układem współrzędnych

Na rysunku 3.8 przedstawiono graf symbolizujący wymianę informacji w opisywanym układzie. Przykładowo, pierwszy robot otrzymuje (lub mierzy) pozycję robota drugiego. Zadana odległość pomiędzy robotem pierwszym a drugim wynosi d_1 , itd.



Rysunek 3.8: Graf przedstawiający wymianę informacji w algorytmie cyklicznej pogoni

Zakłada się, że odległości spełniają nierówność trójkąta, czyli zachodzi:

$$d_1 + d_2 > d_3, \quad d_1 + d_3 > d_2, \quad d_2 + d_3 > d_1. \quad (3.15)$$

Pozycja robotów w lokalnym układzie współrzędnych robota i jest opisana przez zależność $q_{ij} = R_i p_j + \tau_i$.

Tak jak w przykładzie 3.1.1 zakłada się, że dynamika robotów jest opisana przez równanie:

$$\dot{q}_{ii} = u_i, \quad i \in \{1, 2, 3\}, \quad (3.16)$$

które w globalnym układzie współrzędnych może zostać zapisane jako:

$$\dot{p}_i = R_i^{-1} u_i \quad i \in \{1, 2, 3\}, \quad (3.17)$$

gdzie $u_i \in \mathbb{R}^2$ jest sygnałem sterującym.

W celu dalszej analizy definiuje się wektory różnic położzeń:

$$\begin{aligned} r_1 &= p_1 - p_2 \\ r_2 &= p_2 - p_3 \\ r_3 &= p_3 - p_1. \end{aligned} \tag{3.18}$$

Dla skrócenia zapisu wprowadza się symbol indeksu $[i] = (i + 1) \bmod 3$, $i \in \{1, 2, 3\}$. Po jego zastosowaniu równanie (3.18) przyjmuje postać:

$$r_i = p_i - p_{[i]}. \tag{3.19}$$

Jednocześnie w związku z równaniem (3.18) zachodzi:

$$r_1 + r_2 + r_3 = 0. \tag{3.20}$$

Podobnie jak w przykładzie 3.1.1, sterowanie ma postać

$$u_i = -R_i r_i e_i, \quad i \in \{1, 2, 3\}, \tag{3.21}$$

gdzie e_i są różnicami kwadratów pomiędzy oczekiwanych wartości odległości, a aktualną odległością pomiędzy sąsiadami:

$$e_i = \|R_i r_i\|^2 - d_i^2, \quad i \in \{1, 2, 3\}. \tag{3.22}$$

Korzystając z własności macierzy rotacji (R_i jest ortogonalna) równanie (3.22) upraszcza się do:

$$e_i = \|r_i\|^2 - d_i^2, \quad i \in \{1, 2, 3\}. \tag{3.23}$$

Równania ruchu formacji (3.17) po zamknięciu pętli sprzężenia zwrotnego ze sterowaniem opisanym za pomocą (3.21) mogą zostać zapisane następująco:

$$\dot{p}_i = -r_i e_i. \tag{3.24}$$

W celu analizy stabilności wygodnie jest zdefiniować układ opisujący dynamikę wzajemnych odległości pomiędzy pojazdami:

$$\dot{r}_i = -r_i e_i + r_{[i]} e_{[i]}. \tag{3.25}$$

Punkty równowagi układu (3.24) spełniają zależność:

$$r_i e_i = 0, \quad i \in \{1, 2, 3\}. \quad (3.26)$$

Z powyższego wynika, że zbiór punktów równowagi układu (3.24) może zostać zdefiniowany jako suma zbiorów $\mathcal{E} \cup \mathcal{Z}$ gdzie zbiór

$$\mathcal{E} = \{p_i : e_i = 0, \quad \forall i \in \{1, 2, 3\}\}, \quad (3.27)$$

opisuje sytuację, w której wszystkie dystanse pomiędzy robotami są równe oczekiwanym odległościom (czyli osiągnięto cel sterowania) oraz zbiór

$$\mathcal{Z} = \mathcal{C} \cup \mathcal{Q}, \quad (3.28)$$

składający się ze zbioru

$$\mathcal{C} = \{p_i : r_i = 0, \quad \forall i \in \{1, 2, 3\}\}, \quad (3.29)$$

który opisuje sytuację trzech robotów zajmujących to samo miejsce na płaszczyźnie oraz zbioru

$$\mathcal{Q} = \bigcup_{i=1}^3 \{p_i : r_i = 0 \wedge e_{[i]} = 0 \wedge e_{[[i]]} = 0, \quad \forall i \in \{1, 2, 3\}\}, \quad (3.30)$$

opisujący sytuację, w której dwa roboty znajdują się w tym samym miejscu, a trzeci jest w zadanej odległości.

Ponieważ nie jest możliwy przypadek $e_i = 0$ tylko dla jednego i , zbiór $\mathcal{Z} \cup \mathcal{E}$ opisuje wszystkie punkty równowagi układu (3.24). Ponadto, z nierówności trójkąta (3.15) wynika, że nie jest możliwy przypadek, w którym zachodzi $e_i = 0$ dla wszystkich i jeśli dwa roboty zajmują to samo miejsce na płaszczyźnie. Oznacza to, że zbiory \mathcal{E} oraz \mathcal{Z} są rozłączne.

W pracy [79] przedstawiono dowód asymptotycznej stabilności punktów równowagi należących do zbioru \mathcal{E} , natomiast w pracach [15, 16] uzyskano wynik wykładniczej stabilności oraz dokonano dodatkowych uogólnień na inne postacie sterowania. W powyższych publikacjach dowiedziono również, że każde początkowe położenie robotów poza współliniowym zbiega do zadanej formacji. Poniżej zamieszczono główne punkty dowodu przedstawionego w [15].

Po pierwsze można zauważyć, że rozwiązania układu (3.24) oraz układu (3.25) wraz z ograniczeniem (3.20) istnieją i są jednoznaczne ze względu na spełniony

warunek Lipschitza prawej strony równania (zachodzi nawet warunek mocniejszy: ciągłość pierwszych pochodnych).

Następnie do analizy stabilności punktów równowagi ze zbioru \mathcal{E} można użyć funkcji zdefiniowanej jako

$$V = e_1^2 + e_2^2 + e_3^2. \quad (3.31)$$

Pochodna funkcji (3.31) na trajektoriach systemu (3.25) wynosi

$$\begin{aligned} \dot{V} &= e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 = \\ &= -2[(r_1^\top r_1 e_1^2 - r_1^\top r_2 e_1 e_2) + (r_2^\top r_2 e_2^2 - r_2^\top r_3 e_2 e_3) + \\ &\quad + (r_3^\top r_3 e_3^2 - r_3^\top r_1 e_3 e_1)] = \\ &= -\|e_1 r_1 - e_2 r_2\|^2 - \|e_2 r_2 - e_3 r_3\|^2 - \|e_3 r_3 - e_1 r_1\|^2 \end{aligned} \quad (3.32)$$

W powyższych obliczeniach wykorzystano zależność:

$$\|a - b\|^2 = (a - b)^\top (a - b) = a^\top a + b^\top b - 2a^\top b = \|a\|^2 + \|b\|^2 - 2a^\top b \quad (3.33)$$

Ponieważ funkcja V jest:

- różniczkowalna z ciągłymi pochodnymi,
- półdodatnio określona,
- jej pochodna na trajektoriach systemu jest niedodatnia,

to jest ona funkcjonałem Lapunowa. Zgodnie z twierdzeniem Lapunowa wszystkie trajektorie układu (3.25) są ograniczone:

$$\sum_{i=1}^3 (\|r_i\|^2 - d_1^2)^2 = \sum_{i=1}^3 e_i^2 \leq \sum_{i=1}^3 e_i^2(0) = \sum_{i=1}^3 (\|r_i(0)\|^2 - d_1^2)^2 \quad (3.34)$$

Ograniczone trajektorie układu (3.25) nie oznaczają jeszcze zbieżności trajektorii układu (3.24), czyli nie gwarantują, że układ osiągnie zadaną formację. Celem dalszej analizy jest wskazanie warunków, w których zachodzi powyższe.

Do dalszych rozważań potrzebna jest definicja zbioru opisującego położenie współliniowe. Zbiór ten zostanie oznaczony jako \mathcal{N} . Można zauważyć, że zawiera on w sobie zbiór \mathcal{Z} . Formalnie jest to zbiór punktów w \mathbb{R}^6 dla których zachodzi:

$$\mathcal{N} = \{p_i : \text{rz}[r_1 \ r_2 \ r_3]_{2 \times 3} < 2, \quad r_1 + r_2 + r_3 = 0\} \quad (3.35)$$

W związku z nierównością trójkąta (3.15) zbiór \mathcal{N} , podobnie jak zbiór \mathcal{Z} , jest rozłączny ze zbiorem \mathcal{E} . Wykorzystując zależność (3.20) zbiór \mathcal{N} można opisać w na-

stępujący sposób:

$$\mathcal{N} = \{p_i : \det[r_1 \ r_2] = 0\}. \quad (3.36)$$

Zbiór \mathcal{N} jest inwariantny ponieważ na trajektoriach systemu 3.24 zachodzi:

$$\begin{aligned} \frac{d}{dt} \det[r_1 \ r_2] &= \\ &= \dot{r}_1^\top G r_2 + r_1^\top G \dot{r}_2 = \\ &= (-r_1^\top e_1 + r_2^\top e_2) G r_2 + r_1^\top G (-r_2 e_2 + r_3 e_3) = \\ &= -r_1^\top e_1 D r_2 - r_1^\top e_2 D r_2 - r_1^\top e_3 D r_2 = \\ &= -(e_1 + e_2 + e_3) \det[r_1 \ r_2]. \end{aligned} \quad (3.37)$$

W powyższym równaniu wykorzystano fakt, że dla dowolnych wektorów $a, b \in \mathbb{R}^2$ zachodzi $\det[a \ b] = a^\top G b$, gdzie:

$$G = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (3.38)$$

oraz zależności (3.20) i (3.25).

Wynik (3.37) oznacza, że jeśli $\det[r_1 \ r_2] = 0$ zachodzi dla $t = 0$ to zachodzi również dla dowolnego $t > 0$, czyli każda początkowa pozycja, która jest współliniowa pozostanie współliniowa.

Analizując proponowany funkcjonal Lapunowa w celu określenia dalszych właściwości układu, można zauważyć, że równanie (3.32) może zostać zapisane w postaci:

$$\dot{V} = -e^\top Q^\top Q e, \quad (3.39)$$

gdzie

$$Q = \begin{bmatrix} -r_1 & r_2 & 0 \\ 0 & -r_2 & r_3 \\ r_1 & 0 & -r_3 \end{bmatrix}. \quad (3.40)$$

Rząd macierzy Q jest związany następującą zależnością z wektorem różnic położzeń r_i :

$$\text{rz} Q < 3 \Leftrightarrow \text{rz}[r_1 \ r_2 \ r_3]_{2 \times 3} < 2. \quad (3.41)$$

Oznacza to, że jeśli położenia nie są współliniowe to macierz $Q^\top Q$ jest dodatnio określona. Inaczej mówiąc $Q^\top Q$ jest dodatnio określona wtedy i tylko wtedy gdy

$p \notin \mathcal{N}$.

W celu określenia tempa zbieżności zdefiniowany zostanie zbiór $\mathcal{S}(\rho)$, zależny od pewnej dodatniej stałej ρ :

$$\mathcal{S}(\rho) = \{p : e_1^2 + e_2^2 + e_3^2 < \rho, \quad r_1 + r_2 + r_3 = 0\}. \quad (3.42)$$

Można zauważyć, że $\mathcal{E} \subset \mathcal{S}(\rho)$ oraz $\mathcal{S}(\rho) \rightarrow \mathcal{E}$ gdy $\rho \rightarrow 0$. W związku z tym, że zbiory \mathcal{E} i \mathcal{N} są rozłączne, możliwe jest wybranie dostatecznie małego ρ tak aby $\mathcal{S}(\rho)$ i \mathcal{N} również były rozłączne. Niech $\hat{\mathcal{S}}(\rho)$ oznacza domknięcie zbioru $\mathcal{S}(\rho)$, a $\mu(Q^\top Q)$ oznacza najmniejszą wartość własną macierzy $Q^\top Q$:

$$\lambda = \inf_{z \in \hat{\mathcal{S}}} \mu(Q^\top Q). \quad (3.43)$$

Ponieważ macierz $Q^\top Q$ jest dodatnio określona to $\lambda > 0$. W związku z powyższym jeśli pozycja początkowa $p(0) \in \mathcal{S}(\rho)$ to zachodzi $V(0) \leq \rho$ oraz $p(t) \in \mathcal{S}(\rho)$, $t \geq 0$ (ponieważ V jest nierosnąca). Ponadto \dot{V} jest ograniczona od góry przez wyrażenie $-\lambda(e_1^2 + e_2^2 + e_3^2)$ co przekłada się na zbieżność trajektorii układu (3.24) znajdujących w zbiorze $\mathcal{S}(\rho)$ do zbioru \mathcal{E} nie wolniejszą niż $e^{(-\frac{\lambda}{4}t)}$.

Zbieżność trajektorii (3.25) oznacza, że roboty osiągają zadaną formację (odległości pomiędzy robotami osiągają pożądane wartości).

Aby pokazać obszar atrakcji zbioru \mathcal{E} należy zauważyć, że z równania (3.32) wynika, że $\dot{V} = 0$, gdy $p \in \mathcal{E}$, czyli wszystkie uchyby odległości są równe 0 lub gdy $r_1 e_1 = r_2 e_2 = r_3 e_3$, czyli wszystkie wektory prędkości są równe. Oznaczając przez \mathcal{M} zbiór:

$$\mathcal{M} = \{p : p \in \mathcal{N}, r_1 e_1 = r_2 e_2 = r_3 e_3\}, \quad (3.44)$$

można zauważyć, że \mathcal{M} i \mathcal{E} są rozłączne, co wynika z rozłączności \mathcal{N} i \mathcal{E} .

Można wykazać, że $\dot{V} = 0$ wtedy i tylko gdy $p \in \mathcal{M} \cup \mathcal{E}$. W tym celu należy zaobserwować, że jeśli zachodzi $r_1 e_1 = r_2 e_2 = r_3 e_3$ i jednocześnie $p \notin \mathcal{E}$ to przynajmniej jeden uchyb odległości $e_i \neq 0$. Zachowanie powyższej równości wymaga wtedy liniowej zależności par r_i i $r_{[i]}$, czyli zachodzi $\det [r_1 \quad r_2] = 0$. Liniowa zależność par r_i i $r_{[i]}$ zachodzi tylko w przypadku $p \in \mathcal{N}$.

Z równania (3.37) wynika, że jeśli punkt początkowy jest poza zbiorem \mathcal{N} to nie może osiągnąć tego zbioru w skończonym czasie. Jednocześnie oznacza to, że nie może osiągnąć zbioru \mathcal{M} , ponieważ zawiera się on w \mathcal{N} .

Pochodna funkcjonału Lapunowa (3.32) jest ujemna, gdy $p \notin \mathcal{M}$, co oznacza, że funkcjonal Lapunowa (3.31) monotonicznie maleje, a trajektoria systemu musi osiągnąć zbiór $\mathcal{S}(\rho)$ w skończonym czasie. Po osiągnięciu zbioru $\mathcal{S}(\rho)$ trajektoria

zbiega wykładniczo do pożądanego punktu równowagi.

Podsumowując, dla systemu opisanego równaniami $\dot{p}_i = -r_i e_i$, $i \in \{1, 2, 3\}$, gdzie $e_i = \|r_i\|^2 - d_i^2$ oraz $r_i = p_i - p_{[i]}$ wykazano, że uchyby odległości e_i zbiegają do 0 jeśli początkowe położenie nie jest współliniowe ($p(0) \notin \mathcal{N}$). Ze zbieżności e_i wynika, że położenia p_i zbiegają do ustalonych wartości. Jeśli początkowe położenie jest współliniowe to uchyby odległości e_i zbiegają do ustalonych wartości, a położenia p_i uciekają do ∞ .

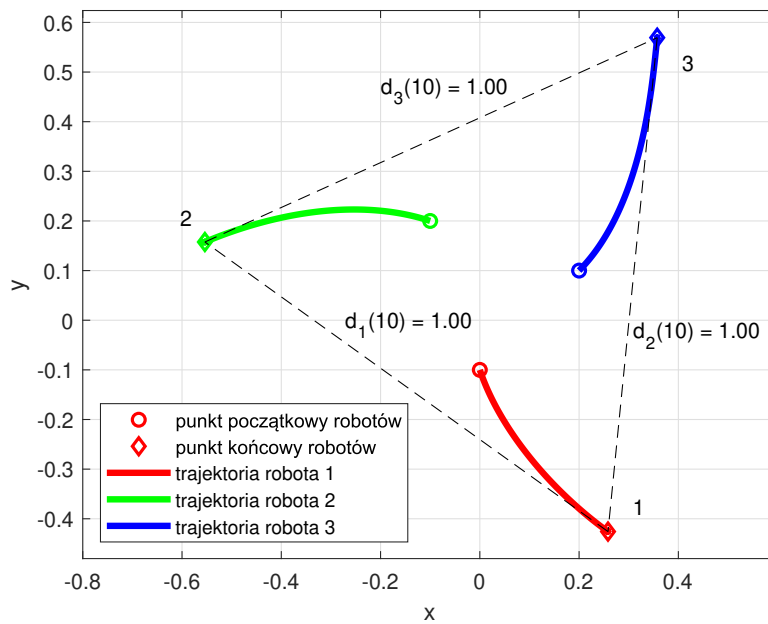
Powyższa analiza pozwala sądzić, że układ złożony z trzech robotów, które posiadają tylko i wyłącznie informacje o swoim wzajemnym położeniu w lokalnym układzie współrzędnych mogą osiągnąć zadaną formację gdy sterowanie będzie proponowanej postaci, a początkowa pozycja nie będzie współliniowa.

W celu weryfikacji powyższych rozważań przeprowadzono eksperymenty symulacyjne. Na wykresach przedstawiono wyniki dla trzech przypadków:

- początkowe uchyby odległości mniejsze niż zadane,
- początkowe uchyby odległości większe niż zadane,
- pozycja początkowa współliniowa.

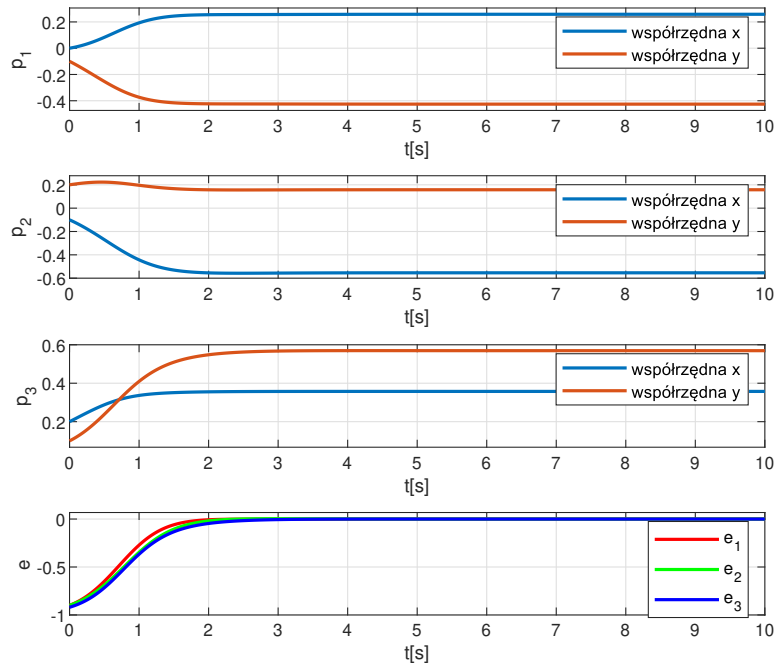
W eksperymencie przyjęto, że zadaniem pojazdów jest osiągnięcie formacji trójkąta równobocznego o boku równym 1.

Na rysunkach 3.9 oraz 3.10 przedstawiono ruch pojazdów na płaszczyźnie XY oraz trajektorie układu.



Rysunek 3.9: Ruch robotów zaznaczony na płaszczyźnie XY

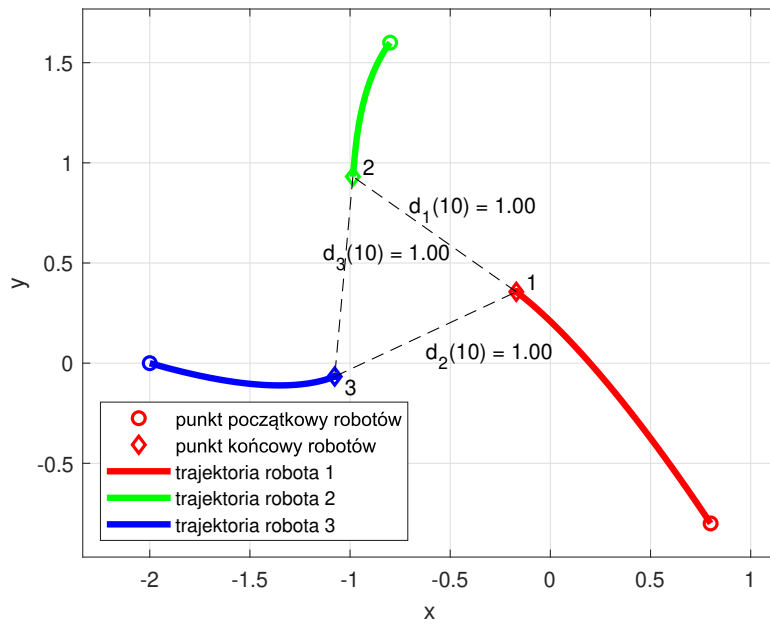
Początkowa pozycja pojazdów została oznaczona okręgami a końcowa znakiem diamentu. Kolor trajektorii odpowiada poszczególnym robotom (odpowiednio czerwony, zielony, niebieski). Kształt trajektorii tworzą łuki wynikające z definicji zadania - cyklicznej pogoni.



Rysunek 3.10: Przebieg położenia oraz uchyb odległości robotów 1, 2 i 3

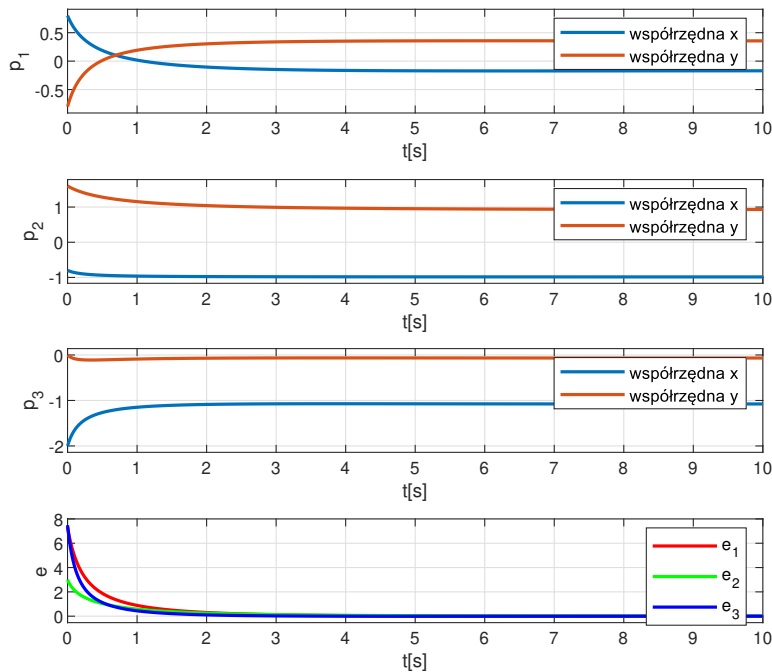
Przedstawione ilustracje obrazują zachowanie pojazdów w sytuacji gdy ich początkowa odległość jest mniejsza niż zadana. Zgodnie z oczekiwaniem roboty osiągną zadaną formację trójkąta równobocznego o boku 1. Każdy z pojazdów osiąga stałe położenie, a uchyby odległości zbiegają do 0.

Na rysunkach 3.11 oraz 3.12 przedstawiono wyniki takiego samego eksperymentu gdy początkowa pozycja robotów oznacza większe niż zadane uchyby odległości.



Rysunek 3.11: Ruch robotów zaznaczony na płaszczyźnie XY

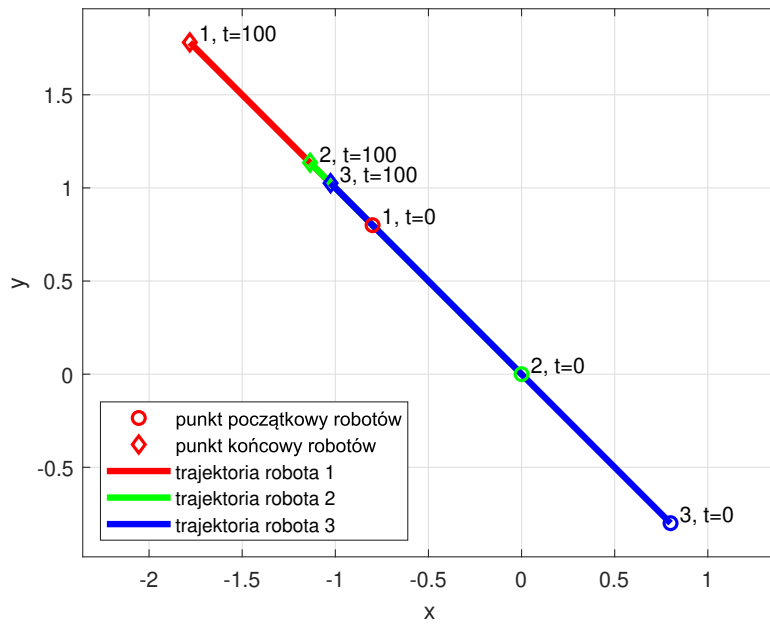
Przebieg położenia pojazdów na płaszczyźnie XY pokazuje, że roboty zbliżają się do siebie w celu osiągnięcia zadanej formacji. Na wykresie zaznaczono osiągnięte odległości pomiędzy poszczególnymi robotami w 10 sekundzie eksperymentu.



Rysunek 3.12: Przebieg położenia oraz uchyb odległości robotów 1, 2 i 3

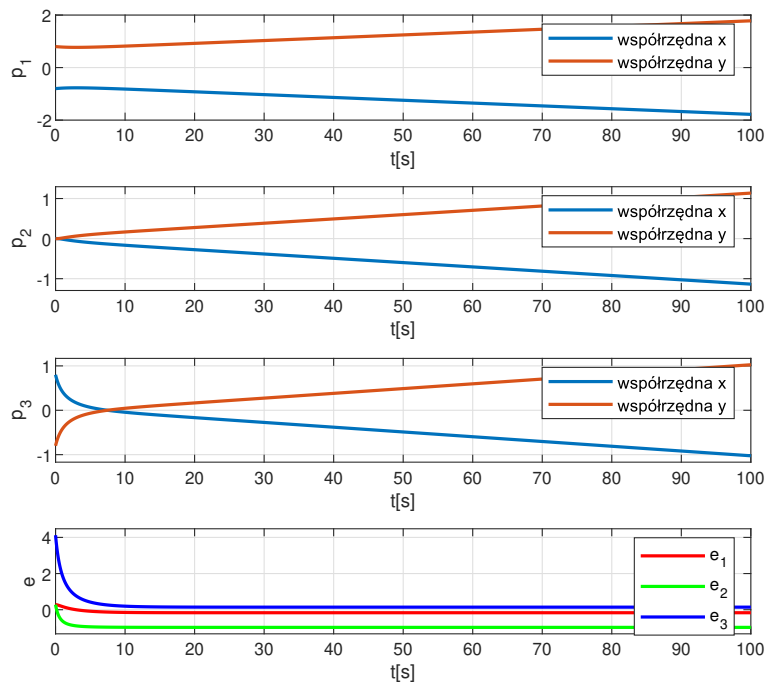
Przedstawione ilustracje obrazują zachowanie pojazdów w sytuacji gdy ich początkowa odległość jest większa niż zadana. Podobnie jak w poprzednim eksperymencie roboty osiągną zadaną formację trójkąta równobocznego o boku 1.

Na rysunkach 3.13 oraz 3.14 przedstawiono ruch pojazdów na płaszczyźnie XY oraz trajektorie układu gdy pozycja początkowa pojazdów jest współliniowa.



Rysunek 3.13: Ruch robotów zaznaczony na płaszczyźnie XY

Przebieg położenia pojazdów na płaszczyźnie XY pokazuje, że roboty pozostają w pozycji współliniowej podczas trwania eksperymentu. Dodatkowymi markerami zaznaczono pozycje początkową poszczególnych robotów oraz ich pozycję po upływie 100 sekund.



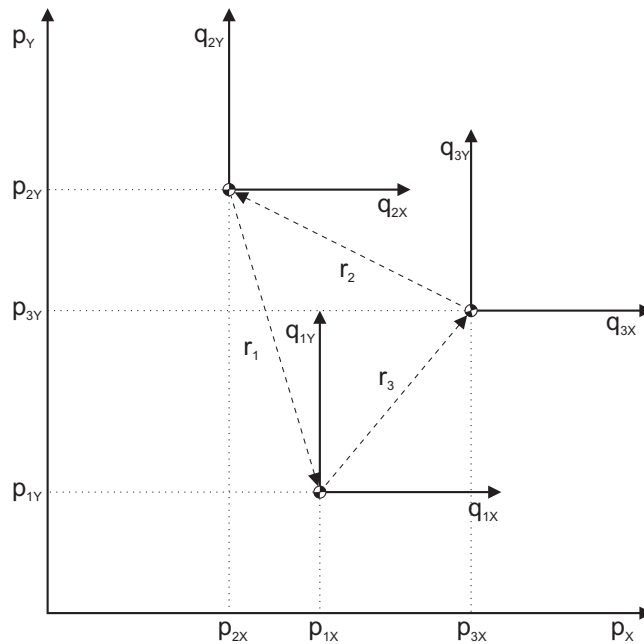
Rysunek 3.14: Przebieg położenia oraz uchyb odległości robotów 1, 2 i 3

Przedstawione ilustracje obrazują zachowanie pojazdów w sytuacji gdy ich początkowa pozycja jest współliniowa. Zgodnie z oczekiwaniami teoretycznymi pojazdy pozostają współliniowe przez cały czas trwania eksperymentu. Ich uchyby odległości osiągają ustalone, niezerowe wielkości. Pozycja robotów rozbiega się do ∞ , a prędkość robotów osiąga stałą wartość.

3.1.3 Przykład 3, formacja bazująca na wzajemnym położeniu

Celem przykładu jest zobrazowanie różnicy pomiędzy algorytmami bazującymi na odległościach, a algorytmami bazującymi na wzajemnych położeniach. Pierwszy przypadek był rozważany w dwóch poprzednich przykładach. Sterowanie bazujące na wzajemnym położeniu wymaga aby lokalne układy współrzędnych miały jednakową orientację, tak jak przedstawiono na rysunku 3.15.

Dodatkowy celem przykładu jest demonstracja elementów algebraicznej teorii grafów w opisie formacji robotów.



Rysunek 3.15: Zależność pomiędzy lokalnymi układami współrzędnych robotów, a globalnym, zewnętrznym układem współrzędnych

Podobnie jak w poprzednich przykładach, zostanie przyjęty prosty model pojazdów:

$$\dot{p}_i = u_i, \quad i = 1, 2, 3, \quad (3.45)$$

gdzie $p_i \in \mathbb{R}^2$ opisuje pozycję robota..

W odróżnieniu od metod bazujących na wzajemnej odległości, w metodzie bazującej na wzajemnym przemieszczeniu zadaniem sterowania jest osiągnięcie formacji opisanej przez zadane wektory przemieszczeń pomiędzy sąsiadami $r_{ij}^* \in \mathbb{R}^2$

Sterowanie, które pozwala osiągnąć zadaną formację ma postać

$$u_i = - \sum_{j=1,\dots,3} a_{ij}(p_i - p_j - r_{ij}^*) \quad (3.46)$$

gdzie a_{ij} są elementami macierzy sąsiedztwa, a r_{ij}^* oznaczają zadane przemieszczenia pomiędzy parami pojazdów i, j .

Przy pomocy metod algebraicznej teorii grafów można uzyskać zwarty opis dynamiki układu (np. [63]). Oznaczając przez $p = [p_1 \ p_2 \ p_3]^T \in \mathbb{R}^6$ wektor opisujący pozycje wszystkich pojazdów oraz przez $r^* \in \mathbb{R}^6$ wektor wszystkich zadanych przemieszczeń, pozwala zapisać dynamikę układu w postaci:

$$\dot{p} = (L(\mathcal{D}) \otimes \mathbb{I}_2)p - (A(\mathcal{D}) \otimes \mathbb{I}_2)r^* \quad (3.47)$$

gdzie $L(\mathcal{D})$ to laplasjan grafu opisujący wymianę informacji pomiędzy robotami, $A(\mathcal{D})$ to macierz sąsiedztwa, \otimes oznacza iloczyn Kroneckera, a \mathbb{I}_2 jest macierzą jednostkową rzędu 2.

Dla analizowanego przypadku (zdefiniowanego za pomocą grafu przedstawionego na rysunku 3.8) macierz stopni $\Delta(\mathcal{D})$ oraz macierz sąsiedztwa $A(\mathcal{D})$ mają postać:

$$\Delta(\mathcal{D}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A(\mathcal{D}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}. \quad (3.48)$$

Wynikający z powyższych laplasjan (obliczony zgodnie z definicją 2.24 ma następującą postać:

$$L(\mathcal{D}) = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}. \quad (3.49)$$

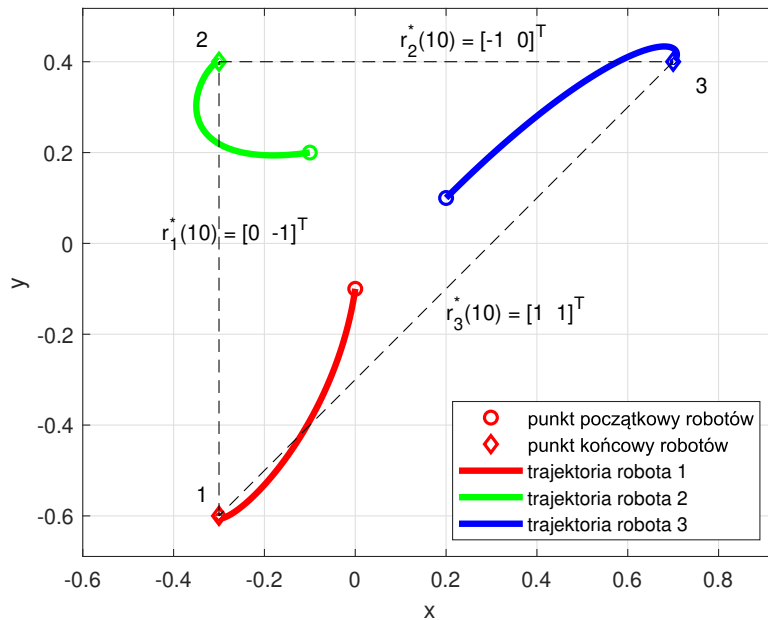
Powyższy zapis jest wygodną formą przy większej liczbie rozważanych pojazdów oraz przy implementacji algorytmów sterowania formacją.

Na rysunkach 3.16 oraz 3.17 przedstawiono wyniki eksperymentu symulacyjnego,

w którym przyjęto następujące oczekiwane przemieszczenia:

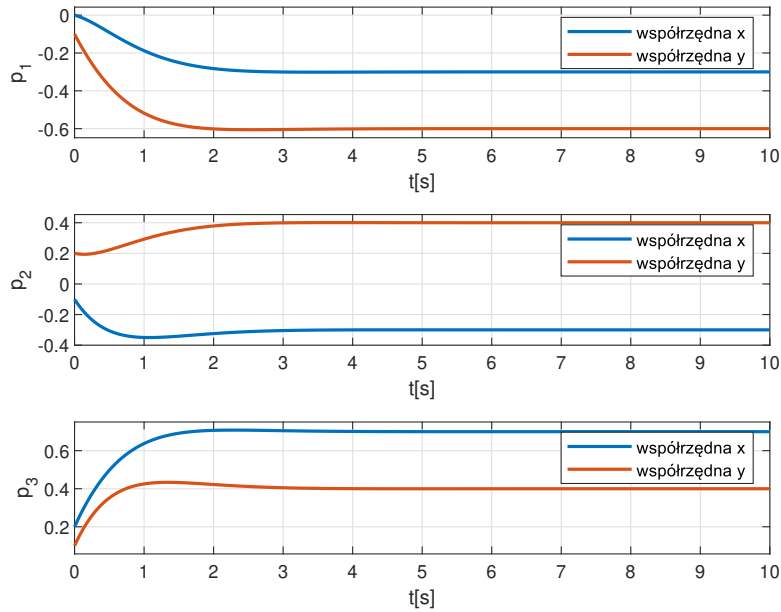
$$r_{12}^* = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad r_{23}^* = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad r_{31}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (3.50)$$

Przebieg położenia pojazdów na płaszczyźnie XY pokazuje, że roboty osiągają wzajemne położenia zgodnie z definicją. Istotną różnicą w stosunku do poprzednich przykładów jest utrzymana orientacja całej formacji zgodnie z orientacją globalnego układu współrzędnych. Względne położenia końcowe są przesuniętymi wektorami przemieszczeń.



Rysunek 3.16: Ruch robotów zaznaczony na płaszczyźnie XY

Na wykresie 3.17 przedstawiono przebieg czasowy położenia poszczególnych robotów. Wykres pokazuje, że pojazdy osiągają ustalone położenie.



Rysunek 3.17: Przebieg położenia robotów 1, 2 i 3

3.2 Podsumowanie

W rozdziale przedstawiono przykłady algorytmów sterujących dla formacji robotów mobilnych dla metod bazujących na wzajemnych odległościach i przemieszczeniach pomiędzy robotami.

Przeprowadzona analiza dowodzi, że opisywane algorytmy spełniają zadanie gdy sterowane obiekty są opisane prostymi modelami i nie mają ograniczeń ruchu.

W kolejnych rozdziałach powyższe trzy przykłady staną się punktem wyjścia do opracowania algorytmów sterowania zespołem robotów mobilnych.

W rozdziale 4 zostanie przedstawiona analiza oraz wyniki symulacyjne i eksperymentalne dla układu nieholonomicznych pojazdów. W rozdziale 5 obiektem zainteresowania będzie zespół niestabilnych robotów mobilnych.

4 Algorytmy sterowania zespołem robotów nieholonomicznych

Poniższy rozdział opisuje eksperymenty związane ze sterowaniem formacją robotów mobilnych o bardziej złożonej dynamice niż przykłady przedstawione w rozdziale 3. Przedmiotem badań są rzeczywiste roboty mobilne, a weryfikacja proponowanych metod odbywa się na ich cyfrowych odpowiednikach. W eksperymentach wykorzystano zespół robotów mobilnych umieszczonych w zdalnie dostępnym laboratorium - Robotarium.

4.1 Robotarium

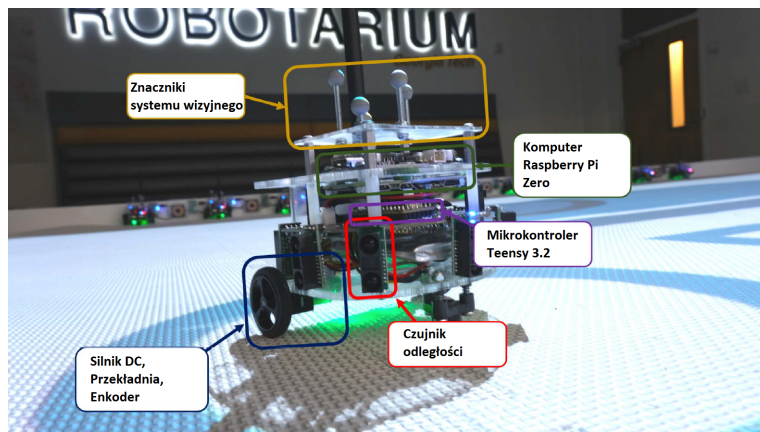
W 2017 roku uruchomione zostało laboratorium znajdujące się w Instytucie Technicznym Georgii (ang. *Georgia Institute of Technology*) (rysunek 4.1). Głównym wyposażeniem laboratorium jest zespół 20 robotów mobilnych umieszczonych na platformie o wymiarach 3,2 m x 2m. Nad platformą znajduje się zestaw 8 kamer, które umożliwiają śledzenie położenia robotów. Kamery pozwalają na odtworzenie położenia robotów z submilimetrową precyzją z częstotliwością 120 Hz zapewniając w ten sposób informację zwrotną potrzebną do sterowania robotów [70]. Do identyfikacji i wykrywania położenia robotów służą znaczniki umieszczone na górnej powierzchni robota. Ponadto dodatkowa kamera rejestruje obraz z przebiegu eksperymentu.



Rysunek 4.1: Robotarium - laboratorium robotów mobilnych o zdalnym dostępie (źródło: www.cwiconstruction.com/projects/georgia-tech-robotarium)

W laboratorium użyto robotów GRITSBot X [95], które zostały opracowane dla

laboratorium na podstawie doświadczeń nabytych przy budowie poprzedniej wersji robotów GRITSBot [70] (rysunek 4.2). Są to niedrogie pojazdy mobilne opracowane z myślą o badaniach i eksperymentach dotyczących rojów robotów. Pojazd jest wyposażony w dwa niezależnie napędzane koła umieszczone na wspólnej osi po obu stronach robota. Dodatkowo zamontowane jest koło samonastawne jako element podporowy zapobiegając przechyleniu pojazdu. Taki typ pojazdu jest określany jako robot klasy (2,0) [22] lub pochodzącym z języka angielskiego terminem *differential wheeled robot*. Pojazd o opisywanej budowie systemu napędowego posiada ograniczone możliwości ruchu. Wynikają one z mniejszej liczby dostępnych sterowań niż liczba stopni swobody. Roboty mają 11 cm szerokości, 8,5 cm długości i około 7,5 cm wysokości (nie licząc anteny).



Rysunek 4.2: GRITSBot X - robot używany w laboratorium Robotarium (na podstawie: [94])

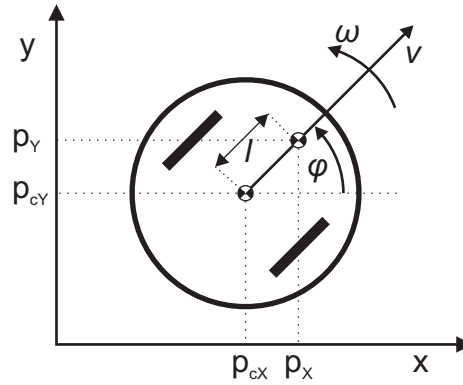
Do napędzania kół wykorzystano miniaturowe silniki DC wyposażone w przekładnię redukcyjną ($n = 100 : 1$) oraz w kwadraturowy enkoder umożliwiający pomiar pozycji i prędkości wału silnika. Zastosowany napęd pozwala na osiągnięcie prędkości liniowej około 20 cm/s oraz kątowej około 3,6 rad/s.

4.2 Przekształcenie sterowania dla robota klasy (2,0)

Proponowane algorytmy sterowania formacją zwracają wektor prędkości liniowej zakładając brak ograniczeń ruchu pojazdu. Robot klasy (2,0) nie jest w stanie podążać za takim sterowaniem w sposób bezpośredni ze względu na ograniczenia ruchu. Rozwiązaniem tego problemu jest zastosowanie odwzorowania sterowania obliczonego dla punktu kinematycznego na kinematykę robota klasy (2,0) przy pomocy przekształcenia opisanego w [64, 96] (w literaturze zwyczajowo określanego przez termin ang. *near-identity diffeomorphism*, NID).

Idea tego przekształcenia polega na definicji punktu $p = [p_x \ p_y]^T$ znajdującego na prostej prostopadłej do osi kół robota i przechodzącej przez środek robota $p_c =$

$[p_{cx} \ p_{cy}]^T$ w odległości l . Oznaczenia przedstawiono na rysunku 4.3.



Rysunek 4.3: Oznaczenia opisujące położenie i sterowania robota klasy (2,0). Zadana prędkość w ruchu liniowym została oznaczona przez v , a zadana prędkość w ruchu obrotowym przez ω .

Punkt p jest zdefiniowany jako:

$$p = p_c + l \begin{bmatrix} \cos \varphi \\ \sin \varphi \end{bmatrix}, \quad (4.1)$$

gdzie $0 < l \in \mathbb{R}$.

Popularnym sposobem opisu kinematyki robota klasy (2,0) jest postać

$$\dot{\mathbf{x}} = \begin{bmatrix} \cos(\varphi) & 0 \\ \sin(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (4.2)$$

gdzie v i ω oznaczają sygnały sterujące odpowiednio w ruchu postępowym i obrotowym, oraz $\mathbf{x} = [p_{cx}, p_{cy}, \varphi]$ określa stan robota czyli jego pozycję i orientację.

Poszukiwane przekształcenie ma za zadanie odwzorować sterowanie $u \in \mathbb{R}^2$ na sygnały v i ω .

Różniczkując równanie (4.1) otrzymuje się:

$$\dot{p} = \dot{p}_c + l\dot{\varphi} \begin{bmatrix} -\sin \varphi \\ \cos \varphi \end{bmatrix} \quad (4.3)$$

Wybierając $0 < l \in \mathbb{R}$ można dokonać odwzorowania sterowania dla punktu

kinematycznego na dynamikę robota klasy (2,0).

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = R_l^{-1}(\varphi)\dot{p} = R_l^{-1}(\varphi)u, \quad (4.4)$$

gdzie

$$R_l(\varphi) = \begin{bmatrix} \cos(\varphi) & -l \sin(\varphi) \\ \sin(\varphi) & l \cos(\varphi) \end{bmatrix}, \quad (4.5)$$

oraz $\dot{p} = u$ jest równaniem opisującym ruch pojazdu, a równanie $\dot{\varphi} = \omega$ opisuje zmianę orientacji robota. Macierz odwrotna do $R_l(\varphi)$ istnieje dla każdego φ oraz dla $l > 0$ i jest dana wzorem:

$$R_l^{-1}(\varphi) = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\frac{1}{l} \sin(\varphi) & \frac{1}{l} \cos(\varphi) \end{bmatrix}. \quad (4.6)$$

4.3 Metodologia badań

Prowadzone badania mają na celu weryfikację poprawności algorytmów sterujących formacją robotów. Zdefiniowano kilka algorytmów, w których opracowano sterowanie dla uproszczonego modelu matematycznego robotów będących punktami kinematycznymi. Opisy używanych algorytmów sterujących, w postaci równań sterowania, zostały zamieszczone przy opisie eksperymentów.

W kolejnym kroku wykorzystano transformację opisaną w podrozdziale 4.2 oraz środowisko symulacyjne z modelem robotów GRITSBot X w celu weryfikacji algorytmów dla układów o bardziej złożonej dynamice. Podobne podejście zostało opisane w [105] gdzie wykazano również, że właściwość asymptotycznej stabilności algorytmu sterowania formacją zostanie zachowana.

Następnie przesłano odpowiednio sformatowane programy sterujące do zdalnego laboratorium gdzie zostały one zrealizowane na rzeczywistych robotach. Po otrzymaniu wyników przedstawiono ich analizy w podrozdziale 4.4. Dodatkowo otrzymane nagrania z eksperymentów umieszczono na platformie YouTube, gdzie można je obejrzeć korzystając z odnośników zebranych w tabeli 6.1.

Robotarium jest wyposażone w globalny system śledzący położenie robotów oraz w serwer zarządzający ich pracą. W celu przetestowania algorytmów zdecentralizowanych w badaniach przyjęto, że roboty mają dostęp wyłącznie do lokalnych in-

formacji odnośnie swoich sąsiadów. Algorytmy wykorzystują dane o wzajemnych odległościach pomiędzy robotami, symulując w ten sposób założenie o zdecentralizowanym sterowaniu. W rzeczywistości takie dane mogłyby pochodzić z lokalnie zamontowanych czujników, co jednak zwiększyłoby koszt i rozmiar robotów. Wykorzystanie centralnego systemu stanowi znaczne uproszczenie w budowie robotów.

4.4 Eksperymenty

Poniżej przedstawiono wyniki i analizy dla różnych typów algorytmów sterowania formacją robotów mobilnych. W podrozdziałach 4.4.1 i 4.4.2 przeanalizowano wyniki związane z przykładami wprowadzającymi (opisanymi odpowiednio w sekcji 3.1.1 oraz sekcji 3.1.2) przy uwzględnieniu innej dynamiki robotów. W podrozdziale 4.4.3 przeanalizowano algorytm śledzenie lidera dla formacji złożonej z trzech robotów mobilnych, a następnie rozszerzono ten eksperyment na przypadek zmiennych w czasie zadanych dystansów pomiędzy robotami 4.4.4. W podrozdziale 4.4.5 przeanalizowano algorytmy osiągnięcia formacji dla większej liczby robotów.

4.4.1 Przykład wprowadzający – algorytm osiągnięcia zadanej odległości przez dwa roboty

Eksperyment ten jest bezpośrednim nawiązaniem do przykładu wprowadzającego (3.1.1), w którym dwa pojazdy mobilne mają za zadanie utrzymać stałą odległość pomiędzy sobą. Istotną różnicą w stosunku do pierwotnego eksperymentu jest zmiana obiektów sterowania. W przykładzie wprowadzającym pojazdy opisane były przez punkty kinematyczne, a w opisywanym eksperymencie roboty mobilne klasy (2,0). Wykonano zarówno doświadczenia symulacyjne jak i eksperymenty z rzeczywistymi robotami mobilnymi. Aby zapewnić powtarzalne warunki została zdefiniowana początkowa pozycja robotów. Ponieważ oprogramowanie zdalnego laboratorium wybiera różne roboty do wykonanie eksperymentów, ich pozycja początkowa może się różnić pomiędzy realizacjami. W związku z powyższym zarówno w symulacji jak i w badaniach rzeczywistych pierwszy etap eksperymentu polega na osiągnięciu początkowej pozycji robotów. Etap ten nie jest prezentowany na wykresach.

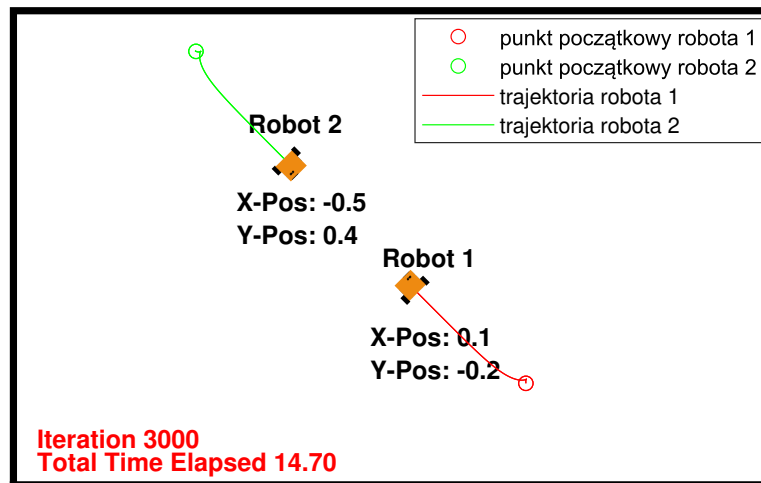
Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned} u_1 &= -(p_1 - p_2)(\|p_1 - p_2\|^2 - d^2) \\ u_2 &= -(p_2 - p_1)(\|p_2 - p_1\|^2 - d^2), \end{aligned} \quad (4.7)$$

gdzie $u_1, u_2 \in \mathbb{R}^2$ są wektorami sterującymi odpowiednio dla robota 1 i 2, $p_1, p_2 \in \mathbb{R}^2$

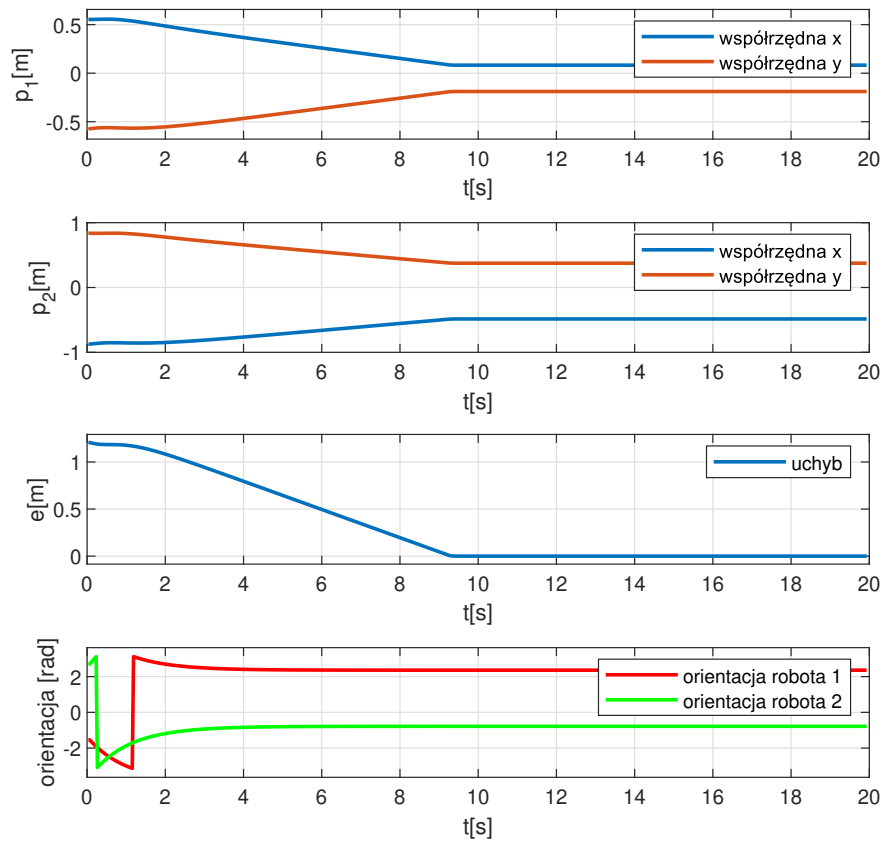
oznacza położenie robotów, a d jest zadaną odległością. W przykładzie przyjęto zadaną odległość $d = 0,8$ m. Sterowania dla obu robotów są symetryczne i różnią się zadany kierunkiem ruchu.

Rysunek 4.4 przedstawia widok symulacyjny z naniesioną końcową pozycją robotów. Trajektorie pojazdów oraz punkt początkowy zostały zaznaczone czerwoną i zieloną krzywą, odpowiednio dla robota pierwszego oraz drugiego. Czarny prostokąt przedstawia obszar platformy w zdalnym laboratorium.



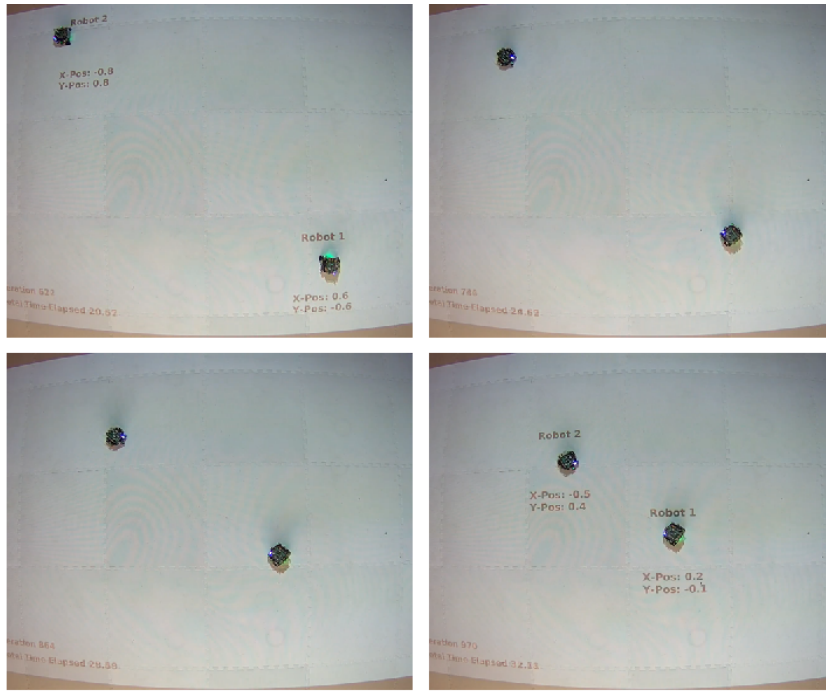
Rysunek 4.4: Ruch robotów zaznaczony na płaszczyźnie XY

Wykresy na rysunku 4.5 przedstawiają przebieg położenia robotów, odległość pomiędzy robotami oraz orientację robotów uzyskane dla eksperymentu symulacyjnego. Charakter przebiegów jest podobny do obserwowanego w symulacji dla punktów kinematycznych (rysunek 3.4). Czas osiągnięcia zadanej odległości jest znacznie dłuższy i wynosi około 9 sekund w porównaniu do 4 sekund otrzymanych w rozważaniach teoretycznych. Taka różnica wynika z ograniczenia na maksymalne sterowanie, które jest wprowadzone w tym eksperymencie symulacyjnym. Powodem ograniczenia jest jak najlepsze oddanie zachowania rzeczywistych robotów. Istotną nowością w wynikach uzyskanych dla pojazdów mobilnych używanych w tym eksperymencie jest występowanie dodatkowej zmiennej stanu opisującej roboty - orientacji. Została ona przedstawiona na dolnym wykresie na rysunku 4.5. W pierwszych sekundach eksperymentu można zaobserwować nagłą zmianę orientacji obydwu robotów wynikającą z konieczności obrócenia się pojazdów w stronę swojego sąsiada. Takie zachowanie wynika z użytej transformacji sterowania pomiędzy punktami kinematycznymi a robotem klasy (2,0).



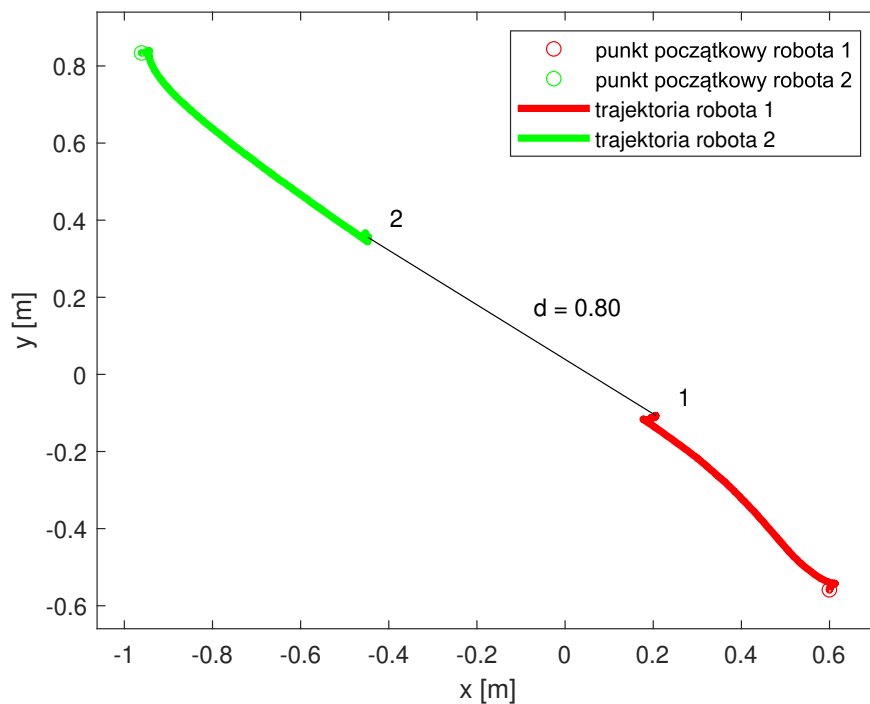
Rysunek 4.5: Przebieg położenia, uchybu odległości i orientacji robotów 1 i 2

Po weryfikacji algorytmu sterującego za pomocą narzędzi symulacyjnych wykonano eksperymenty na rzeczywistych robotach. Na rysunku 4.6 przedstawiono migawki z nagrania eksperymentu przeprowadzonego zdalnie w laboratorium Robotarium. Na czterech kadrach widać położenia robotów w początkowej chwili eksperymentu, a także po czterech, ośmiu i dwunastu sekundach. Na początku eksperymentu oraz po osiągnięciu zadanej odległości projektor umieszczony ponad platformą wyświetla informację o numerze robotów oraz ich początkowym i końcowym położeniu.



Rysunek 4.6: Migawki z nagrania rzeczywistego eksperymentu dla początku eksperymentu (lewa górna), czwartej (prawa górna), ósmej (lewa dolna) i dwunastej sekundy (prawa dolna)

Rysunek 4.7 zawiera wykres położenia robota na płaszczyźnie XY. Kolorem zielonym i czerwonym zostały przedstawione położenia robotów. Warunki początkowe zostały wybrane podobnie do tych z przykładu wprowadzającego oraz w symulacji weryfikacyjnej. We wszystkich przypadkach zadana odległość wynosi 0,8 m. Z rysunku widać, że pojazdy osiągnęły zadaną odległość.

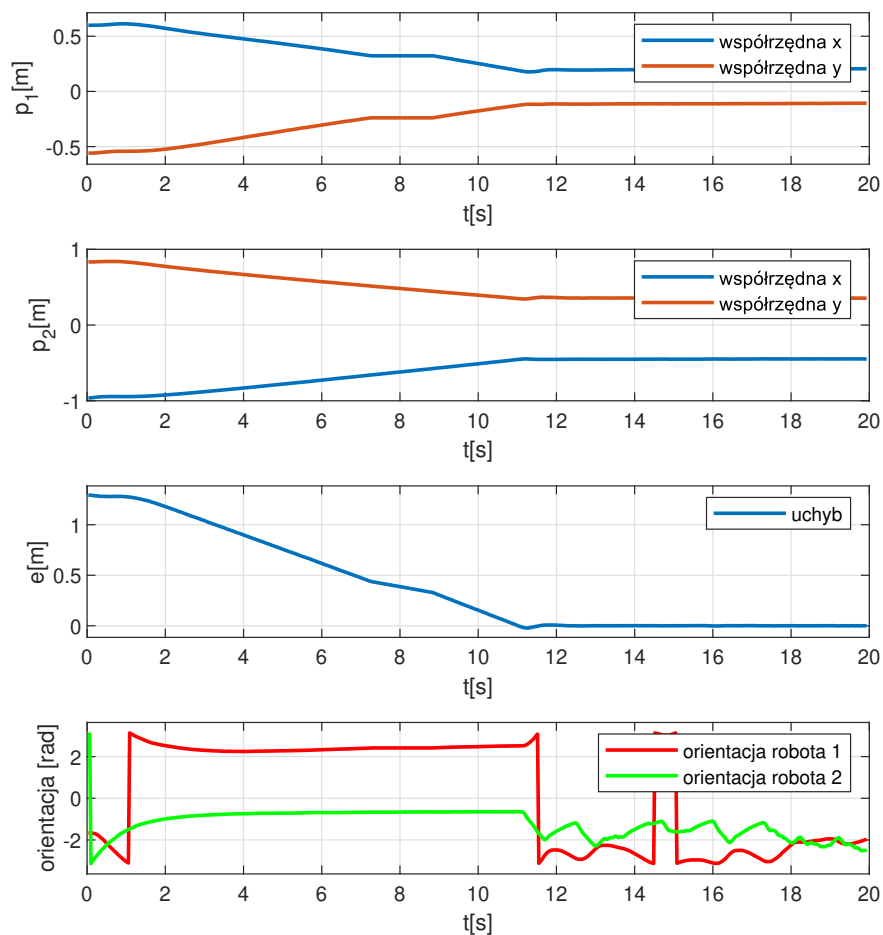


Rysunek 4.7: Ruch robotów zaznaczony na płaszczyźnie XY

Przebiegi czasowe zostały przedstawione na rysunku 4.8. Czas osiągnięcia zadanej odległości wydłużył się o około 20%, z 9 do 11 sekund. Zmiana położenia robota pierwszego nie jest jednostajna, tak jak miało to miejsce w eksperymentach symulacyjnych, a orientacja pojazdów wykazuje zwiększoną zmienność.

Eksperymenty na rzeczywistych pojazdach pokazały, że w obecności szumów oraz zjawisk, które nie są uwzględnione w modelu (np. dynamika pojazdów, tarcie, poślizg kół) czas dojścia do zadanej formacji może ulec wydłużeniu. Jednakże w każdym z przeprowadzonych eksperymentów roboty osiągały zadaną odległość pomiędzy sobą co pokazuje, że zaproponowany algorytm sterujący działa poprawnie również dla obiektów o złożonej dynamice.

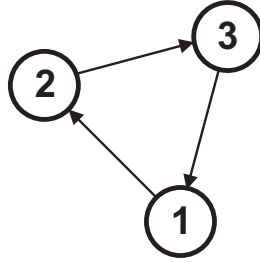
Orientacja robotów po osiągnięciu pozycji zadanej wykazuje większą zmienność niż w badaniach symulacyjnych. Wynika to z zastosowanego przekształcenia sterowania oraz szumu pomiarowego. Przy pojawiających się małych błędach położenia robot zmienia swoją orientację w celu korekty uchybu. Sama orientacja robota nie jest bezpośrednio sterowaną wielkością.



Rysunek 4.8: Przebieg położenia, wzajemnej odległości i orientacji robotów 1 i 2

4.4.2 Algorytm cyklicznej pogoni dla formacji trzech robotów

Celem eksperymentu jest zbadanie zachowania robotów w sytuacji, gdy każdy z trzech robotów śledzi położenie jednego ze swoich sąsiadów będąc jednocześnie liderem dla drugiego. Zadaniem robotów jest utworzenie formacji trójkąta równobocznego. Analiza ogólniejszego przypadku jest przedstawiona w przykładzie 3.1.2. Sytuacja ta może zostać opisana za pomocą digrafu przedstawionego na rysunku 4.9.



Rysunek 4.9: Digraf opisujący wymianę informacji w przypadku trzech robotów będących jednocześnie liderem dla jednego sąsiada i podążającym za drugim. Strzałkami zaznaczono kierunek wymiany informacji, np. robot 2 mierzy i podąża za położeniem robota 1.

Ze względu na jednakową zadaną odległość pomiędzy każdą parą, informacja ta została pominięta na rysunku. Dla zobrazowania różnicy zachowania formacji, kierunek wymiany informacji jest przeciwny niż w przykładzie 3.1.2.

Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned}
 u_1 &= -(p_1 - p_3)(\|p_1 - p_3\|^2 - d^2) \\
 u_2 &= -(p_2 - p_1)(\|p_2 - p_1\|^2 - d^2) \\
 u_3 &= -(p_3 - p_2)(\|p_3 - p_2\|^2 - d^2)
 \end{aligned} \tag{4.8}$$

gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, a $d = 0,8$ m jest zadaną odległością.

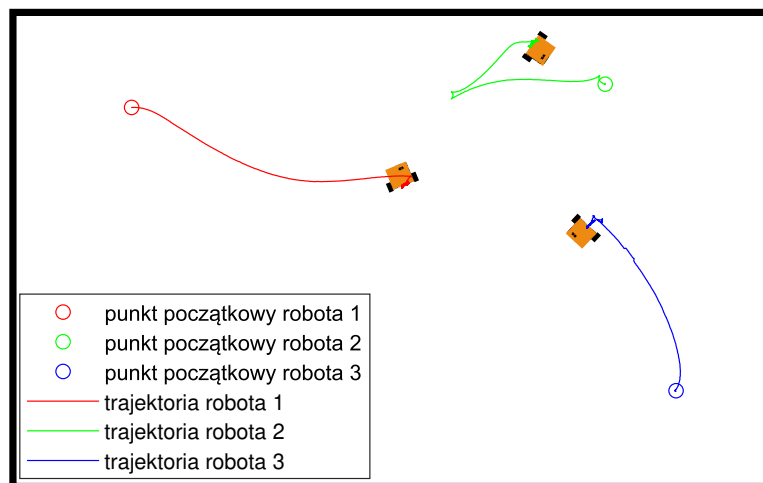
W celu weryfikacji poprawności zaproponowanej metody wykonano eksperymenty symulacyjne. Ponieważ ich wyniki były poprawne i zbliżone do przebiegów uzyskanych w zdalnym laboratorium pominięto ich przedstawienie.

W eksperymencie na rzeczywistych robotach badanie zostało podzielone na dwa etapy. W pierwszym roboty osiągną zadane położenie początkowe (4.9) w celu zagwarantowania powtarzalnych warunków.

$$p_1(0) = \begin{bmatrix} -1.1 \\ 0.6 \end{bmatrix} [\text{m}], \quad p_2(0) = \begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix} [\text{m}], \quad p_3(0) = \begin{bmatrix} 1.2 \\ -0.6 \end{bmatrix} [\text{m}] \tag{4.9}$$

W drugim etapie uruchomiony zostaje algorytm sterujący przedstawiony w przykładzie 3.1.2 i opisywany przez sterowanie (4.8) (wraz z przekształceniem opisanym w sekcji 4.2).

Na rysunku 4.10 przedstawiono trajektorie robotów na płaszczyźnie XY. Czarną obwiednią zaznaczono obszar roboczy platformy testowej. Punkty początkowe robotów zostały oznaczone kołami w kolorze czerwonym, zielonym i niebieskim odpowiednio dla robota pierwszego, drugiego i trzeciego. Symbolem robota zaznaczono końcowe położenie w eksperymencie.



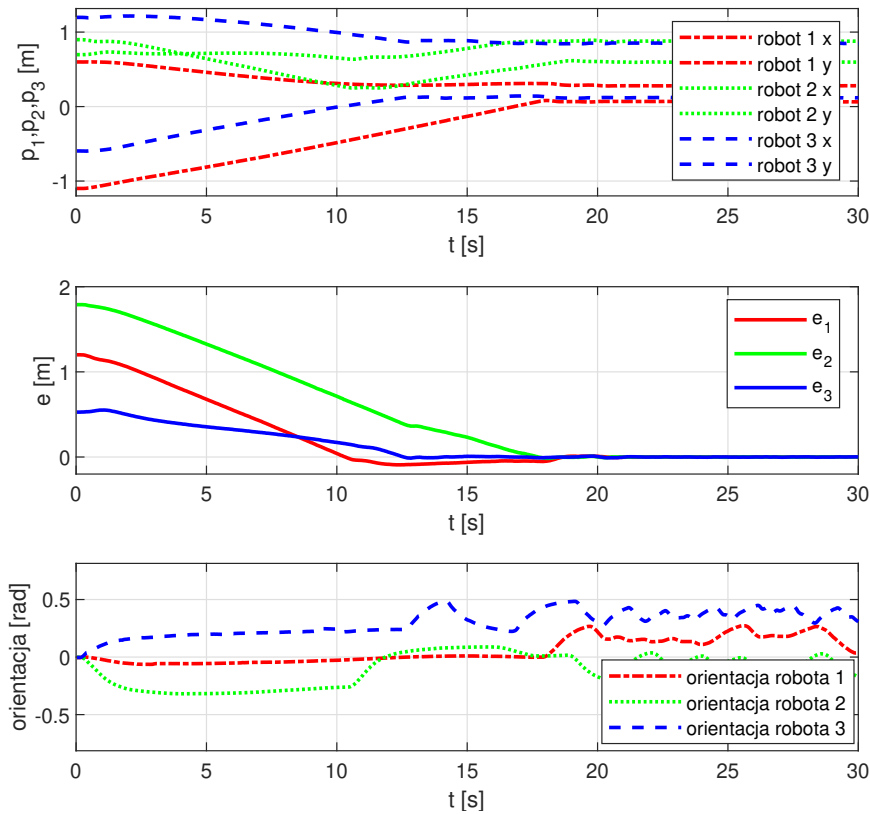
Rysunek 4.10: Ruch robotów zaznaczony na płaszczyźnie XY

Ponieważ początkowe odległości pomiędzy robotami są większe niż zadane to podczas eksperymentu pojazdy zbliżają się do siebie, a finalnie osiągną zadania formację - trójkąt równoboczny o boku 0,8 m. Jest to wynik zgodny z oczekiwaniami.

Na rysunku 4.10 widać również wpływ jednokierunkowej wymiany informacji. Na przykład robot 2 śledzący położenie robota 1 musi wykonać korektę swojego położenia w związku z tym, że robot 1 śledzący położenie robota 3 znacznie przemieszcza się w trakcie eksperymentu.

Na wykresie 4.11 przedstawiono trajektorie ruchu robotów, uchyb odległości wzajemnej robotów oraz ich orientację w trakcie trwania eksperymentu.

Wykres dowodzi, że dla opisywanego scenariusza roboty osiągną zadaną formację (uchyby wzajemnych odległości zbiegają do 0). Potwierdza to poprawność proponowanego algorytmu sterującego.

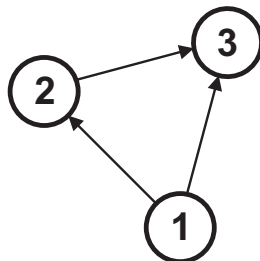


Rysunek 4.11: Przebieg położenia, uchybu wzajemnej odległości i orientacji robotów 1, 2 i 3

4.4.3 Algorytm śledzenia lidera dla trzech robotów

Celem tego eksperymentu jest weryfikacja algorytmu sterowania dla formacji robotów w przypadku nierównorzędnych ról. W eksperymencie pierwszemu z robotów zostaje przypisana rola lidera. Pozostałe dwa roboty mają za zadanie osiągnąć zadaną formację, drugi robot na podstawie pomiaru odległości do lidera, a trzeci na podstawie pomiaru odległości do dwóch sąsiadów.

Wymiana informacji w opisywanym zagadnieniu jest przedstawiona za pomocą grafu na rysunku 4.12. Podobnie jak poprzednio strzałki oznaczają przepływ informacji, np. pozycja robota pierwszego jest mierzona przez dwa roboty sąsiadujące.



Rysunek 4.12: Digraf opisujący wymianę informacji w przypadku trzech robotów, z których pierwszy jest liderem, drugi podąża za pierwszym, a trzeci za pierwszymi dwoma

Taka definicja pozwala na realizację zadań związanych z ruchem formacji pojazdów. Jeśli dla lidera zostanie zdefiniowana trajektoria ruchu to pozostałe roboty

w celu utrzymania formacji muszą podążać za pierwszym pojazdem.

W pracy [59] wykazano, że jeśli prędkość lidera zmierza do 0 to cały układ jest wykładniczo stabilny, czyli osiąga zadane odległości.

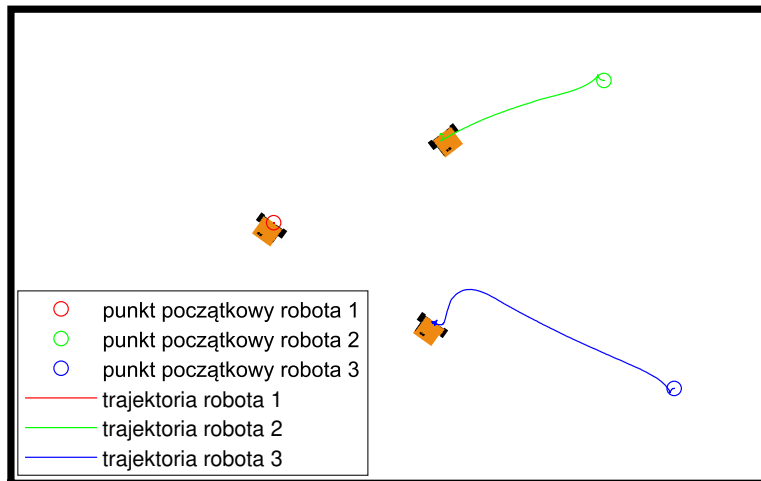
Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned} u_1 &= 0 \\ u_2 &= -(p_2 - p_1)(\|p_2 - p_1\|^2 - d^2) \\ u_3 &= -(p_3 - p_1)(\|p_3 - p_1\|^2 - d^2) - (p_3 - p_2)(\|p_3 - p_2\|^2 - d^2) \end{aligned} \quad (4.10)$$

gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, a $d = 0,8$ m jest zadaną odległością.

W pierwszym eksperymencie nie zadaje się trajektorii dla lidera czyli jego położenie pozostaje niezmiennie. Wyniki tego eksperymentu przedstawiono poniżej.

Na rysunku 4.14 przedstawiono trajektorie robotów na płaszczyźnie XY. Czarną obwiednią zaznaczono obszar roboczy platformy testowej. Punkty początkowe robotów zostały oznaczone kołami w kolorze czerwonym, zielonym i niebieskim odpowiednio dla robota pierwszego, drugiego i trzeciego. Symbolem robota zaznaczono końcowe położenie w eksperymencie.



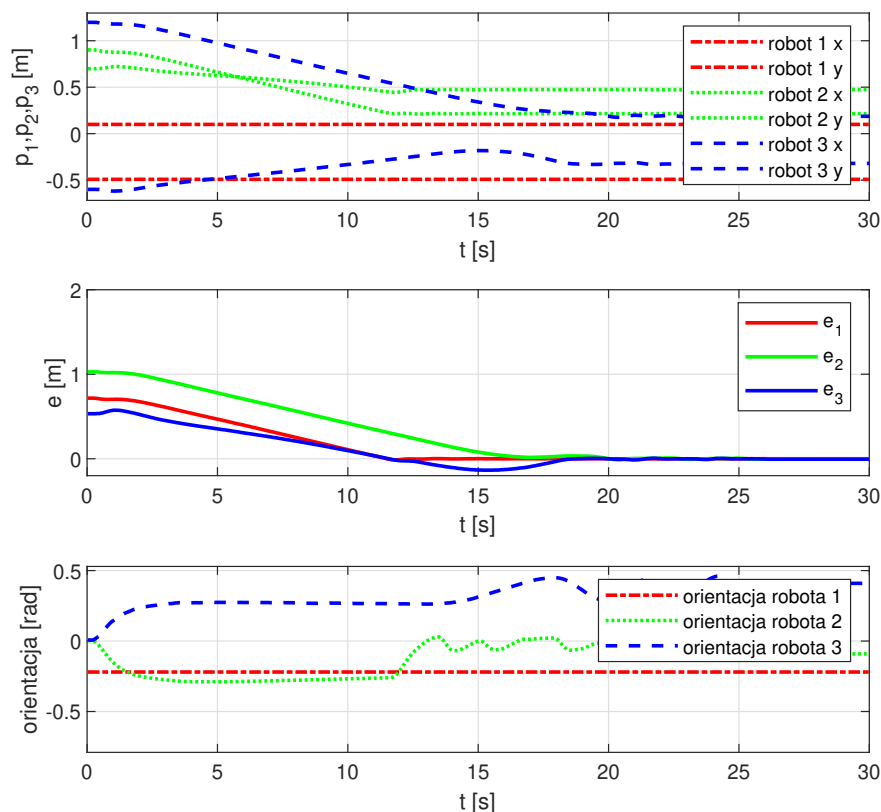
Rysunek 4.13: Ruch robotów zaznaczony na płaszczyźnie XY

Zgodnie z oczekiwaniem położenie lidera w eksperymencie nie zmienia się. Pozostałe dwa roboty osiągają zadaną formację. Ruch drugiego robota odbywa się w kierunku lidera, co zgadza się z proponowanym schematem wymiany informacji. Kierunek ruchu robota trzeciego jest związany z dwoma sąsiadami. W związku z powyższym, początkowo robot zmierza do położenia znajdującego się pomiędzy

robotem 1 i 2. W końcowej fazie eksperymentu robot koryguje swoją pozycję w celu osiągnięcia zadanej odległości od obu sąsiadów. Taki wynik jest zgodny z przewidywaniami teoretycznymi.

Na wykresie 4.14 przedstawiono trajektorie ruchu robotów, uchyb odległości wzajemnej robotów oraz ich orientację w trakcie trwania eksperymentu.

Wykres dowodzi, że dla opisywanego scenariusza roboty osiągnęły zadaną formację (uchyby wzajemnych odległości zbiegają do 0). Potwierdza to poprawność proponowanego algorytmu sterującego.



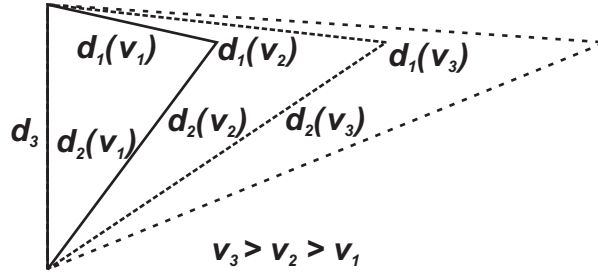
Rysunek 4.14: Przebieg położenia, uchybu wzajemnej odległości i orientacji robotów 1, 2 i 3

4.4.4 Algorytm śledzenie lidera, przypadek z niestacjonarnymi wagami

Eksperyment jest rozszerzeniem poprzedniego przypadku algorytmu śledzenia lidera. Role przypisane robotom oraz sposób wymiany informacji są takie jak w poprzednim przypadku (sekcja 4.4.3).

W eksperymencie wprowadzono dwie modyfikacje. Lider formacji porusza się wzdłuż prostej trajektorii, nadsyłając za sygnałem referencyjnym prędkości liniowej. Profil prędkości został zdefiniowany za pomocą sygnału trapezoidalnego. Oznacza to występowanie fazy rozpędzania, poruszania się ze stałą prędkością oraz fazy hamowania.

Druga modyfikacja polega na dopuszczeniu sytuacji niestacjonarnych odległości pomiędzy robotami. W eksperymencie założono, że zadane odległości pomiędzy liderem a śledzącymi robotami są zależne od prędkości robotów. Zadana odległość pomiędzy robotem 2 i 3 pozostaje stała. Motywacją takiego scenariusza jest sytuacja w ruchu drogowym. Ze względów bezpieczeństwa poruszające się pojazdy powinny zachować odległość pomiędzy poprzedzającymi samochodami zależną od ich prędkości. Sytuacja ta została zobrazowana na rysunku 4.15.



Rysunek 4.15: Zależność pomiędzy kształtem formacji a prędkością lidera

Powyższe założenie wymagają dostosowania algorytmu sterującego formacją robotów. Dla uproszczenia zakłada się, że zadany profil prędkości jest znany wszystkim robotom. To silne założenie mogłoby zostać złagodzone za pomocą obserwatora prędkości lidera [81]. W takiej sytuacji sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za ruch wzdłuż zadanej trajektorii oraz doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned}
 u_1 &= v \\
 u_2 &= -k_p(p_2 - p_1)(\|p_2 - p_1\|^2 - d_1^2) + k_v v \\
 u_3 &= -k_p((p_3 - p_1)(\|p_3 - p_1\|^2 - d_2^2(v)) + \\
 &\quad + (p_3 - p_2)(\|p_3 - p_2\|^2 - d_3^2(v))) + k_v v
 \end{aligned} \tag{4.11}$$

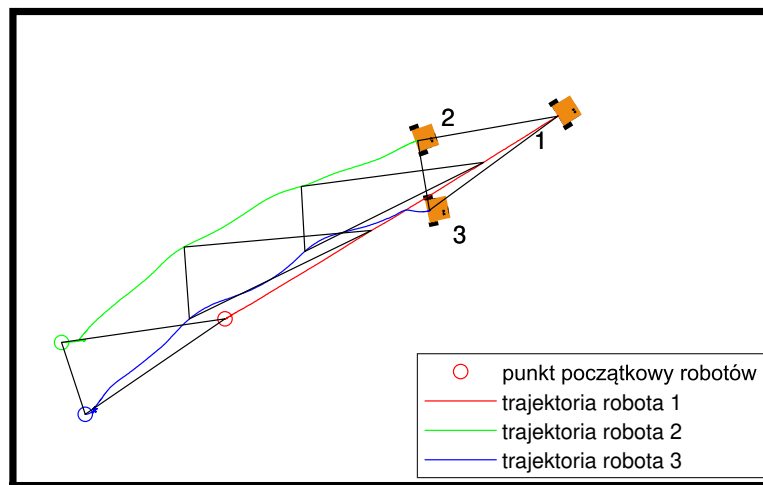
gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, v jest zadany profil prędkości, d_i są zadanymi odległościami, a k_p , k_v parametrami regulatora. Podobna postać sterowania została przedstawiona w pracy [101].

W eksperymencie na rzeczywistych robotach badanie zostało podzielone na dwa etapy. W pierwszym roboty osiągną zadane położenie początkowe (4.12) w celu zagwarantowania powtarzalnych warunków.

$$p_1(0) = \begin{bmatrix} -0,7 \\ -0,3 \end{bmatrix} [\text{m}], \quad p_2(0) = \begin{bmatrix} -1,4 \\ -0,4 \end{bmatrix} [\text{m}], \quad p_3(0) = \begin{bmatrix} -1,3 \\ -0,7 \end{bmatrix} [\text{m}] \tag{4.12}$$

Przyjęto następujące parametry regulatora $k_p = 1$, $k_v = 0,8$. Zależność pomiędzy zadaną odległością, a prędkością jest następująca: $d_1(v) = d_{10} + 2v$, $d_2(v) = d_{20} + 2v$. Zadane odległości: $d_{10} = 0,6$ m, $d_{20} = 0,67$ m, $d_3 = 0,3$ m

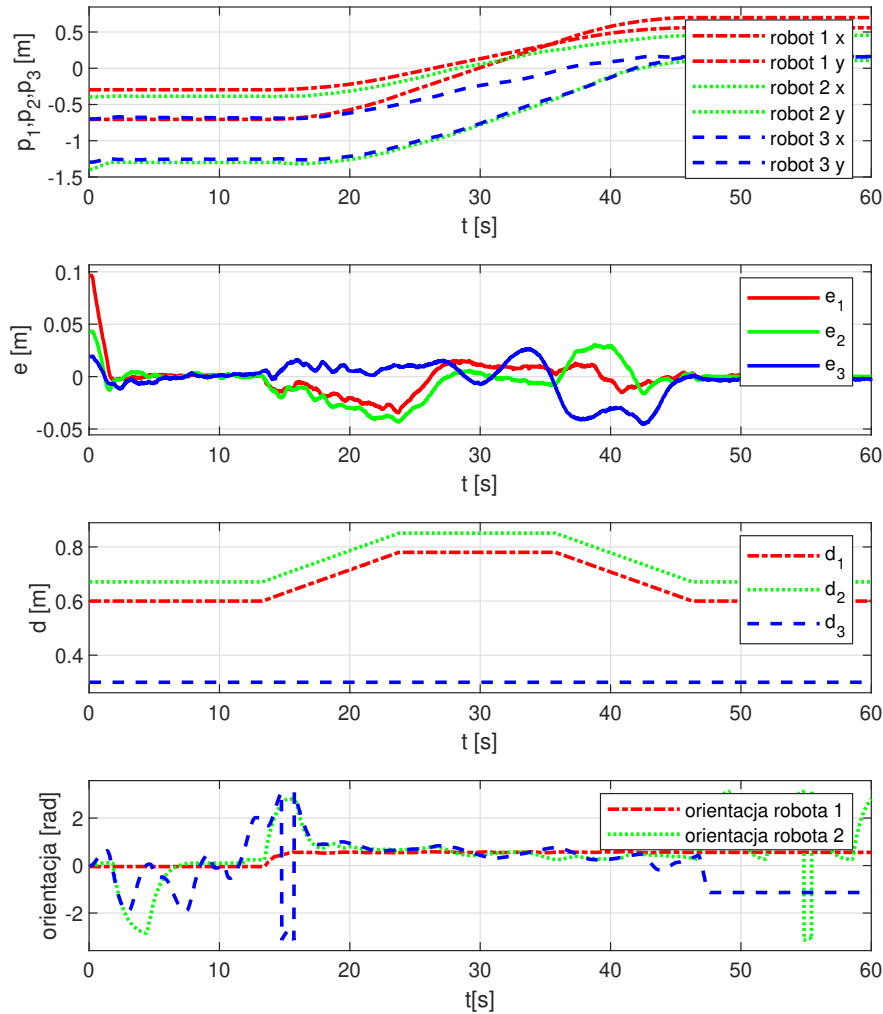
Na rysunku 4.16 przedstawiono trajektorie robotów na płaszczyźnie XY. Czarną obwiednią zaznaczono obszar roboczy platformy testowej. Punkty początkowe robotów zostały oznaczone kołami w kolorze czerwonym, zielonym i niebieskim odpowiednio dla robota pierwszego, drugiego i trzeciego. Symbolem robota zaznaczono końcowe położenie w eksperymencie. Czarnymi trójkątami oznaczono zmieniającą się w czasie formację robotów.



Rysunek 4.16: Ruch robotów zaznaczony na płaszczyźnie XY

Na wykresie 4.17 przedstawiono trajektorie ruchu robotów, uchyb odległości wzajemnej robotów, zadane odległości (zmiennie w czasie) oraz orientację pojazdów w trakcie trwania eksperymentu. Podczas ruchu lidera, roboty 2 i 3 nadążają za nim. W czasie przyspieszania i hamowania uchyby odległości osiągają niezerowe wartości. W czasie ruchu ze stałą prędkością roboty poruszają się w zadanej formacji. Po zakończeniu ruchu przez lidera, roboty 2 i 3 osiągają zadaną formację.

Wykres dowodzi, że dla opisywanego scenariusza roboty osiągają zadaną formację (uchyby wzajemnych odległości zbiegają do 0) i są w stanie utrzymać ją podczas podążania za liderem. Podczas fazy rozpędzania i hamowania pojawia się uchyb ustalony. W ruchu ze stałą prędkością oraz podczas spoczynku pojazdy osiągają zadane odległości wzajemne. Potwierdza to poprawność proponowanego algorytmu sterującego.

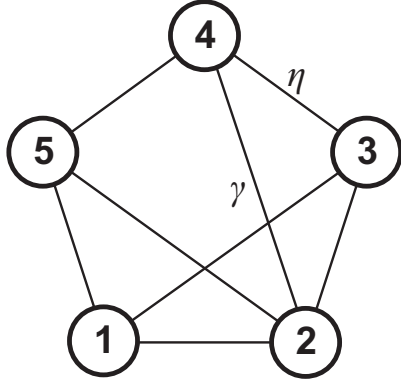


Rysunek 4.17: Przebieg położenia, uchybu wzajemnej odległości, zadanych odległości i orientacji robotów 1, 2 i 3

4.4.5 Algorytm sterowania formacją pięciu robotów z dwukierunkową wymianą informacji

Celem eksperymentu jest zbadanie zachowania większej liczby robotów oraz analiza wpływu warunków początkowych na osiągnięty efekt. Zadaniem robotów jest utworzenie formacji pięciokąta foremnego. Zakłada się, że sąsiedztwo pomiędzy robotami opisuje graf przedstawiony na rysunku 4.18. Jest to graf prosty (nieskierowany) co oznacza, że wymiana informacji jest dwukierunkowa – jeśli robot A śledzi położenie robota B, to jednocześnie B śledzi A.

Graf zawiera 8 krawędzi (nie jest grafem pełnym), aby pokazać, że nie wszystkie roboty muszą sąsiadować, aby zrealizować zadanie, a warunkiem wystarczającym jest strukturalna sztywność grafu (opisana w definicji 2.20.).



Rysunek 4.18: Graf opisujący wymianę informacji w przypadku pięciu robotów, w każdym przypadku wymiana informacji odbywa się w dwóch kierunkach (nie wszystkie roboty wymieniają informacje między sobą)

W celu analizy przypadku pięciu robotów oraz przy projektowaniu algorytmu sterującego posłużono się pojęciami z teorii grafów. Do opisu zadanych odległości użyto macierzy wag $D(\mathcal{G})$. Jej elementy d_{ij} oznaczają zadane odległości pomiędzy pojazdami i, j . Dla opisywanego przypadku macierz sąsiedztwa $A(\mathcal{G})$ i macierz wag mają postać:

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad D(\mathcal{G}) = \begin{bmatrix} 0 & \eta & \gamma & 0 & \eta \\ \eta & 0 & \eta & \gamma & \gamma \\ \gamma & \eta & 0 & \eta & 0 \\ 0 & \gamma & \eta & 0 & \eta \\ \eta & \gamma & 0 & \eta & 0 \end{bmatrix}, \quad (4.13)$$

gdzie η oznacza bok pięciokąta, a γ jego przekątną i wynosi $\gamma = \frac{1+\sqrt{5}}{2}\eta$.

Sterowanie dla poszczególnych robotów, które pozwala osiągnąć zadaną formację ma postać

$$u_i = - \sum_{j=1, \dots, 5} a_{ij} (\|p_i - p_j\|^2 - d_{ij}^2) (p_i - p_j), \quad (4.14)$$

gdzie a_{ij} są elementami macierzy sąsiedztwa, a d_{ij} oznaczają zadane odległości pomiędzy parami pojazdów i, j (są elementami macierzy wag $D(\mathcal{G})$).

Przeprowadzono eksperymenty z dwoma zestawami warunków początkowych aby zbadać ich wpływ na osiągniętą formację. Dla sąsiedztwa opisanego w tym przykładzie są dwa poprawne rozwiązania docelowe. Jednym z nich jest graf przedstawiony na rysunku 4.18, gdzie kolejność oznaczenia wierzchołków jest przeciwna do ruchu wskazówek zegara. Drugim rozwiązaniem jest odwrotna kolejność oznaczenia wierzchołków. Oba rozwiązania są więc względem siebie symetryczne, a opisujące je

macierze sąsiedztwa i stopni takie same.

Oczekuje się, że w zależności od warunków początkowych rozwiązanie zbiegnie do pierwszego lub drugiego przypadku.

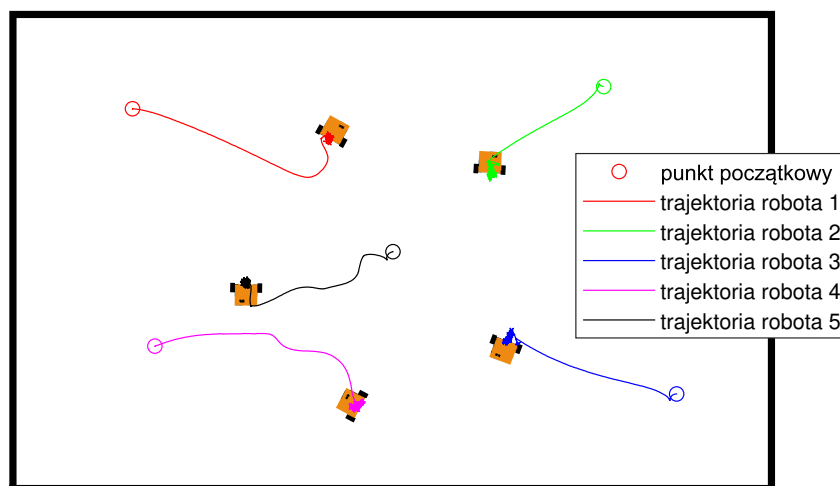
Pierwszy zestaw warunków początkowych jest następujący:

$$p_N(0) = \begin{bmatrix} -1,1 & 0,9 & 1,2 & -1,0 & 0,0 \\ 0,6 & 0,7 & -0,6 & 0,5 & 0,0 \end{bmatrix} [\text{m}], \quad (4.15)$$

gdzie $p_{P/N}(0) = [p_1(0) | p_2(0) | p_3(0) | p_4(0) | p_5(0)]$. Drugi zestaw warunków początkowych:

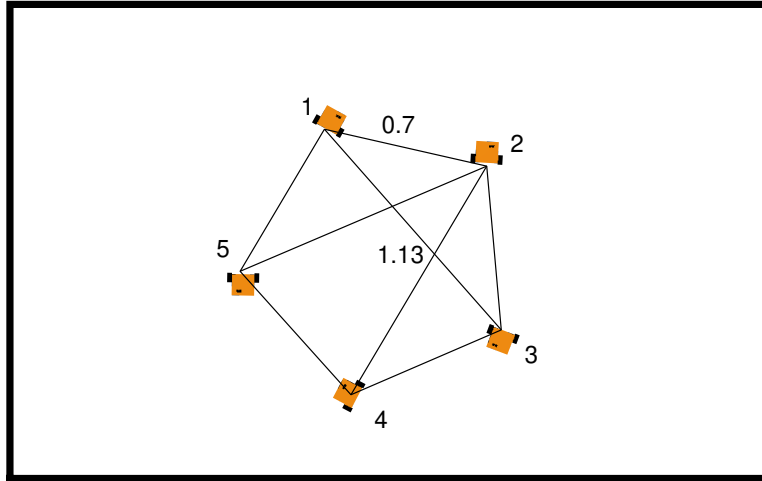
$$p_P(0) = \begin{bmatrix} -0,8 & 1,1 & 1,2 & -1,0 & 0,0 \\ -0,4 & -0,5 & 0,6 & 0,5 & 0,0 \end{bmatrix} [\text{m}], \quad (4.16)$$

Na rysunku 4.22 przedstawiono trajektorie robotów na płaszczyźnie XY, gdy punkty początkowe są równe (4.15), a zadany bok pięciokąta $\eta = 0,7\text{m}$. Czarną obwiednią zaznaczono obszar roboczy platformy testowej. Punkty początkowe robotów zostały oznaczone kołami w kolorze czerwonym, zielonym, niebieskim, różowym i czarnym odpowiednio dla robota pierwszego, drugiego, trzeciego, czwartego i piątego. Symbolem robota zaznaczono końcowe położenie w eksperymencie.



Rysunek 4.19: Ruch robotów zaznaczony na płaszczyźnie XY

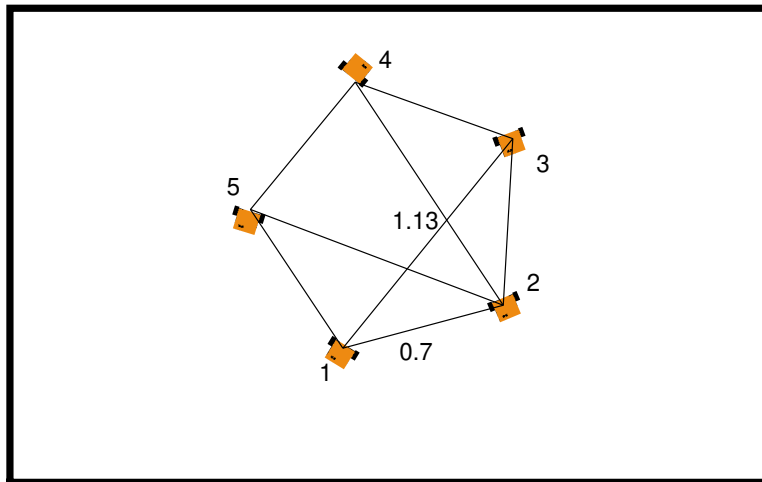
Na rysunku 4.20 zaznaczono osiągniętą formację (dla warunków początkowych (4.15)). Numery przy robotach odpowiadają wierzchołkom grafu. Krawędzie oznaczają sąsiedztwo robotów, a tym samym wymianę informacji przez roboty. Przy krawędziach zaznaczono założone odległości pomiędzy robotami.



Rysunek 4.20: Osiągana formacja w przypadku warunków początkowych (4.15)

Można zauważyć, że osiągnięta formacja jest lustrzanym odbiciem założonej trajektorii zdefiniowanej przez graf 4.18.

Na rysunku 4.21 zaznaczono osiągniętą formację (dla warunków początkowych (4.16)). Kolejność wierzchołków jest zgodna z grafem 4.18.



Rysunek 4.21: Osiągana formacja w przypadku warunków początkowych (4.16)

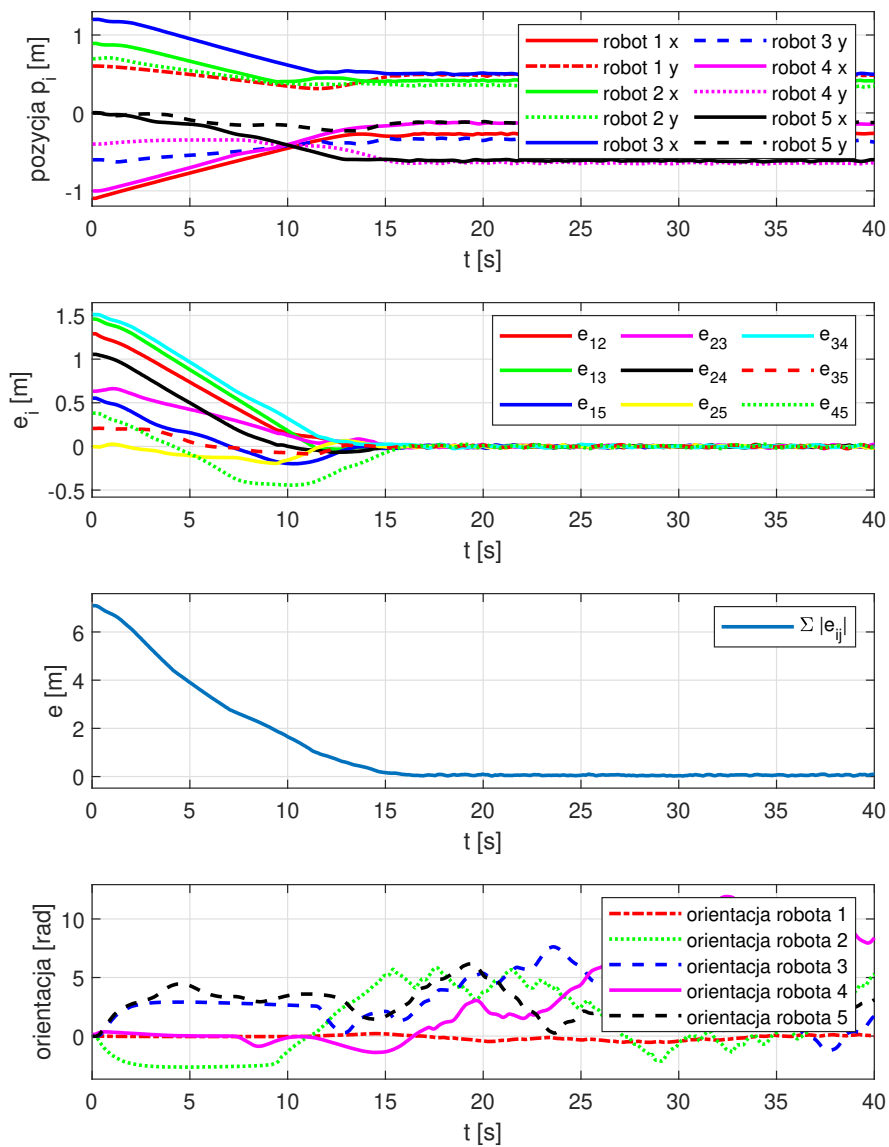
Na wykresie 4.22 przedstawiono trajektorie ruchu robotów, uchyb odległości wzajemnej robotów, uchyb całkowity oraz orientację robotów w trakcie trwania eksperymentu.

Wykres dowodzi, że dla opisywanego scenariusza roboty osiągną zadaną formację (uchyby wzajemnych odległości oraz uchyb całkowity zbiegają do 0). Potwierdza to poprawność proponowanego algorytmu sterującego.

Istotnym wnioskiem płynącym z przeprowadzonych badań jest fakt występowania dwóch zbiorów punktów równowagi. Formacja osiąga jeden ze zbiorów w zależ-

ności od warunków początkowych. Przeprowadzona analiza matematyczna (w przykładzie 3.1.2) potwierdza, że dla każdej niewspółliniowej konfiguracji początkowej roboty osiągną punkt równowagi będący zadaną formacją.

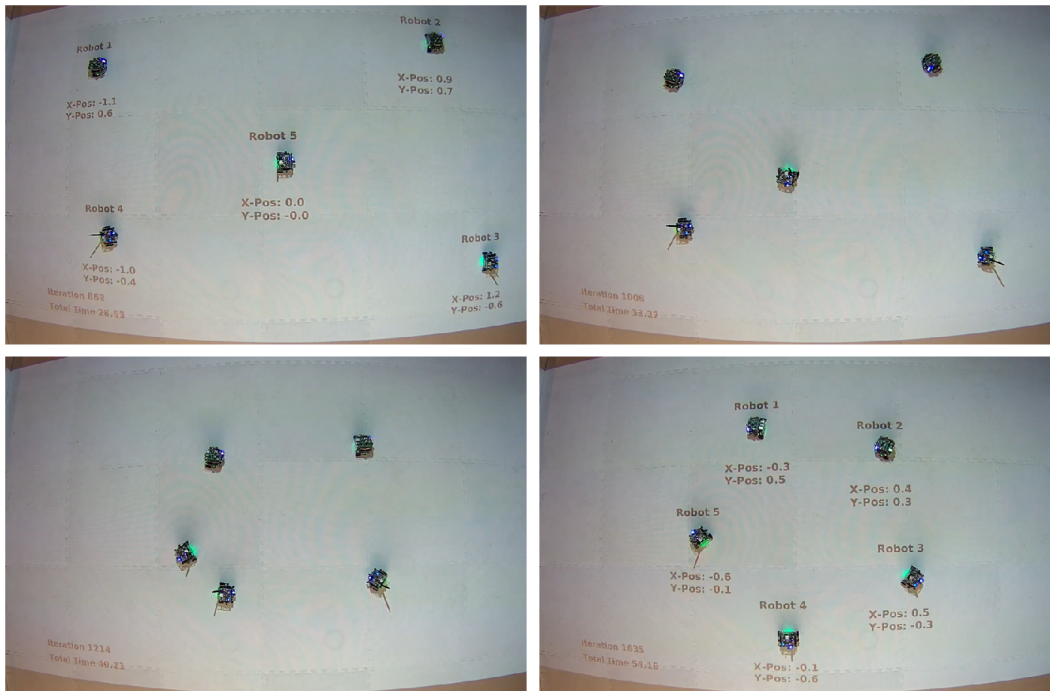
Warunkiem koniecznym asymptotycznej stabilności jest strukturalna sztywność grafu [3]. Można ją zagwarantować przez definicję wystarczającej liczby krawędzi w grafie, czyli określenie sąsiadujących par robotów.



Rysunek 4.22: Przebieg położenia, uchybu wzajemnej odległości, sumy wartości bezwzględnej uchybów i orientacji robotów 1-5

Na rysunku 4.23 przedstawiono migawki z przeprowadzonego eksperymentu. Na pierwszym kadrze widać położenia robotów w początkowej chwili eksperymentu. Kolejne dwa kadry przedstawiają pozycję robotów w czwartej i jedenastej sekundzie eksperymentu w trakcie zmiany położenia. Ostatni kadr przedstawia roboty po osiągnięciu zadanej formacji. Na początku eksperymentu oraz po osiągnięciu zadanej

odległości projektor umieszczony ponad platformą wyświetla informację o numerze robotów oraz ich początkowym i końcowym położeniu.



Rysunek 4.23: Migawki z nagrania rzeczywistego eksperymentu dla początku eksperymentu, czwartej i jedenastej sekundy oraz po osiągnięciu pozycji zadanej

4.5 Podsumowanie wyników

Przeprowadzone badania symulacyjne i eksperymentalne na robotach klasy (2,0) potwierdziły możliwość zastosowania algorytmów sterowania formacją robotów mobilnych opracowanych dla układów o uproszczonej dynamice.

Implementacja algorytmów potwierdziła poprawność matematycznej analizy przeprowadzonej w rozdziale 3. Proponowane algorytmy zapewniają zbieżność położenia robotów do zadanej formacji. Występowanie rzeczywistych zjawisk, takich jak szумы pomiarowe, ma wpływ na osiągnięte rezultaty, ale nie uniemożliwia zastosowania opisanych metod.

W tabeli 6.1 zamieszczono odnośniki do nagrań z przeprowadzonych eksperymentów.

5 Algorytmy sterowania zespołem robotów niestabilnych

W rozdziale zawarto opis oraz wyniki algorytmów sterowania zespołem robotów niestabilnych. Ponadto, w pierwszej części, opisano platformę sprzętowo – programową, wykonaną i użytą w celu implementacji i weryfikacji algorytmów sterowania rozproszonego dla zespołu robotów mobilnych. Platforma składa się z jednakowych n robotów mobilnych oraz oprogramowania umożliwiającego swobodne programowanie i zarządzanie robotem oraz eksperymenty symulacyjne.

Zamysłem stojącym za przedstawioną platformą było opracowanie stanowiska badawczo – edukacyjnego i udostępnienie go zainteresowanym osobom. Głównym założeniem jest uproszczenie programowania robotów, co zostało osiągnięte przez wykorzystanie graficznego języka opisu za pomocą diagramów blokowych. Implementacja algorytmów sterujących odbywa się dzięki automatycznej generacji kodu, a za uruchomienie, w reżimie czasu rzeczywistego, odpowiada karta pomiarowo – sterująca wyposażona w procesor oraz odpowiednie oprogramowanie i pozwalająca użytkownikowi końcowemu osiągnięcie precyzyjnych i powtarzalnych rezultatów. Takie podejście pozwala skupić się na aspektach związanych z algorytmami sterującymi, pomijając złożone kwestie realizacji sprzętowo – programowej [49].

Na rynku dostępne są platformy oferujące zbliżoną funkcjonalność. Najważniejsze rozwiązania są skupione wokół projektu ROS/ROS 2 (Robot Operating System). ROS to przede wszystkim zestaw bibliotek programistycznych przeznaczonych do rozwoju systemów robotycznych. Roboty mobilne korzystające z tego rozwiązania to m.in. Turtlebot [111] oraz ROSbot [109]. Wśród innych rozwiązań można wymienić platformę QBot opracowaną przez firmę Quanser [108]. Istotną cechą różniącą dostępne rozwiązania od opisywanego w pracy jest użycie pojazdu mobilnego o niestabilnej dynamice.

Głównym komponentem platformy jest mobilny robot posiadający dwa napędzane koła umieszczone w jednej osi bez dodatkowych punktów podparcia. Ze względu na środek ciężkości pojazdu, znajdujący się ponad osią obrotu, robot jest niestabilny w pozycji pionowej i do jej utrzymania wymagane jest aktywne sterowanie. Podobna konstrukcja została opisana w [36, 33]. Dynamika pojazdu przypomina układ waha dła odwróconego będący popularnym i powszechnie uznanym [6, 86] stanowiskiem testowym algorytmów sterowania.

Do realizacji tego zadania robot został wyposażony w zestaw czujników mierzących położenie kół i pochylenie robota oraz w sterownik oparty na układzie rekonfigurowalnym oraz mikroprocesor zapewniający przetwarzanie sygnałów i ob-

liczanie sterowania w czasie rzeczywistym. Programowanie robota może odbywać się przy pomocy zewnętrznego komputera wyposażonego w oprogramowanie MATLAB/Simulink korzystając z metodologii projektowania układów sterowania w oparciu na modelu matematycznym (ang. *Model Based Design*, MBD).

Z punktu widzenia sterowania położeniem robota jest on podobny do nieholonomicznych pojazdów przedstawionych w rozdziale 4 z tą różnicą, że robot nie posiada samonastawnego koła podpierającego. W związku z powyższym w zadaniach sterowania podstawowym priorytetem jest utrzymanie pionowej postawy, a kontrola położenia robota i związane z tym zadania śledzenia trajektorii są zadaniem o niższym priorytecie.

W kolejnych podrozdziałach zostanie przedstawiona budowa sprzętowa robota oraz rozwiązania programowe umożliwiające prace z robotem. Ponadto zostanie opisany model matematyczny robota wraz z identyfikacją jego parametrów fizycznych oraz weryfikacją poprawności proponowanego modelu. Finalnie zostaną przedstawione wyniki implementacji algorytmów sterowania rozproszonego dla struktury robotów w oparciu o prezentowany model matematyczny oraz eksperymenty na rzeczywistym robocie.



Rysunek 5.1: Autonomiczne pojazdy dwukołowe; (a) Robot Handle opracowany przez firmę Boston Dynamics [źródło: bostondynamics.com]; (b) Pojazd PUMA (ang. Personal Urban Mobility and Accessibility) opracowany przez firmy GM i Segway [źródło: newatlas.com/segway-car-project-puma/11413/]

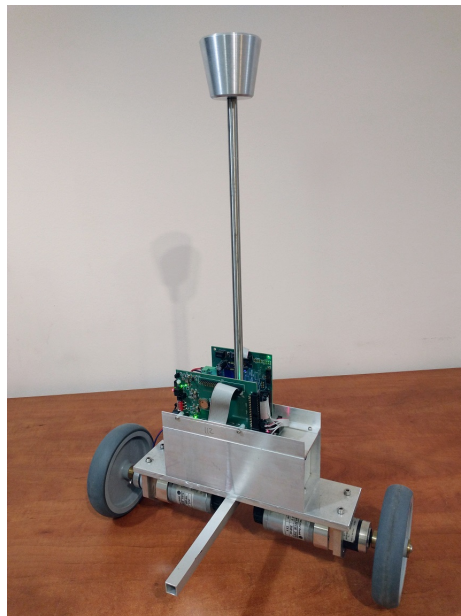
Motywacją dla rozważania algorytmów sterowania formacją niestabilnych robotów mobilnych są potencjalne zyski ze stosowania takich rozwiązań. Niestabilne pojazdy oferują większą zwrotność i manewrowość niż ich stabilne odpowiedniki (np. samolot F16 [92]). Przykładowym rozwiązaniem przemysłowym wykorzystującym

taką dynamikę jest robot Handle stworzony przez firmę Boston Dynamics 5.1a. Jest to robot logistyczny przeznaczony do zadań związanych z paletyzacją. Został on wyposażony w robotyczne ramię z chwytakiem próżniowym. Dzięki dużej zwrotności robot może operować na małych przestrzeniach. Innym rozwiązaniem jest koncepcyjny pojazd PUMA (ang. *Personal Urban Mobility and Accessibility*), przedstawiony na obrazku 5.1b, opracowany przez firmy Segway i GM jako autonomiczna miejska taksówka.

Opisywana platforma została zrealizowana w ramach projektu NCBIR INNO-TECH-K1/HI1/12/158460/NCBR/12. Do zadań autora tej pracy należało opracowanie i wykonanie prototypu pojazdu, opracowanie modelu matematycznego i identyfikacja jego parametrów oraz opracowanie algorytmów sterujących.

5.1 Dwukołowy, niestabilny robot mobilny

Dwukołowy niestabilny robot mobilny, przedstawiony na rysunku 5.2, jest pojazdem zaprojektowanym do realizacji eksperymentów związanych ze sterowaniem. Zastosowane rozwiązania sprzętowo-programowe m.in. zasilanie bateryjne i wbudowany system sterujący pozwalają na autonomiczną pracę. Programowanie i nadzór nad robotem odbywa się przy pomocy komputera PC. Dzięki integracji z pakietem MATLAB/Simulink wszystkie etapy opracowania algorytmu sterującego mogą zostać przeprowadzone w jednym środowisku. Szczegółowy opis platformy sprzętowej został przedstawiony w podrozdziale 5.1.1, a zastosowane rozwiązania programowe w podrozdziale 5.1.2.



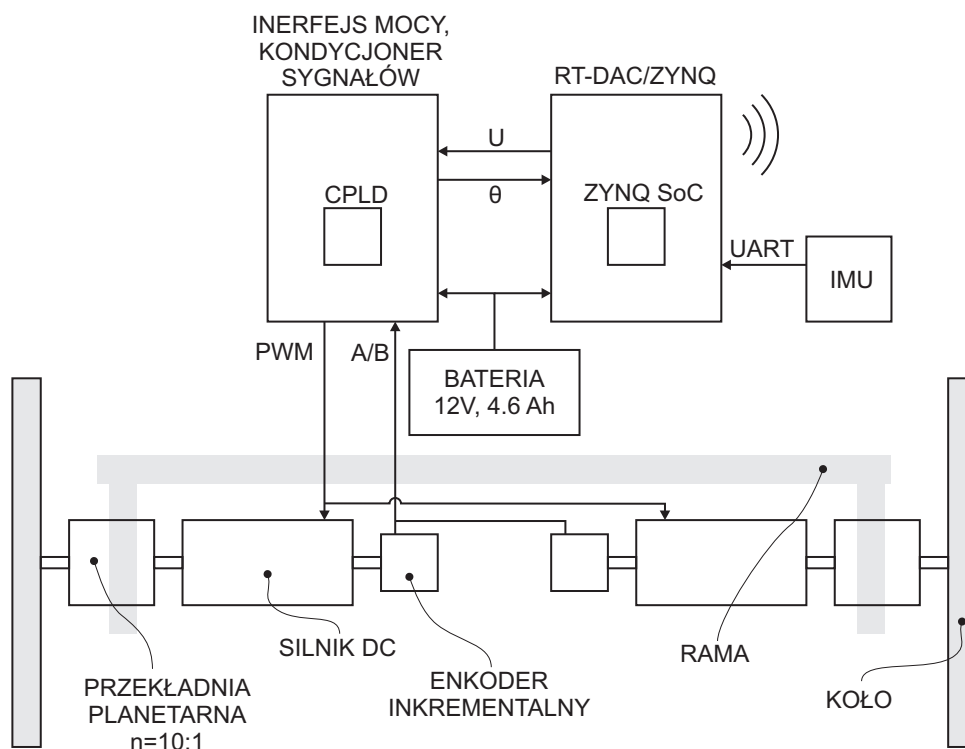
Rysunek 5.2: Fotografia dwukołowego, niestabilnego robota mobilnego

5.1.1 Platforma sprzętowa

Opisywany robot mobilny został zbudowany na aluminiowej ramie, do której przymocowano napęd, obudowę mieszczącą baterie oraz szyny do mocowania układów elektronicznych. Do ramy robota został zamontowany pręt zakończony przeciwwagą w celu przemieszczenia środka ciężkości robota ponad oś obrotu. Do napędu robota wykorzystano dwa silniki DC (Buehler Motor 1.13.044.001), każdy połączony z przekładnią planetarną zapewniającą redukcję prędkości obrotowej ($n = \frac{1}{10}$) (Neugart PLE040-010). Użyto koła o średnicy 150 mm, których gładki bieżnik został pokryty termoplastyczną gumą. Do pomiaru położenia kąтового wału użyto enkoderów inkrementalnych o rozdzielczości 400 imp/obrót (HEDS-5600-H06) umieszczonych bezpośrednio na wale silnika. Sterowanie kierunkiem oraz prędkością obrotową odbywa się poprzez mostki H umożliwiające generację bipolarnego sygnału o zmiennym wypełnieniu - PWM (ang. *Pulse Width Modulation*). Jako magazyn energii zastosowano akumulator żelowy o napięciu 12 V i pojemności 4,6 Ah pozwalający na kilkugodzinną nieprzerwaną pracę urządzenia. Poszczególne komponenty robota przedstawiono na rysunku 5.3.

Zastosowane silniki DC są dobrym wyborem ze względu na łatwość sterowania prędkością obrotową. Nominalny moment obrotowy, prędkość obrotową oraz przełożenie dobrano w taki sposób, aby umożliwić osiągnięcie prędkości liniowej robota około 1 m/s.

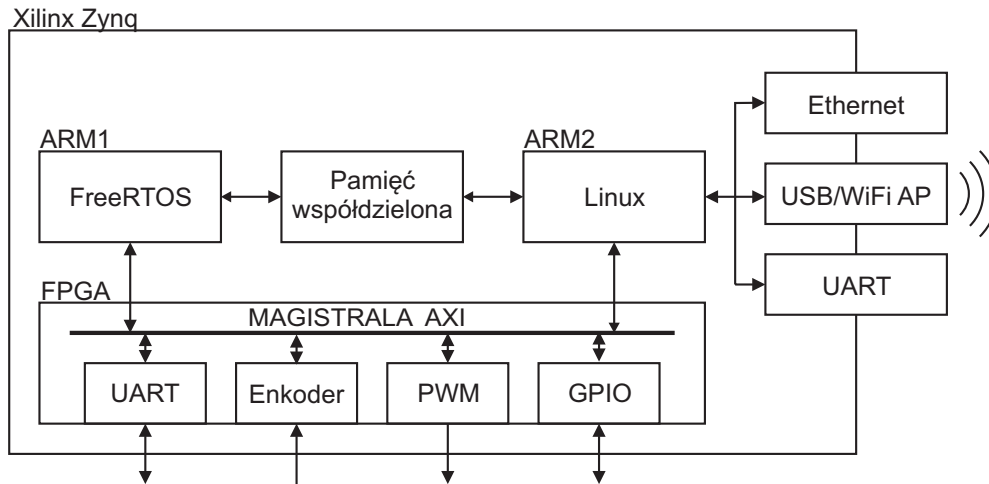
Wybór przekładni planetarnej jest podyktowany jej niskim luzem kątowym. Zjawisko luzu negatywnie wpływa na stabilizację robota w niestabilnym punkcie równowagi. Wynika to z charakterystyki pracy regulatora, który podczas stabilizacji często zmienia znak sterowania. W momencie zmiany kierunku obrotów silnik przez pewien czas pracuje bez obciążenia, co powoduje szybkie rozpędzenie rotora, a następnie uderzenie w momencie styku zębów przekładni. Zastosowana przekładnia minimalizuje to zjawisko.



Rysunek 5.3: Dwukołowy robot mobilny - schemat

Do pomiaru pochylenia robota wykorzystano czujnik inercyjny (ang. *Inertial Measurement Unit*, IMU) wyposażony w 3 osiowy akcelerometr i 3 osiowy żyroskop (ASM330LLH). Jest to mikroukład elektromechaniczny (ang. *microelectromechanical system*, MEMS) mierzący wektor przyspieszenia oraz prędkości obrotowej. W celu odtworzenia kąta pochylenia robota zastosowano obserwator stanu w formie filtra Kalmana, wykorzystujący dane z dwóch osi akcelerometru i jednej osi żyroskopu. W podrozdziale 5.1.6 podano szczegółowy opis algorytmu.

Platforma wykorzystuje kartę pomiarowo – sterującą RT-DAQ/Zynq. Została ona opracowana przez firmę Inteco na bazie modułu MicroZed wyprodukowanego przez firmę Avnet. Jednostkę wykonawczą stanowi układ scalony Xilinx Zynq, który integruje wielordzeniowy procesor ARM Cortex-A9 oraz układ FPGA wykonany w technologii 28 nm. Dwurdzeniowy mikroprocesor wykorzystuje architekturę ARM Cortex-A9. Jest ona przeznaczona dla złożonych systemów wymagających dużej mocy obliczeniowej. Procesory tego typu można znaleźć w popularnych systemach wbudowanych, takich jak Raspberry Pi czy BeagleBone. Układy wyposażone w procesor z rdzeniem Cortex-A posiadają szereg zalet, które powodują, że są one dobrze dopasowane do zastosowań związanych z przetwarzaniem sygnałów i automatycznym sterowaniem [34].



Rysunek 5.4: Architektura karty RT-DAQ/Zynq (na podstawie [29])

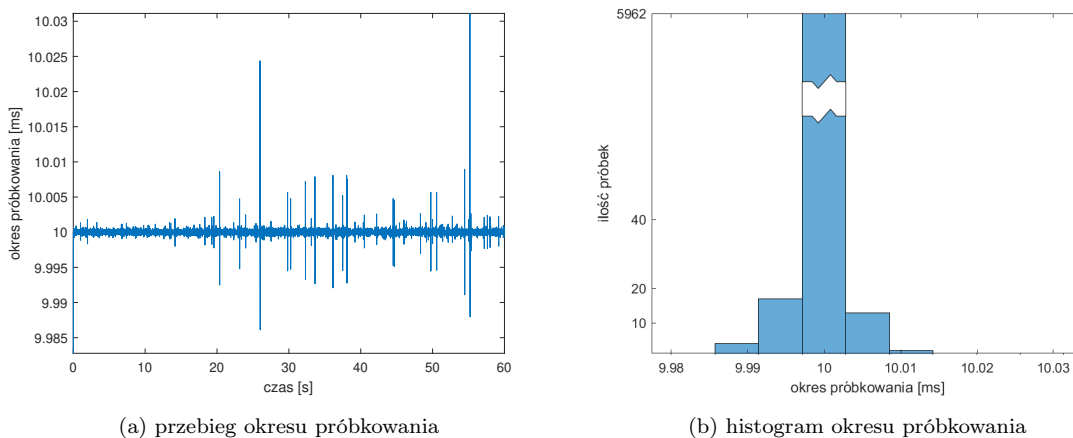
Karta RT-DAQ/Zynq wykorzystuje układ FPGA do realizacji zadań sprzętowych potrzebnych do sterowania urządzeń mechatronicznych, takich jak generacja sygnału PWM oraz dekodowanie sygnału enkodera kwadraturowego. Może ona również zostać wyposażona w interfejsy komunikacyjne, takie jak SPI, I2C czy UART do wymiany informacji z zewnętrznymi urządzeniami. W przypadku niestabilnego robota interfejs UART został użyty do wymiany danych z czujnikiem bezwładnościowym. Układ FPGA umożliwia także implementację algorytmów w celu ich akceleracji sprzętowej, a także wywoływania zadań z okresami próbkowania do 100 MHz.

Dwa rdzenie układu Xilinx Zynq są skonfigurowane w trybie asymetrycznej wielowątkowości, co oznacza, że każdy z nich ma przydzielone określone, specjalizowane zadanie. Architektura karty przedstawia rysunek 5.4. Pierwszy z procesorów pracuje pod kontrolą systemu czasu rzeczywistego FreeRTOS. Jest to jeden z najprostszych systemów czasu rzeczywistego dla systemów wbudowanych oferujący niezbędne mechanizmy, takie jak scheduler umożliwiający wywłaszczanie zadań oraz metody do tworzenia m.in. wątków, semaforów i timerów. W podstawowej wersji systemu FreeRTOS brakuje zaawansowanych funkcji znanych z innych systemów operacyjnych, takich jak obsługa sieci czy zaawansowane zarządzanie pamięcią. Dzięki maksymalnemu uproszczeniu funkcjonalności uzyskuje się bardzo dobre właściwości czasu rzeczywistego. Z tego powodu głównym zadaniem pierwszego procesora jest uruchomienie zadań wymagających utrzymania reżimu czasu rzeczywistego, do których należy przede wszystkim odczytanie wejść, odtworzenie stanu obiektu, obliczenie sterowania i ustawienie wyjść. Drugi procesor pracuje pod kontrolą systemu operacyjnego ogólnego przeznaczenia - Linux. Dzięki jego dużej elastyczności odpowiada on za komunikację z systemem nadrzędnym i umożliwia zarządzaniem parametrami pracy pierwszego procesora. Oba procesory komunikują się między sobą za pomocą mechanizmu pamięci współdzielonej. Taki rozdział zadań pozwala uży-

skać bardzo wysoką wydajność sterowania przy jednoczesnym zapewnieniu łatwości obsługi systemu. Szczegółowa analiza opisywanego układu w kontekście różnych zastosowań została przedstawiona w [29, 30, 51].

Jednym z parametrów charakteryzujących jakość systemu czasu rzeczywistego jest odchyłka pomiędzy założonym czasem próbkowania, a rzeczywistym określaną angielskim terminem *jitter*. Na wykresie 5.5a przedstawiono okres próbkowania układu. Pomiar odbywa się z uruchomionym algorytmem sterowania formacją oraz komunikacją z zewnętrznym komputerem. Założony okres próbkowania to 10 ms.

Na wykresie 5.5b przedstawiono histogram okresu próbkowania dla 60 sekundo-
wego eksperymentu. Ponad 99 % próbek ma odchyłkę od zadanego okresu (*jitter*) mniejszą niż 3 μ s. Maksymalna odchyłka od zadanej (najgorszy przypadek) wynosi 30 μ s.

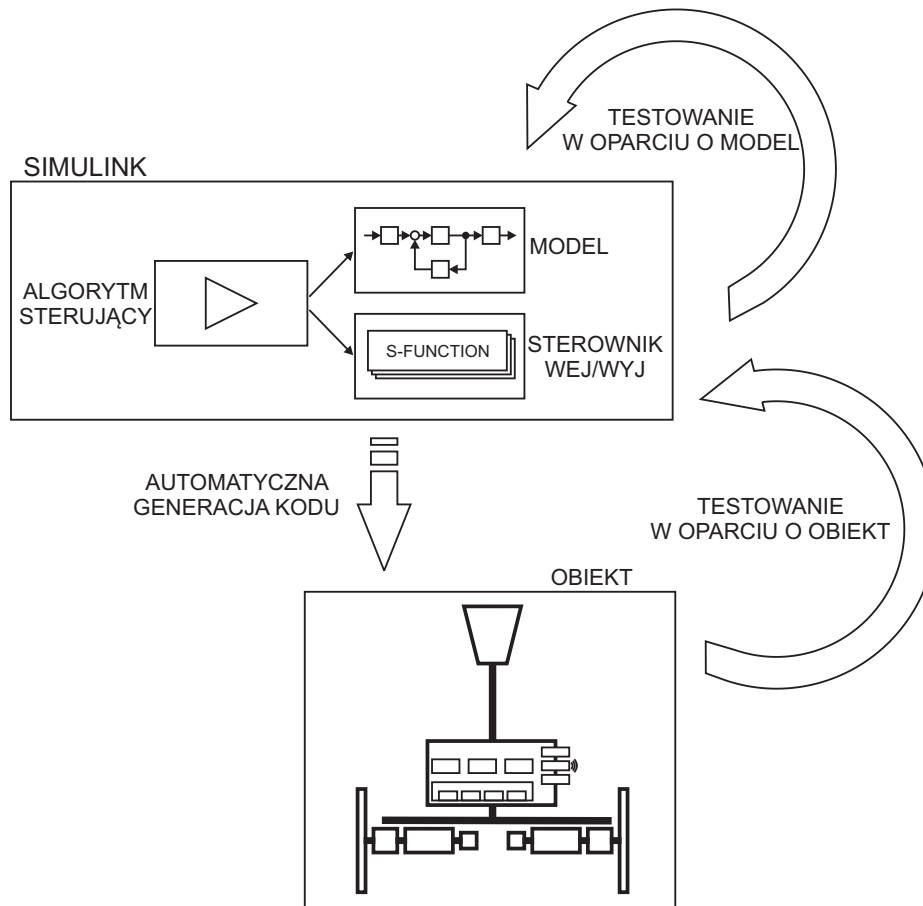


Rysunek 5.5: Wydajność układu sterowania

5.1.2 Rozwiązania programowe

Opisywany niestabilny robot mobilny jest zintegrowany z oprogramowaniem MATLAB/Simulink. Opracowanie algorytmu sterującego odbywa się za pomocą diagramu blokowego będącego graficznym językiem programowania. To samo środowisko pozwala na testy i weryfikację rozwiązania przy wykorzystaniu modeli matematycznych oraz zebranych danych z rzeczywistych eksperymentów. Pozwala to na wykorzystanie metodologii MBD przy projektowaniu algorytmów sterujących zarówno dla pojedynczego robota, jak i dla zespołu robotów mobilnych.

Istotnym komponentem metodologii MBD jest automatyczna generacja kodu. Po opracowaniu i przetestowaniu algorytmu narzędzia, takie jak MATLAB Coder i Simulink Coder, tłumaczą diagram blokowy na język C/C++. Dostarczona konfiguracja karty RT-DAQ/Zynq pozwala połączyć i skompilować kod z zewnętrznymi



Rysunek 5.6: Metodologia projektowania algorytmów na podstawie modelowania matematycznego oraz procedur automatycznej generacji kodu

bibliotekami, co prowadzi do uzyskania aplikacji wykonywalnej. Aplikacja ta może zostać uruchomiona na docelowej platformie. W trakcie prowadzonych eksperymentów wygenerowana aplikacja komunikuje się z oprogramowaniem MATLAB/Simulink pozwalając na edycję parametrów algorytmu oraz akwizycję danych eksperymentalnych. Schemat metodologii został przedstawiony na rysunku 5.6.

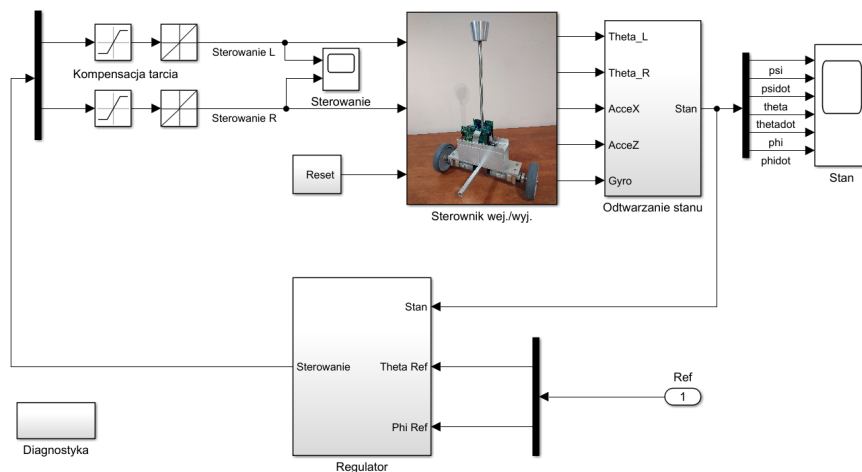
Dzięki wykorzystaniu metodologii MBD skraca się czas potrzebny na uruchomienie aplikacji. Ponadto wbudowane mechanizmy weryfikacji umożliwiają testowanie opracowanych rozwiązań na docelowej platformie. Powyższe cechy umożliwiają skupienie uwagi na poprawnym opracowaniu algorytmu przy jednoczesnym pominięciu złożonej implementacji na docelową platformę obliczeniową.

Przykładowy schemat blokowy realizujący zadanie sterowania pojedynczym robotem został przedstawiony na rysunku 5.7. Schemat taki jest hierarchiczną strukturą złożoną z bloków oraz połączeń pomiędzy nimi. Połączenia symbolizują wymianę informacji natomiast poszczególne bloki grupują komponenty algorytmu sterującego. Należą do nich:

- Sterownik wejściowo – wyjściowy — jest to fragment oprogramowania, który

definiuje sposób wymiany informacji pomiędzy oprogramowaniem a sprzętem. Blok ten został zdefiniowany przy pomocy języka C za pomocą specjalnej struktury wymaganej przez oprogramowanie Simulink – *S-funkcji*. Określa ona charakter danych wejściowych i wyjściowych oraz definiuje metody wykorzystywane na poszczególnych etapach działania symulacji oraz wynikowej aplikacji.

- Obserwator stanu — blok ten odpowiada za przetworzenie danych pomiarowych i obliczenie na ich podstawie stanu robota. Do odtworzenia pochylenia robota wykorzystano filtr Kalmana opisany w podrozdziale 5.1.6
- Regulator — blok ten odpowiada za obliczenie sterowania utrzymującego pionową pozycję robota oraz umożliwia zadanie przemieszczenia liniowego oraz obrotu robota. Regulator został opisany w podrozdziale 5.1.7
- Wizualizacja (scope) — blok ten odpowiada za prezentację sygnałów sterujących i stanu robota. Ponadto pozwala on gromadzić i zapisywać dane z przebiegu eksperymentu do późniejszej analizy.
- Diagnostyka — blok ten zawiera informacje o okresach próbkowania oraz dostarcza danych diagnostycznych na temat karty pomiarowo – sterującej i robota.



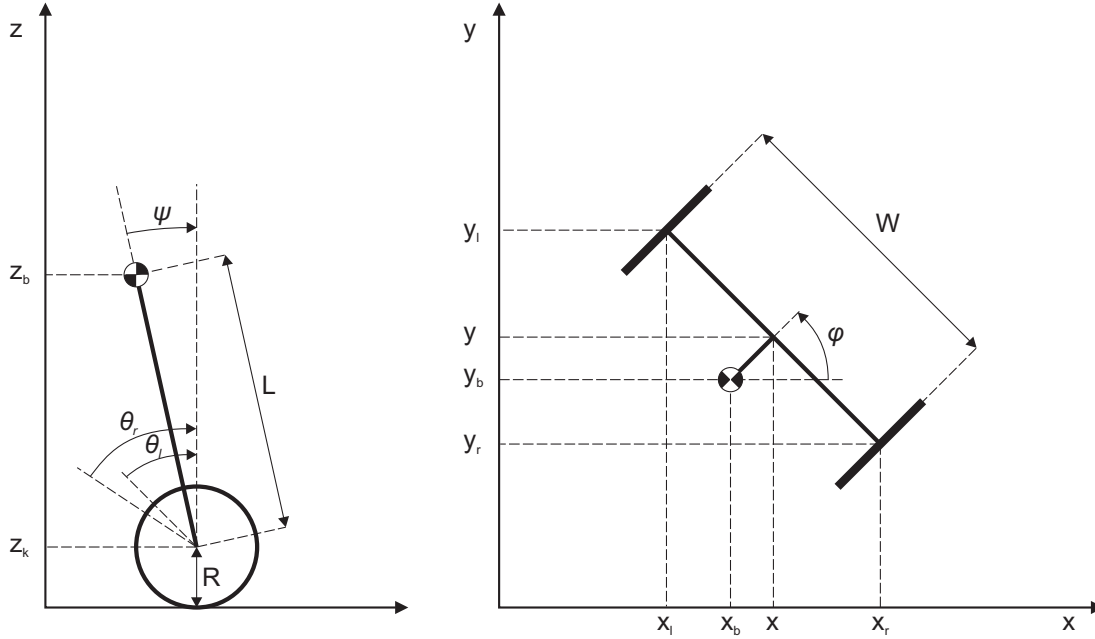
Rysunek 5.7: Diagram blokowy programu Simulink realizujący sterowanie pojedynczym robotem

5.1.3 Model matematyczny

Zastosowanie metodologii MBD wymaga opracowania modelu matematycznego obiektu. Jego uzyskanie dla opisywanego robota jest możliwe przez analizę właściwości mechanicznych. Do przeprowadzenia analizy i uzyskania modelu matematycznego

wykorzystano funkcję Eulera – Lagrange’a [99]. W kolejnych akapitach opisano oznaczenia i zależności fizyczne, następnie wyprowadzono równania ruchu. Finalnie uzyskano nieliniowy model oraz przedstawiono równania opisujące zachowanie obiektu w pobliżu górnego punktu równowagi.

Oznaczenia zmiennych stanu i parametrów fizycznych dwukołowego robota przedstawiono na rysunku 5.8.



Rysunek 5.8: Dwukołowy robot mobilny - oznaczenia. Po lewej: widok z boku, po prawej: widok z góry

Zależności pomiędzy zmiennymi stanu zostały zebrane w równaniach (5.1)–(5.4). ψ , θ , φ oznaczają odpowiednio pochylenie robota, średni kąt kół oraz obrót (orientację) robota na płaszczyźnie. Kąty ψ , θ przyjmują wartości dodatnie zgodnie z kierunkiem wskazówek zegara, w przeciwieństwie do kąta φ . Dzięki takiemu zapisowi otrzymuje się wygodną formę opisującą położenie robota na płaszczyźnie określoną przez współrzędne x, y .

$$\theta = \frac{1}{2}(\theta_l + \theta_r) \quad (5.1)$$

$$\varphi = \frac{R}{W}(\theta_r - \theta_l) \quad (5.2)$$

$$\dot{x} = R\dot{\theta} \cos \varphi \quad (5.3)$$

$$\dot{y} = R\dot{\theta} \sin \varphi \quad (5.4)$$

Przez R oznaczono promień koła, a W określa rozstaw kół (podsumowanie wszystkich wielkości fizycznych zebrano w tabeli 5.1).

Położenie środków mas kół oraz robota opisują równania:

$$\begin{aligned}x_b &= x + L \sin \psi \cos \varphi \\y_b &= y + L \sin \psi \sin \varphi \\z_b &= L \cos \psi\end{aligned}\tag{5.5}$$

$$\begin{aligned}x_l &= x - \frac{W}{2} \sin \varphi \\y_l &= y + \frac{W}{2} \cos \varphi \\z_l &= 0\end{aligned}\tag{5.6}$$

$$\begin{aligned}x_r &= x + \frac{W}{2} \sin \varphi \\y_r &= y - \frac{W}{2} \cos \varphi \\z_r &= 0\end{aligned}\tag{5.7}$$

gdzie L jest odległością pomiędzy osią obrotu kół, a środkiem ciężkości robota (bez uwzględnienia kół).

Zgodnie z twierdzeniem Königa, energia kinetyczna bryły sztywnej może zostać wyrażona jako suma energii kinetycznej masy skupionej w środku masy w ruchu postępowym (5.8) oraz energii kinetycznej ruchu obrotowego (5.9) wokół środka masy [82].

$$T_1 = \frac{m}{2}(\dot{x}_l^2 + \dot{y}_l^2) + \frac{m}{2}(\dot{x}_r^2 + \dot{y}_r^2) + \frac{M}{2}(\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2)\tag{5.8}$$

$$T_2 = \frac{J_w}{2}(\dot{\theta}_l^2 + \dot{\theta}_r^2) + \frac{J_\psi}{2}\dot{\psi}^2 + \frac{J_\varphi}{2}\dot{\varphi}^2 + \frac{J_m}{2}(\dot{\theta}_l - \dot{\psi})^2 + \frac{J_m}{2}(\dot{\theta}_r - \dot{\psi})^2\tag{5.9}$$

Założono się, że tensory momentów bezwładności wyrażone w układzie współrzędnych poszczególnych mas są macierzami diagonalnymi, oznacza to, że momenty de-wiacji zerują się. Energia potencjalna robota, przy założeniu, że punkt odniesienia znajduje się w osi kół, wyrażona jest wzorem

$$V = Mgz_b.\tag{5.10}$$

Całkowita energia mechaniczna układu jest sumą T_1 , T_2 oraz V , natomiast lagranż-
jan, wynosi

$$L = T_1 + T_2 - V.\tag{5.11}$$

Jeśli jako współrzędne uogólnione zostaną przyjęte ψ , θ , φ a lagranżjan ma postać (5.11) to równania ruchu robota można otrzymać przez rozwiązanie równań Eulera-Lagrange'a (5.12), (5.13), (5.14),

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} = F_\psi, \quad (5.12)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = F_\theta, \quad (5.13)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) - \frac{\partial L}{\partial \varphi} = F_\varphi, \quad (5.14)$$

gdzie F_ψ , F_θ , F_φ to siły uogólnione działające w osiach równoległych do współrzędnych uogólnionych, odpowiednio ψ , θ , φ . Po podstawieniu definicji (5.1) – (5.7) do równań opisujących energię (5.8), (5.9), (5.10) oraz obliczeniu pochodnych, otrzymuje się:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} = & (ML^2 + J_\psi + 2J_m)\ddot{\psi} + (MRL \cos \psi - 2J_m)\ddot{\theta} - \\ & ML^2\dot{\varphi}^2 \sin \psi \cos \psi - MgL \sin \psi, \end{aligned} \quad (5.15)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = & (MRL \cos \psi - 2J_m)\ddot{\psi} + \\ & ((2m + M)R^2 + 2J_w + 2J_m)\ddot{\theta} - MRL\dot{\psi}^2 \sin \psi, \end{aligned} \quad (5.16)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) - \frac{\partial L}{\partial \varphi} = & \left(\frac{mW^2}{2} + ML^2 \sin^2 \psi + \frac{W^2}{2R^2} (J_w + J_m) + J_\varphi \right) \ddot{\varphi} \\ & + 2ML^2\dot{\psi}\dot{\varphi} \sin \psi \cos \psi. \end{aligned} \quad (5.17)$$

Siły uogólnione F_ψ , F_θ , F_φ są zdefiniowane przez równania:

$$F_\psi = -nk_t(i_l + i_r) - f_m(2\dot{\psi} - \dot{\theta}_l - \dot{\theta}_r), \quad (5.18)$$

$$F_\theta = F_l + F_r, \quad (5.19)$$

$$F_\varphi = \frac{W}{2R}(F_l - F_r), \quad (5.20)$$

gdzie

$$F_l = -nk_t\dot{i}_l + f_m(\dot{\psi} - \dot{\theta}_l) - f_w\dot{\theta}_l, \quad (5.21)$$

$$F_r = -nk_t i_r + f_m(\dot{\psi} - \dot{\theta}_r) - f_w \dot{\theta}_r, \quad (5.22)$$

a n , k_t , f_m , f_w oraz i_l i_r oznaczają odpowiednio: przełożenie przekładni, stałą mechaniczną silnika, współczynnik tarcia w osi obrotu kół, współczynnik tarcia kół o powierzchnię oraz prąd w uzwojeniu silnika prawego i lewego. W równaniach przyjęto uproszczony model tarcia, w którym uwzględnia się wyłącznie tarcie wiskotyczne. W opisywanym rozwiązaniu sterowanie silnikami DC odbywa się za pomocą sygnału napięciowego. Uzasadnione jest założenie, że indukcyjność silnika jest pomijalnie mała. Wynika to ze znacznie krótszej stałej czasowej związanej z prądem niż stałej czasowej mechanicznej związanej z bezwładnością. Takie założenie pozwala zapisać zależność pomiędzy przyłożonym napięciem, a prądem płynącym przez uzwojenie silnika (odpowiednio dla silnika prawego i lewego l/r):

$$i_{l/r} = \frac{u_{l/r} + k_e(\dot{\psi} - \dot{\theta}_{l/r})}{R_{DC}}, \quad (5.23)$$

gdzie $u_{l/r}$ oznacza sterowanie – napięcie przyłożone do uzwojeń silnika, k_e jest stałą elektryczną silnika a R_{DC} – rezystancją uzwojeń silnika. Po podstawieniu równania (5.23) do (5.21) i (5.22) otrzymuje się równania opisujące siły uogólnione działające w poszczególnych osiach w zależności od sterowania.

$$F_\psi = -\frac{nk_t}{R_{DC}}(u_l + u_r) - \frac{2nk_t k_e f_m}{R_{DC}}(\dot{\psi} - \dot{\theta}), \quad (5.24)$$

$$F_\theta = \frac{nk_t}{R_{DC}}(u_l + u_r) + \frac{2nk_t k_e f_m}{R_{DC}}\dot{\psi} - \left(\frac{2nk_t k_e f_m}{R_{DC}} + f_w\right)\dot{\theta} \quad (5.25)$$

$$F_\varphi = \frac{W}{2R} \left(\frac{nk_t}{R_{DC}}(u_l + u_r)\right) - \frac{W^2}{2R^2} \left(\frac{2nk_t k_e f_m}{R_{DC}} + f_w\right)\dot{\varphi}, \quad (5.26)$$

Dla przejrzystości zapisu sił uogólnionych wprowadzono parametry:

$$d_1 = \frac{nk_t}{R_{DC}} \quad (5.27)$$

$$d_2 = 2\frac{nk_t k_e f_m}{R_{DC}} \quad (5.28)$$

$$d_3 = \frac{W}{R} \quad (5.29)$$

co prowadzi do następujących równań:

$$\begin{aligned} F_\psi &= -d_1(u_l + u_r) - d_2(\dot{\psi} - \dot{\theta}), \\ F_\theta &= d_1(u_l + u_r) + d_2\dot{\psi} - (d_2 + f_w)\dot{\theta} \\ F_\varphi &= d_3 d_1(u_l + u_r) - d_3^2(d_2 + f_w)\dot{\varphi}, \end{aligned} \quad (5.30)$$

Podstawiając równania (5.15), (5.16), (5.17) do równań (5.12), (5.13), (5.14) otrzymuje się równania opisujące dynamikę układu w postaci uwikłanej:

$$(ML^2 + J_\psi + 2J_m)\ddot{\psi} + (MRL \cos \psi - 2J_m)\ddot{\theta} - ML^2\dot{\varphi}^2 \sin \psi \cos \psi - MgL \sin \psi = F_\psi \quad (5.31)$$

$$(MRL \cos \psi - 2J_m)\ddot{\psi} + ((2m + M)R^2 + 2J_w + 2J_m)\ddot{\theta} - MRL\dot{\psi}^2 \sin \psi = F_\theta \quad (5.32)$$

$$\left(\frac{mW^2}{2} + ML^2 \sin^2 \psi + \frac{W^2}{2R^2}(J_w + J_m) + J_\varphi \right) \ddot{\varphi} + 2ML^2\dot{\psi}\dot{\varphi} \sin \psi \cos \psi = F_\varphi \quad (5.33)$$

Dla przejrzystości zapisu modelu nieliniowego pogrupowano równania (5.31), (5.32), (5.33) oraz wprowadzono stałe i zmienne pomocnicze.

$$\begin{aligned} c_1\ddot{\psi} + g_1(\psi)\ddot{\theta} &= F_\psi + g_2(\psi, \dot{\varphi}) \\ g_1(\psi)\ddot{\psi} + c_2\ddot{\theta} &= F_\theta + g_3(\psi, \dot{\varphi}) \\ g_4(\psi)\ddot{\varphi} &= F_\varphi - g_5(\psi, \dot{\psi}, \dot{\varphi}) \end{aligned} \quad (5.34)$$

gdzie:

$$g_1(\psi) = MRL \cos \psi - 2J_m \quad (5.35)$$

$$g_2(\psi, \dot{\varphi}) = ML^2\dot{\varphi}^2 \sin \psi \cos \psi + MgL \sin \psi \quad (5.36)$$

$$g_3(\psi, \dot{\psi}) = MRL\dot{\psi}^2 \sin \psi \quad (5.37)$$

$$g_4(\psi) = \frac{mW^2}{2} + ML^2 \sin^2 \psi + \frac{W^2}{2R^2}(J_w + J_m) + J_\varphi \quad (5.38)$$

$$g_5(\psi, \dot{\psi}, \dot{\varphi}) = 2ML^2\dot{\psi}\dot{\varphi} \sin \psi \cos \psi \quad (5.39)$$

$$c_1 = ML^2 + J_\psi + 2J_m \quad (5.40)$$

$$c_2 = (2m + M)R^2 + 2J_w + 2J_m \quad (5.41)$$

Równania (5.34) wraz z (5.35) – (5.41) oraz siłami uogólnionymi (5.30) opisują dynamikę modelu robota. W zmiennych pomocniczych zaznaczono współrzędne uogólnione i ich pochodne, od których zmienne te zależą, w celu ułatwienia analizy.

Dla pochylenia robota $\psi \neq \frac{\pi}{2} + k\pi, k \in \mathbb{Z}$, a więc dla $\cos \psi \neq 0$ układ równań (5.34) można rozwiązać stosując twierdzenie Cramera. Ponieważ pochylenie robota o 90° jest poza obszarem zainteresowania, a ponadto fizyczny model uniemożliwia taką sytuację, jako finalny model matematyczny przyjęto układ rozwikłany. Nieliniowy model obiektu może zostać wyrażony przez równania (z definicją komponentów zamieszczoną w (5.35) oraz definicją sił zawartą w równaniach (5.30)):

$$\ddot{\psi} = \frac{c_2(F_\psi + g_2(\psi, \dot{\varphi})) - g_1(\psi)(F_\theta + g_3(\psi, \dot{\psi}))}{c_1c_2 - (g_1(\psi))^2} \quad (5.42)$$

$$\ddot{\theta} = \frac{c_1(F_\theta + g_3(\psi, \dot{\psi})) - g_1(\psi)(F_\psi + g_2(\psi, \dot{\varphi}))}{c_1c_2 - (g_1(\psi))^2} \quad (5.43)$$

$$\ddot{\varphi} = \frac{F_\varphi - g_5(\psi, \dot{\psi}, \dot{\varphi})}{g_4(\psi)}. \quad (5.44)$$

Podsumowanie parametrów modelu zostało zamieszczone w tabeli 5.1.

Weryfikacja poprawności modelu matematycznego została opisana w podrozdziale 5.1.5.

Dalsza analiza obiektu, w szczególności dobór regulatora stabilizującego postawę, wymaga przeprowadzenia linearyzacji obiektu w górnym położeniu równowagi. Do linearyzacji przyjęto następujące zmienne stanu oraz dokonano podstawienia w celu uzyskania układu równań różniczkowych pierwszego rzędu.

$$x_1 = \psi \quad (5.45)$$

$$x_2 = \dot{\psi}, \quad \Rightarrow \quad \ddot{\psi} = \dot{x}_2 \quad (5.46)$$

$$x_3 = \theta \quad (5.47)$$

$$x_4 = \dot{\theta}, \quad \Rightarrow \quad \ddot{\theta} = \dot{x}_4 \quad (5.48)$$

$$x_5 = \varphi \quad (5.49)$$

$$x_6 = \dot{\varphi}, \quad \Rightarrow \quad \ddot{\varphi} = \dot{x}_6 \quad (5.50)$$

$$f_1 = x_2 \quad (5.51)$$

$$f_2 = \frac{c_2(F_{x_1} + g_2(x_1, x_6)) - g_1(x_1)(F_{x_3} + g_3(x_1, x_2))}{c_1 c_2 - (g_1(x_1))^2} \quad (5.52)$$

$$f_3 = x_4 \quad (5.53)$$

$$f_4 = \frac{c_1(F_{x_3} + g_3(x_1, x_2)) - g_1(x_1)(F_{x_1} + g_2(x_1, x_6))}{c_1 c_2 - (g_1(x_1))^2} \quad (5.54)$$

$$f_5 = x_6 \quad (5.55)$$

$$f_6 = \frac{F_{x_5} - g_5(x_1, x_2, x_6)}{g_4(x_1)} \quad (5.56)$$

Korzystając z powyższych definicji można zapisać równanie różniczkowe

$$\dot{x}(t) = f(x(t), u(t)), \quad (5.57)$$

gdzie f jest funkcją odwzorowującą $\mathbb{R}^6 \times \mathbb{R}^2 \rightarrow \mathbb{R}^6$, $x = [x_1 \cdots x_6]^\top$ jest wektorem stanu, a $u = [u_1, u_2]^\top$ – sterowaniem. Punktem równowagi układu (5.57) nazywamy parę (\bar{x}, \bar{u}) , dla której

$$f(\bar{x}, \bar{u}) = 0. \quad (5.58)$$

W przypadku górnego, niestabilnego punktu równowagi, zgodnie z oznaczeniami na rysunku 5.8, $\bar{x} = [0, 0, 0, 0, 0, 0]^\top$, $\bar{u} = [0, 0]^\top$. Procedura linearyzacji została zdefiniowana w rozdziale 2, w definicji 2.6. W przypadku równania (5.57) oraz wybranego punktu równowagi, otrzymuje się:

$$\dot{x}(t) \approx \left. \frac{\partial f}{\partial x} \right|_{\substack{x=\bar{x} \\ u=\bar{u}}} x(t) + \left. \frac{\partial f}{\partial u} \right|_{\substack{x=\bar{x} \\ u=\bar{u}}} u(t) \quad (5.59)$$

Równanie (5.59) może zostać zapisane w postaci macierzowej poprzez definicję:

$$A := \left. \frac{\partial f}{\partial x} \right|_{\substack{x=\bar{x} \\ u=\bar{u}}}, \quad B := \left. \frac{\partial f}{\partial u} \right|_{\substack{x=\bar{x} \\ u=\bar{u}}} \quad (5.60)$$

Dzięki temu otrzymuje się równania stanu opisujące zachowanie układu w górnym punkcie równowagi:

$$\dot{x} = Ax + Bu. \quad (5.61)$$

Uzyskane powyższą metodą macierze A i B mają postać macierzy blokowych:

$$A = \left[\begin{array}{c|c} A_1 & \mathbf{0} \\ \hline \mathbf{0} & A_2 \end{array} \right] \quad (5.62)$$

$$B = \left[\begin{array}{c} B_1 \\ B_2 \end{array} \right] \quad (5.63)$$

Definicję macierzy A_1 , A_2 , B_1 , B_2 podano w równaniach, odpowiednio: (5.64), (5.66), (5.65) oraz (5.67).

W górnym położeniu równowagi układ (5.61) może zostać zdekomponowany na dwa systemy:

- A_1 , B_1 — związany z liniowym ruchem robota oraz pochyleniem robota, co czyni go podobnym do układu wahadła na wózku,
- A_2 , B_2 — związany z orientacją robota na płaszczyźnie.

Macierz systemu zlinearyzowanego mają następującą postać:

Macierz stanu A_1 :

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{2,1} & a_{2,2} & 0 & a_{2,4} \\ 0 & 0 & 0 & 1 \\ a_{4,1} & a_{4,2} & 0 & a_{4,4} \end{bmatrix} \quad (5.64)$$

gdzie:

$$\begin{aligned} a_{2,1} &= \frac{c_2 \frac{\partial g_2}{\partial x_1} \Big|_{x=0}}{c_1 c_2 - (g_1|_{x=0})^2} & a_{4,1} &= \frac{g_1|_{x=0} \frac{\partial g_2}{\partial x_1} \Big|_{x=0}}{c_1 c_2 - (g_1|_{x=0})^2} \\ a_{2,2} &= \frac{d_2 \left(c_2 + \frac{\partial g_2}{\partial x_2} \Big|_{x=0} \right)}{c_1 c_2 - (g_1|_{x=0})^2} & a_{4,2} &= \frac{d_2 (c_1 + g_1|_{x=0})}{c_1 c_2 - (g_1|_{x=0})^2} \\ a_{2,4} &= \frac{c_2 d_2 + g_1|_{x=0} (d_2 + f_w)}{c_1 c_2 - (g_1|_{x=0})^2} & a_{4,4} &= \frac{-c_1 (d_2 + f_w) - d_2 g_1|_{x=0}}{c_1 c_2 - (g_1|_{x=0})^2} \end{aligned}$$

Macierz wejść B_1 :

$$B_1 = \begin{bmatrix} 0 & 0 \\ b_{2,1} & b_{2,2} \\ 0 & 0 \\ b_{4,1} & b_{4,2} \end{bmatrix} \quad (5.65)$$

gdzie:

$$b_{2,1} = \frac{-d_1(c_2 + g_1|_{x=0})}{c_1c_2 - (g_1|_{x=0})^2} \quad b_{4,1} = \frac{d_1(c_1 + g_1|_{x=0})}{c_1c_2 - (g_1|_{x=0})^2}$$
$$b_{2,2} = \frac{-d_1(c_2 + g_1|_{x=0})}{c_1c_2 - (g_1|_{x=0})^2} \quad b_{4,2} = \frac{d_1(c_1 + g_1|_{x=0})}{c_1c_2 - (g_1|_{x=0})^2}$$

Macierz stanu A_2 :

$$A_2 = \begin{bmatrix} 0 & 1 \\ 0 & a_{6,6} \end{bmatrix} \quad (5.66)$$

gdzie:

$$a_{6,6} = \frac{-d_2d_3^2}{g_5|_{x=0}}$$

Macierz wejść B_2 :

$$B_2 = \begin{bmatrix} 0 & 0 \\ b_{6,1} & b_{6,2} \end{bmatrix} \quad (5.67)$$

gdzie:

$$b_{6,1} = \frac{-d_1d_2}{g_5|_{x=0}} \quad b_{6,2} = \frac{d_1d_2}{g_5|_{x=0}}$$

Po podstawieniu zidentyfikowanych wartości macierze A i B przyjmują postać:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -31.5868 & 0.4764 & 0 & -0.4764 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 27.5889 & -0.1250 & 0 & 0.1250 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -2.6349 \end{bmatrix}, \quad (5.68)$$

$$B = \begin{bmatrix} 0 & 0 \\ -28.8897 & -28.8897 \\ 0 & 0 \\ 110.0881 & 110.0881 \\ 0 & 0 \\ -0.5055 & 0.5055 \end{bmatrix}. \quad (5.69)$$

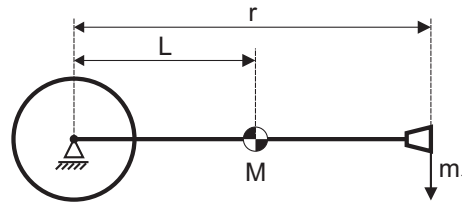
5.1.4 Identyfikacja parametrów fizycznych modelu

Przedstawiony w podrozdziale 5.1.3 model nieliniowy (5.42) zawiera 16 parametrów, które wymagają identyfikacji aby otrzymany model odpowiadał zachowaniu rzeczywistego obiektu. Istnieje wiele metod pozwalających zidentyfikować parametry modelu (patrz np. [28, 83]). Ze względu na model opisujący fizyczne zależności występujące w badanym obiekcie w celu identyfikacji parametrów przeprowadzono pomiary i eksperymenty fizyczne. Wielkości, takie jak masa robota, masa kół, rozstaw i promień kół zmierzono bezpośrednio. Parametry silnika DC (stała elektryczna, stała mechaniczna, rezystancja uzwojeń) i przekładni odczytano z not katalogowych. Pozostałe parametry zostały zidentyfikowane eksperymentalnie (np. środek ciężkości, moment bezwładności robota) lub przybliżone za pomocą obliczeń (np. moment bezwładności kół).

Środek ciężkości robota L został wyznaczony za pomocą eksperymentu fizycznego. Robot został obrócony do pozycji horyzontalnej i podparty w dwóch punktach: w wale silnika oraz na końcu dodatkowej masy (rysunek 5.9). Następnie przy pomocy wagi wyznaczono nacisk w miejscu podparcia. Dzięki znajomości całkowitej masy robota oraz odległości pomiędzy punktami podparcia wyznaczono środek ciężkości

robota.

$$L = \frac{rm_1}{m_1 + m_2} = \frac{rm_1}{M} \quad (5.70)$$



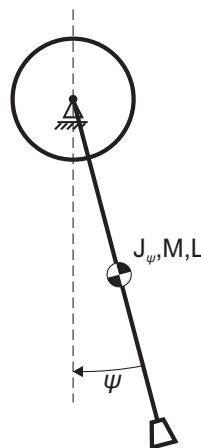
Rysunek 5.9: Wyznaczenie środka ciężkości

W modelu występuje moment bezwładności w osi ψ robota wokół środka masy J_ψ . W celu jego identyfikacji zastosowano dwuetapową procedurę. W pierwszym etapie robot został umieszczony w pozycji odwróconej - z dodatkową masą na dole (rysunek 5.10). Po odchyleniu robota od pozycji pionowej obserwowano okres drgań. Dla małych kątów okres drgań wahadła fizycznego opisuje zależność:

$$T = 2\pi \sqrt{\frac{J}{MgL}} \quad (5.71)$$

Po przekształceniu równania (5.71) otrzymuje się zależność momentu bezwładności od okresu drgań, masy, środka ciężkości i przyspieszenia grawitacyjnego:

$$J = MgL \left(\frac{T}{2\pi} \right)^2 \quad (5.72)$$



Rysunek 5.10: Wyznaczenie momentu bezwładności

Uzyskano w ten sposób moment bezwładności robota wokół osi obrotu. W drugim etapie przy pomocy twierdzenia Steinera (np. [39]) obliczono moment bezwładności wokół środka masy:

$$J_\psi = J - ML^2. \quad (5.73)$$

Współczynnik tarcia pomiędzy korpusem robota a wałem silnika został wyznaczony na podstawie optymalizacji parametrycznej wykorzystując odpowiedź modelu i obiektu. Zdefiniowano współczynnik jakości będący kwadratem błędu i znaleziono wartość tarcia, dla której współczynnik jakości przyjmuje wartość najmniejszą. Założono, że współczynnik tarcia kół o podłoże wynosi zero.

Tabela 5.1: Zestawienie parametrów fizycznych dwukołowego robota

Oznaczenie	Wartość	Jednostka	Opis
J_θ	0,104	kgm ²	Moment bezwładności w osi związanej z pochyleniem
J_ψ	0,048	kgm ²	Moment bezwładności w osi związanej z obrotem
J_w	0,0013	kgm ²	Moment bezwładności koła
J_m	0,00119	kgm ²	Moment bezwładności przekładni i rotora
M	5,41	kg	Waga robota
m	0,35	kg	Waga koła
R	0,075	m	Promień koła
L	0,102	m	Odległość środka masy od osi obrotu
W	0,4	m	Rozstaw kół
R_{DC}	1	Ω	Rezystancja uzwojeń silnika DC
k_t	0,025	N m/A	Stała mechaniczna silnika DC
k_e	0,025	N m/A	Stała elektryczna silnika DC
n	10	—	Przełożenie przekładni
f_m	0,000024	—	Współczynnik tarcia w osi obrotu kół
f_w	0,0	—	Współczynnik tarcia kół o podłoże
g	9,81	m/s ²	Przyspieszenie grawitacyjne

Zestawienie wszystkich parametrów użytych w modelu zawarto w tabeli 5.1.

5.1.5 Weryfikacja modelu matematycznego

Weryfikacja modelu matematycznego została przeprowadzona w środowisku MATLAB/Simulink. Równania modelu nieliniowego oraz zlinearyzowanego zostały zaimplementowane jako diagram programu Simulink, odpowiednio w postaci *S-funkcji* napisanej w języku C oraz bloku State-Space. W eksperymentach skorzystano z linearyzacji w dolnym - stabilnym punkcie równowagi oraz górnym, niestabilnym bę-

dającym punktem pracy robota.

Dla modelu zlinearyzowanego w górnym i dolnym położeniu równowagi obliczono wartości własne macierzy A . Zgodnie z przewidywaniami dolny punkt równowagi posiada parę sprzężoną, wszystkie wartości własne mają niedodatnie części rzeczywiste, w tym dwie wartości są równe zero. Jest to zgodne z przewidywaniami. Para sprzężona wynika z podobieństwa do układu wahadła fizycznego. Dwie zerowe wartości własne mają związek z stabilnym, ale nie asymptotycznie stabilnym, położeniem i orientacją robota. W górnym punkcie równowagi występują wartości własne leżące w prawej półpłaszczyźnie zespolonej, co oznacza niestabilny charakter układu.

Sprawdzono i porównano odpowiedź modelu dla różnych wartości początkowych i przeanalizowano otrzymane wyniki. Dla małych zmian przechylenia robota od stabilnej pozycji ruch robota przypomina ruch wahadła fizycznego - gasnące oscylacje. Jest to obserwacja zgodna z przewidywaniami. Trajektorie układu zlinearyzowanego są podobne do trajektorii układu nieliniowego, co potwierdza poprawność przeprowadzonej linearyzacji, zgodnie z twierdzeniem Hartmana-Grobmana [35].

Dla małych zmian przechylenia robota od pozycji górnej, (niestabilnej) model nieliniowy osiąga pozycję dolną, wokół której oscyluje. Ten wynik również jest zgodny z przewidywaniami. Model zlinearyzowany, ze względu na jego globalną niestabilność, wykazuje uciekanie trajektorii do nieskończoności, co również jest zgodne z przewidywaniami.

5.1.6 Odtwarzanie stanu

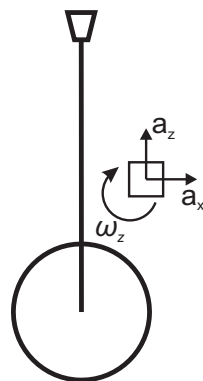
Wektor stanu obiektu zawiera sześć elementów: przechylenie robota, średni kąt kół i obrót (orientację) robota na płaszczyźnie oraz pierwsze pochodne powyższych (prędkości kątowne). Zastosowane czujniki pozwalają na pomiar czterech wartości. Enkodery inkrementalne na wale silnika pozwalają na pomiar położenia lewego i prawego koła. Czujnik bezwładnościowy pozwala zmierzyć przechylenie robota oraz prędkość kątową. Zastosowanie stabilizującego regulatora od stanu wymaga odtworzenia brakujących wielkości. Przegląd metod obserwacji stanu w systemach dynamicznych został przedstawiony w [12].

Średni kąt kół oraz obrót (orientacja) robota na płaszczyźnie są zależne od sumy $\theta = \frac{1}{2}(\theta_l + \theta_r)$ i różnicy pomiarów kąta wału silnika $\varphi = \frac{R}{W}(\theta_r - \theta_l)$, gdzie θ_r i θ_l są (po odpowiednim przeskalowaniu) wielkościami mierzonymi przez enkoder. Konieczne jest odtworzenie średniej prędkości kół oraz prędkości obrotu robota. W pracach [52, 9] zauważono, że odtworzenie prędkości z pomiarów enkoderowych jest problematyczne. Najprostszą metodą obliczenia prędkości jest numeryczne różniczkowanie pomiarów. W zadaniu stabilizacji zakres ruchu jest niewielki, a ograniczona rozdziel-

czość enkodera lub wysokie częstotliwości próbkowania powodują, że impulsy pomiarowe występują rzadko. W takim wypadku numeryczne różniczkowanie sygnału daje bardzo złe rezultaty. Innym podejściem jest zaprojektowanie obserwatora stanu (np. obserwatora Luenbergera lub filtra Kalmana), jednak duża kwantyzacja pomiaru utrudnia użycie klasycznych metod. W związku z powyższym w proponowanym rozwiązaniu zapewniono wysoką jakość pomiaru w celu ograniczenia niekorzystnych zjawisk.

Zastosowane enkodery inkrementalne mają rozdzielczość 400 impulsów na obrót. Wyjście enkoderów jest sygnałem kwadraturowym, z dwoma falami A i B przesuniętymi w fazie. Interpretacja wszystkich zboczy sygnału A i B pozwala wykryć kierunek ruchu oraz zwiększyć czterokrotnie rozdzielczość pomiaru. Dodatkowo w proponowanym rozwiązaniu enkoder został umieszczony na wale silnika, czyli znajduje się przed przekładnią. W związku z tym efektywna rozdzielczość pomiarowa jest zwielokrotniona o przełożenie przekładni. Daje to rozdzielczość 16000 impulsów na pełen obrót koła. Eksperymenty potwierdziły, że wysoka rozdzielczość pomiaru pozwala zastosować numeryczne różniczkowanie pomiaru w celu odtworzenia prędkości kół oraz prędkości obrotu robota. Dzięki tej operacji uzyskuje się cztery elementy wektora stanu.

Pozostałe dwie zmienne stanu: pochylenie robota i prędkość kątowna pochylenia są mierzone bezpośrednio przez czujnik bezwładnościowy wykonany w technologii MEMS. Przybliżone umiejscowienie czujnika oraz jego osie zostały przedstawione na rysunku 5.11.

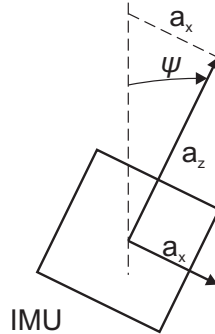


Rysunek 5.11: Układ współrzędnych czujnika bezwładnościowego

Zasada działania czujników tego typu została szczegółowo opisana np. w [98]. Do obliczenia kąta pochylenia robota można wykorzystać informację z akcelerometrów. Na dwóch osiach akcelerometru zmierzony zostanie wektor grawitacji (rysunek 5.12). Zakłada się, że przyspieszenie związane z siłą odśrodkową (ponieważ czujnik jest zamontowany poza osią obrotu) jest pomijalne. Jeśli a_x i a_z oznaczają pomiar akcelerometru w osiach zdefiniowanych na rysunku 5.11 to kąt przechylenia ψ może zostać

wyznaczony z zależności:

$$\psi_{IMU} = \arctan \frac{a_x}{a_z} \quad (5.74)$$



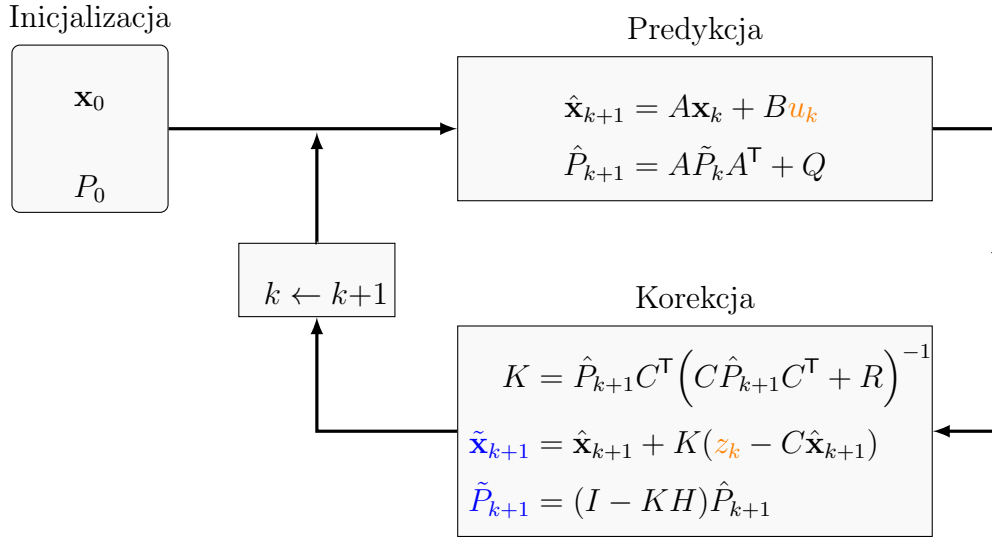
Rysunek 5.12: Wyznaczenie kąta pochylenia robota na podstawie pomiaru z akcelerometrów

Znaczną wadą akcelerometrów wykonanych w technologii MEMS jest duży stosunek szumu do sygnału.

Pomiar prędkości kątowej pochylenia robota może zostać zmierzony bezpośrednio przez żyroskop. Ze względu na konstrukcję wykonania w sygnale z żyroskopu występują różne błędy pomiarowe w szczególności dryft [48]. Powoduje to pomiar niezerowych wartości przy nieruchomym urządzeniu. Dryft żyroskopu zależy od wielu czujników m.in. od temperatury ([8]) dlatego jego kompensacja prostymi metodami (jak np. wykrywanie i odejmowanie wartości początkowej) jest niemożliwa.

Bezpośrednie zastosowanie pomiaru z akcelerometru i żyroskopu do odtworzenia pochylenia i prędkości pochylenia robota jest utrudnione ze względu na ww. wady tj. duży stosunek szumu do sygnału akcelerometru oraz dryft żyroskopu. Rozwiązaniem tego problemu jest zastosowanie filtra Kalmana do fuzji danych z obu czujników w celu poprawnego pomiaru pożądaných wartości.

Filtr Kalmana jest iteracyjnym algorytmem składającym się z dwóch etapów. Schemat algorytmu został przedstawiony na rysunku 5.13. Inicjalizacji algorytmu polega na podaniu początkowej wartości estymaty stanu \mathbf{x}_0 oraz kowariancji P_0 opisującej jej niepewność. W pierwszym kroku algorytm na podstawie modelu procesu, opisanego przez macierze A , B , oraz sterowań u_k dokonuje predykcji stanu $\hat{\mathbf{x}}_{k+1}$ oraz aktualizuje wartość kowariancji \hat{P} . W kolejnym kroku, nazywanym etapem korekcji, algorytm oblicza wzmocnienie Kalmana K . Następnie dokonywana jest korekta stanu wykorzystując dane pomiarowe z_k oraz predykcję stanu. Waga korekcji i predykcji dobierana jest w sposób optymalny ze względu na wartość niepewności obu kroków wyrażoną przez kowariancje Q i R . Wyjściem algorytmu jest estymata stanu $\tilde{\mathbf{x}}$ oraz macierz kowariancji \tilde{P} opisująca jakość estymaty. Szczegółowy opis algorytmu można znaleźć w [37].



Rysunek 5.13: Schemat blokowy filtra Kalmana (na podstawie: martin-thoma.com). Kolorem pomarańczowym oznaczono wejścia algorytmu, kolorem niebieskim oznaczono wyjścia

W przypadku odtwarzania kąta pochylenia robota filtr Kalmana można wykorzystać do fuzji danych z czujnika bezwładnościowego (akcelerometru i żyroskopu). Stan może zostać opisany przez:

$$x^{KF} = \begin{bmatrix} \psi \\ \omega_k^{bias} \end{bmatrix} \quad (5.75)$$

gdzie ψ jest odtwarzanym kątem, ω_k^{bias} określa dryft żyroskopu. Równanie stanu procesu jest następujące:

$$\begin{cases} \psi_{(k+1)} = \psi_k + (\omega_k - \omega_k^{bias})\Delta t + \vartheta_{\psi k} \\ \omega_{(k+1)}^{bias} = \omega_k^{bias} + \vartheta_{\omega k} \end{cases} \quad (5.76)$$

gdzie ω_k jest pomiarem prędkości kątowej w chwili k , czyli wartością mierzoną przez żyroskop, Δt jest okresem próbkowania, a ϑ jest białym szumem (zakłóceniem procesowym).

Równanie wyjścia ma postać

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x^{KF} + \varrho_k \quad (5.77)$$

gdzie ϱ jest białym szumem (zakłóceniem pomiarowym).

W etapie korekcji wykorzystuje się wartość kąta ψ_{IMU} wyznaczaną za pomocą pomiarów z akcelerometru przy pomocy równania (5.74). Krok polega na porównaniu estymaty z fazy predykcji przemnożonej przez macierz wyjścia (C) z wartością

pomiarów kąta:

$$e = \psi_{IMU} - \begin{bmatrix} 1 & 0 \end{bmatrix} x^{KF} \quad (5.78)$$

W celu uzyskania zapisu macierzowego przyjęto oznaczenia:

$$A = \begin{bmatrix} 0 & -\Delta t \\ 0 & 1 \end{bmatrix} \quad (5.79)$$

$$B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \quad (5.80)$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (5.81)$$

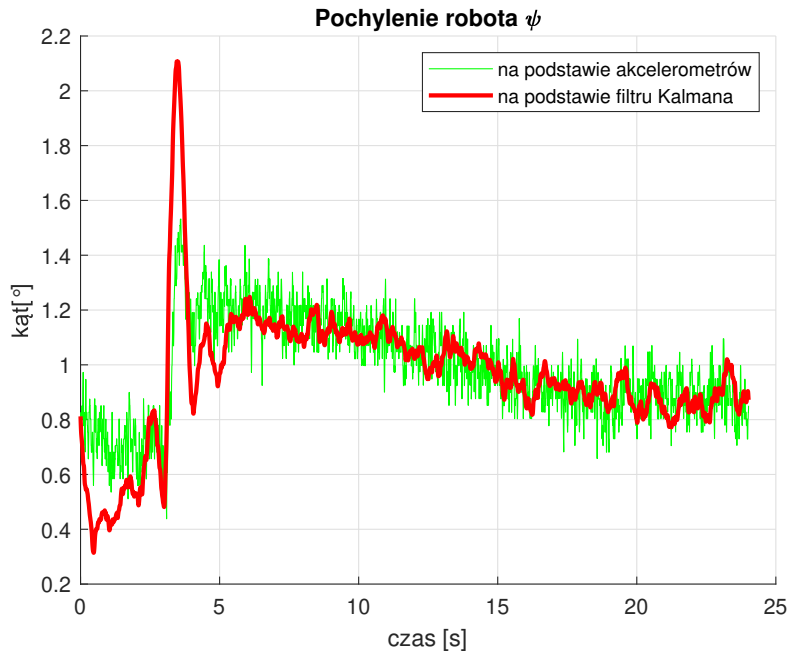
Macierze kowariancji Q (związana z szumem ϑ) oraz R (związana z szumem ϱ) zostały wyznaczone eksperymentalnie. Założono, że szum na etapie predykcji w głównej mierze związany jest z pomiarem żyroskopu, a szum na etapie korekcji związany jest z pomiarem akcelerometru. Wykonano eksperyment w którym czujnik pozostawał w bezruchu, następnie dla zebranych danych wyznaczono diagonalną macierz Q , gdzie elementy na przekątnej są wartościami wariancji zmierzonego sygnału. W takim samym eksperymencie wyznaczono skalar R , którego wartość jest wariancją sygnału z akcelerometru. Wyznaczone wartości Q i R wynoszą:

$$Q = \begin{bmatrix} 4,11 \cdot 10^{-6} & 0 \\ 0 & 4,11 \cdot 10^{-6} \end{bmatrix}, \quad R = 0,11596$$

W inicjalizacji jako początkową estymatę stanu przyjęto obliczony kąt oraz początkową wartość z żyroskopu. Zakłada się przy tym, że w momencie uruchomienia obserwatora czujnik pozostaje w bezruchu. Przyjęto również, że $P_0 = Q$.

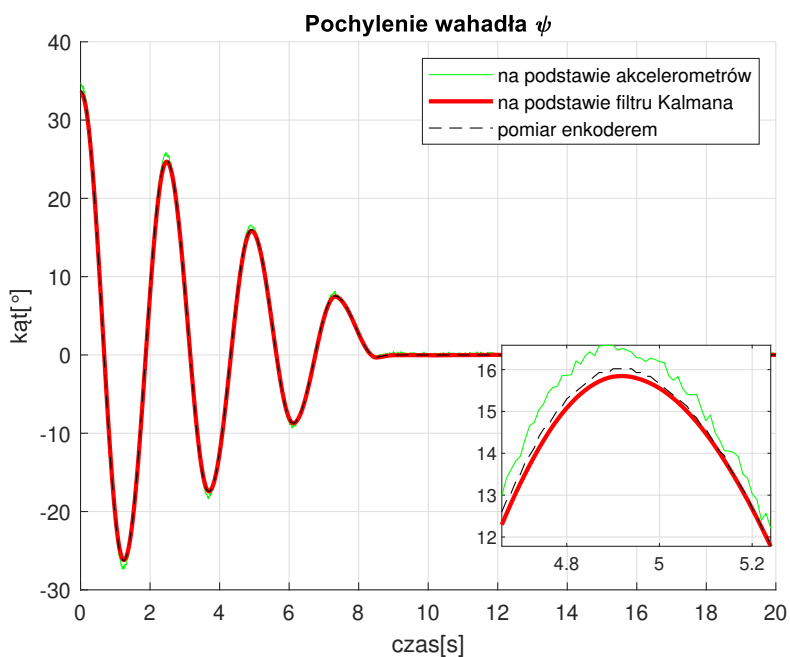
Na etapie predykcji, jako sterowanie u_k , przyjęto pomiar z żyroskopu ω . Przyjęty zapis stanu oraz równania modelu zakładają, że predykcja kąta może bazować na aktualnie zmierzonej prędkości kątowej pod warunkiem odjęcia wartości dryftu. Na etapie korekcji, jako pomiar z_k , przyjęto kąt obliczony z akcelerometrów.

Na rysunku 5.14 przedstawiono wykres wartości kąta wyznaczonego za pomocą akcelerometru oraz odtworzonego przez filtr Kalmana. Przebieg kąta odtwarzany przez filtr Kalmana jest wyraźnie mniej zaszumiony niż obliczony na podstawie wyłącznie danych z akcelerometru.



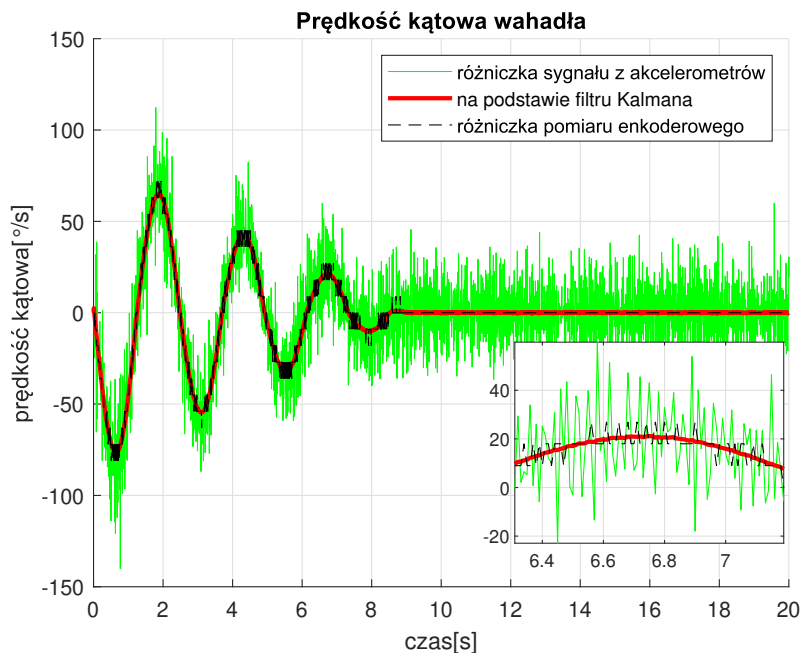
Rysunek 5.14: Porównanie odtwarzania pochylenia robota na podstawie akcelerometrów oraz filtru Kalmana, przypadek małych kątów

Na rysunku 5.15 przedstawiono wynik eksperymentu, w którym czujnik bezwładnościowy został zamontowany na wahadle, w pobliżu osi obrotu. Kąt wahadła jest dodatkowo mierzony za pomocą enkodera w celu uzyskania sygnału referencyjnego. Podczas eksperymentu wahadło zostaje odchyłone od pionu i wykonuje swobodne oscylacje. Wykres pokazuje, że przy większych wychyleniach stosunek szumu do sygnału nie jest dominujący. Wyraźnie widać również, że filtr Kalmana lepiej odwzorowuje sygnał referencyjny.



Rysunek 5.15: Porównanie odtwarzania kąta wahadła na podstawie akcelerometrów, filtra Kalmana oraz enkodera inkrementalnego przypadku dużych kątów

Na rysunku 5.16 przedstawiono odtwarzanie prędkości z sygnałów w opisywanym eksperymencie. W tym wypadku różniczka pomiarów z akcelerometru daje rezultat z dużym stosunkiem szumu do sygnału. W przypadku prędkości odtwarzanej z sygnału enkodera są widoczne błędy wynikające z kwantyzacji pomiarów. Sygnał prędkości odtworzony przez filtr Kalmana jest wolny od obu powyższych wad.



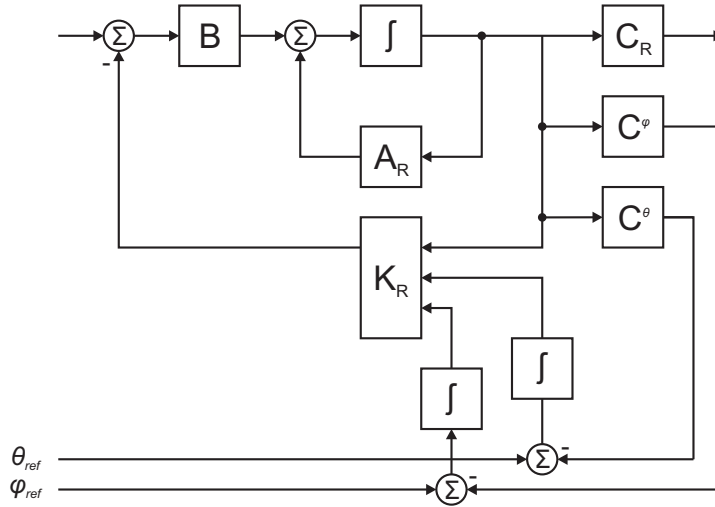
Rysunek 5.16: Porównanie odtwarzania prędkości kątowej wahadła podstawie akcelerometrów, filtra Kalmana oraz enkodera inkrementalnego, przypadek dużych kątów

5.1.7 Regulator utrzymujący pionową postawę i śledzący sygnały referencyjne prędkości liniowej i kątowej

Do utrzymania pionowej postawy robota i śledzenia referencyjnych sygnałów ruchu został wykorzystany regulator od stanu bazujący na regulatorze liniowo-kwadratowym (LQR).

Podstawowy regulator LQ (opisany w sekcji 2.13.) wykazuje bardzo dobre właściwości przy stabilizacji układu w punkcie równowagi. Jednak w zadaniach śledzenia zmiany sygnału referencyjnego konieczne jest wprowadzenie modyfikacji, która zapobiegnie powstawaniu uchybów ustalonych i poprawi właściwości regulatora. W przypadku dwukołowego pojazdu od regulatora wymagane jest śledzenie wartości zadanej dla położenia kół (θ) oraz orientacji robota (φ) (lub ich pochodnych, bezpośrednio związanych z prędkością liniową i kątową robota). Jedną z metod jest rozszerzenie stanu o całki z uchybów obu śledzonych wielkości. Metoda ta została

zaproponowana w [100] i jest wykorzystywana w stabilizacji niestabilnych pojazdów (np. [68]). Schemat regulatora od stanu rozszerzonego o całki z uchybu regulacji sygnałów θ i φ został przedstawiony na rysunku 5.17.



Rysunek 5.17: Schemat blokowy regulatora od rozszerzonego stanu

W przypadku niestabilnego pojazdu mobilnego opisanego zlinearyzowanym równaniem (5.61) rozszerzenie stanu opiera się o definicję dodatkowych dwóch wyjść:

$$\begin{cases} \dot{x} = Ax + Bu \\ \theta = C^\theta x \\ \varphi = C^\varphi x \end{cases} \quad (5.82)$$

gdzie:

$$C^\theta = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (5.83)$$

$$C^\varphi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (5.84)$$

Następnie wprowadzono dwie dodatkowe zmienne stanu będące całkami z uchybów śledzonych wartości.

$$z_\theta = \int (\theta_{ref} - \theta) dt \quad (5.85)$$

$$z_\varphi = \int (\varphi_{ref} - \varphi) dt \quad (5.86)$$

Rozszerzony stan $x_R \in \mathbb{R}^8$ jest zdefiniowany następująco

$$x_R = [x | z_\theta | z_\varphi]^T \quad (5.87)$$

Macierze stanu rozszerzone systemu utworzone są w następujący sposób:

$$A_R = \left[\begin{array}{c|c} A_{6 \times 6} & \mathbf{0}_{6 \times 2} \\ \hline C_{1 \times 6}^\theta & \mathbf{0}_{1 \times 2} \\ \hline C_{1 \times 6}^\varphi & \mathbf{0}_{1 \times 2} \end{array} \right]_{8 \times 8} \quad (5.88)$$

$$B_R = \left[\begin{array}{c} B_{6 \times 2} \\ \hline \mathbf{0}_{2 \times 2} \end{array} \right]_{8 \times 2} \quad (5.89)$$

$$\dot{x}_R = A_R x_R + B_R u \quad (5.90)$$

Dla tak zdefiniowanego systemu wskaźnik jakości przyjmuje postać:

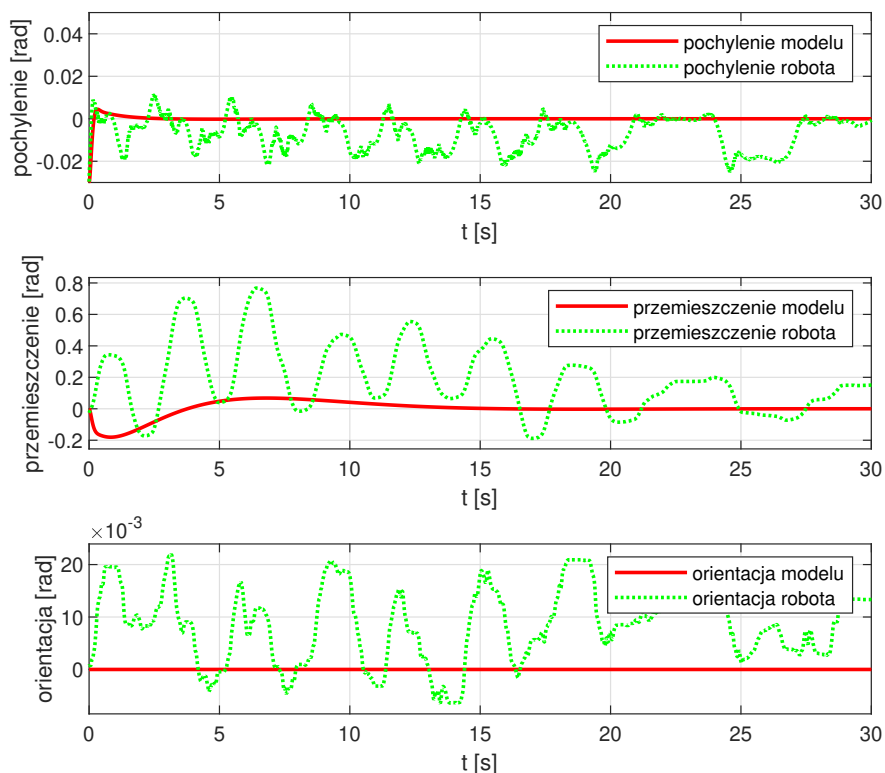
$$J = \int_0^\infty (x_R(\tau)^T Q_R x_R(\tau) + u(\tau)^T R_R u(\tau)) \, d\tau \quad (5.91)$$

Rozwiązanie równania Riccatiego pozwala otrzymać wzmocnienie od stanu K_R będące optymalnym regulatorem dla rozszerzonego systemu (5.90).

Podając jako sygnał referencyjny prędkości liniową ($\dot{\theta}$) i kątową ($\dot{\varphi}$) robota oraz wynikające z nich położenie kół (θ) i orientację robota (φ) można zadać referencyjną trajektorię dla pojazdu, umożliwiając osiągnięcie zadanego przemieszczenia.

Taka konstrukcja regulatora od stanu pozwala jednocześnie utrzymać pionową postawę jak i zadawać ruch robota na płaszczyźnie.

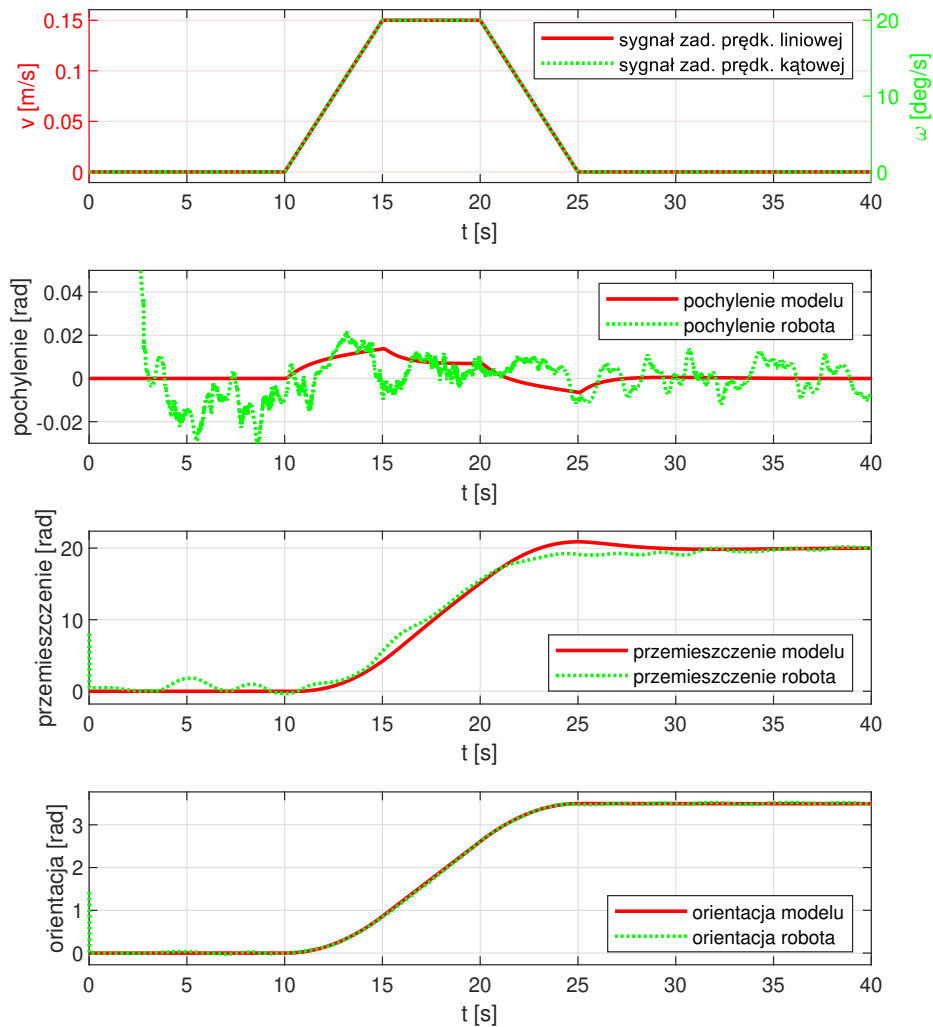
Na rysunku 5.18 przedstawiono przebieg sygnałów pochylenia, położenia (średni kąt kół) i orientacji modelu robota oraz rzeczywistego obiektu w zadaniu stabilizacji.



Rysunek 5.18: Przebieg wybranych elementów stanu robota w zadaniu stabilizacji

Początkowe pochylenie modelu robota zostało określone na $-0,03$ rad, co jest wartością podobną do początkowego pochylenia obiektu w momencie uruchomienia algorytmu stabilizującego. Na przebiegach widać, że model asymptotycznie zbiega do punktu równowagi. Obiekt natomiast oscyluje w pobliżu punktu równowagi, aktywnie korygując swoją pozycję w celu utrzymania pionowej postawy. Powyższe przebiegi potwierdzają poprawność regulatora stabilizującego postawę robota.

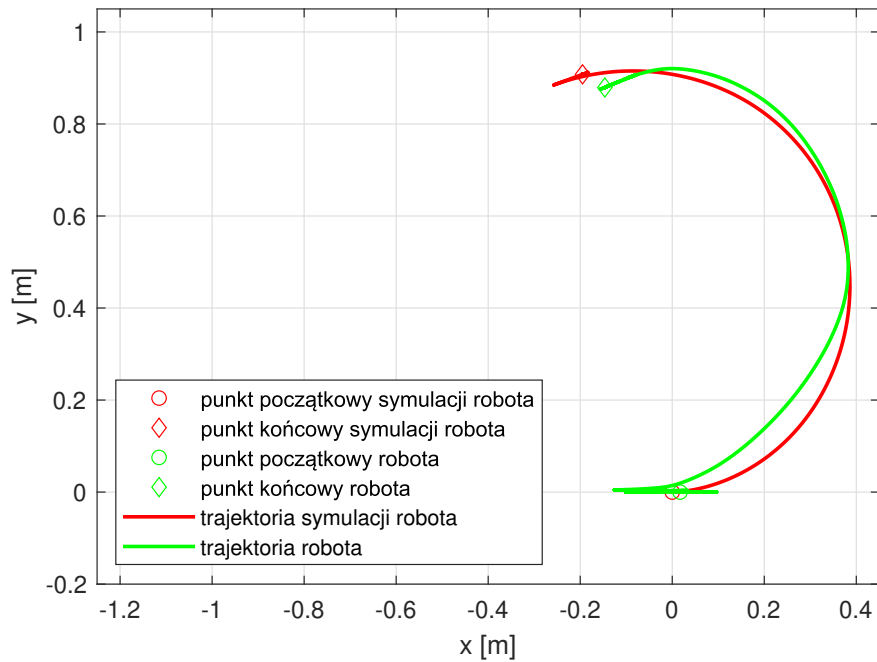
Na rysunku 5.19 przedstawiono przebieg sygnału referencyjnego oraz przebiegi sygnałów pochylenia, położenia (średni kąt kół) i orientacji modelu robota oraz rzeczywistego obiektu podczas realizacji zadanego ruchu.



Rysunek 5.19: Przebieg wybranych elementów stanu robota w zadaniu nadążaniu za sygnałem referencyjnym

Pierwszy wykres przedstawia sygnał referencyjny prędkości liniowej robota v oraz prędkości kątowej robota ω . Oba przebiegi zostały zdefiniowane przy pomocy profilu trapezowego. Taki sygnał dzieli ruch na trzy fazy - rozpędzanie, ruch ze stałą prędkością oraz wyhamowanie. Z pozostałych trzech wykresów widać, że zarówno model jak i obiekt z proponowanym regulatorem realizuje zadany ruch.

Na rysunku 5.20 przedstawiono ruch modelowanego robota oraz rzeczywistego obiektu na płaszczyźnie XY. Okręgami oznaczono położenia początkowe, diamentem zaznaczono pozycje kątowe.



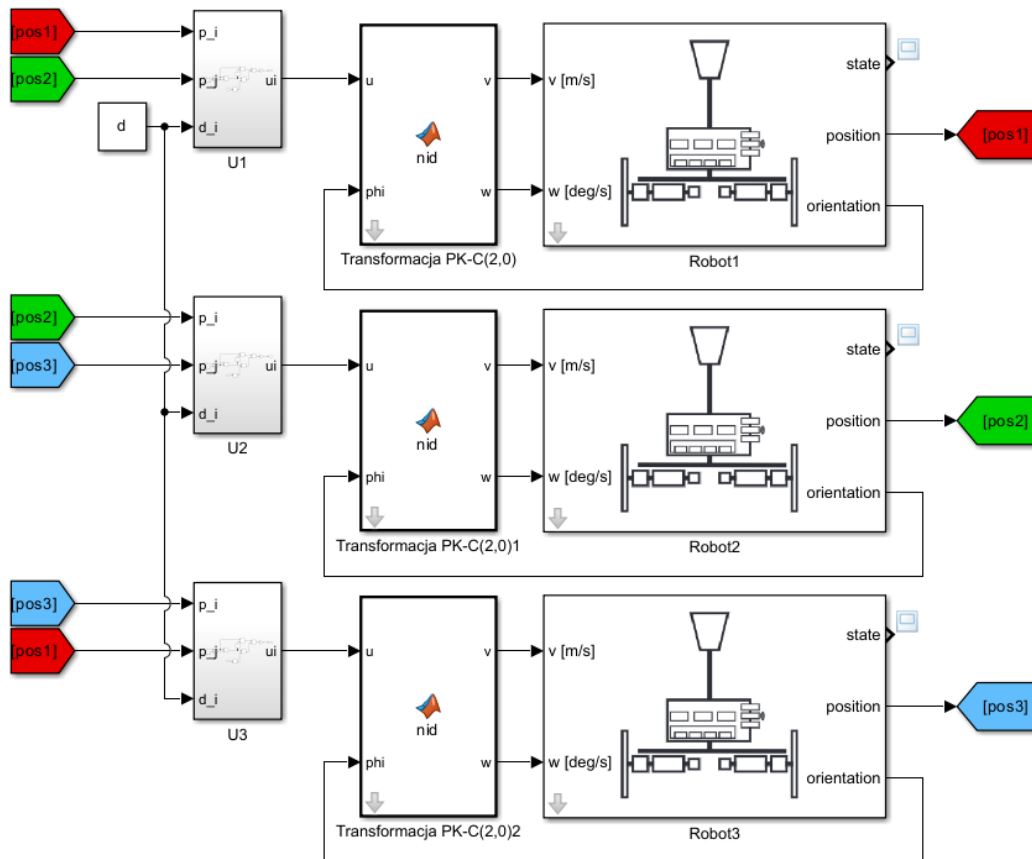
Rysunek 5.20: Porównanie trajektorii ruchu modelu symulacyjnego robota oraz rzeczywistego przebiegu

Na wykresie 5.20 można zaobserwować różnice w realizacji ruchu pomiędzy obiektem i modelem. Proponowany regulator aktywnie nadąża za sygnałami referencyjnymi ale nie koryguje położenia robota na płaszczyźnie względem trajektorii. Podążanie za zadaną trajektorią nie jest wymagane do realizacji algorytmów sterowania formacją.

Powyższe wyniki potwierdzają, że konstrukcja proponowanego regulatora poprawnie stabilizuje postawę robota oraz umożliwia ruch robota w odpowiedzi na zadane prędkości: liniową i kątową.

5.2 Ogólna postać algorytmu sterującego zespołem niestabilnych robotów mobilnych

Na rysunku 5.21 przedstawiono diagram blokowy, który realizuje zadanie sterowania formacją trzech robotów mobilnych w algorytmie cyklicznej pogoni. Na jego przykładzie zostanie przedstawiona ogólna postać algorytmu sterującego zespołem niestabilnych robotów mobilnych.

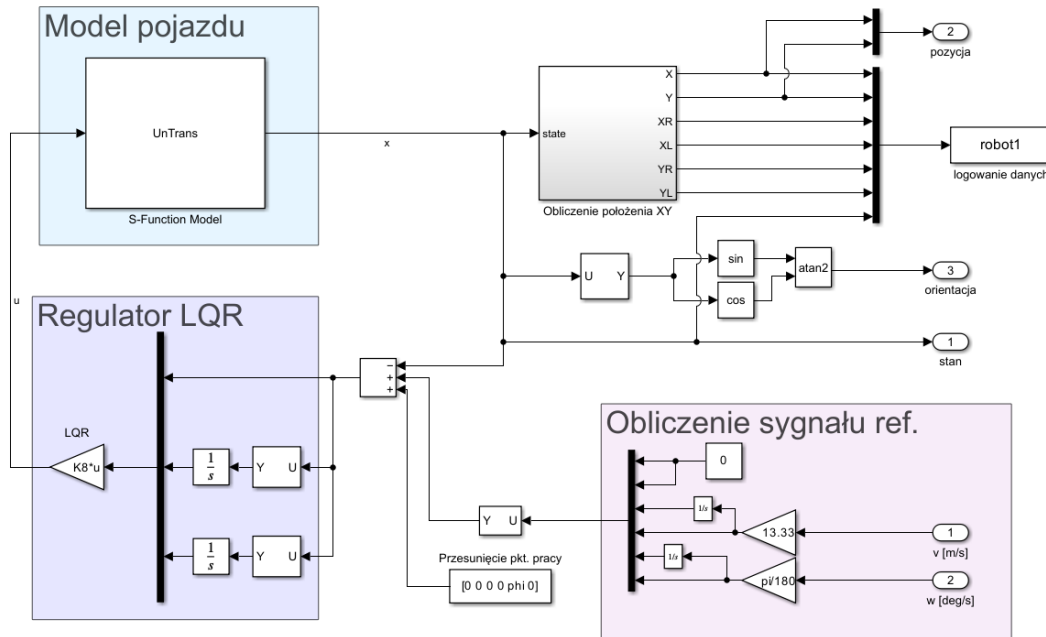


Rysunek 5.21: Schemat blokowy algorytmu sterującego formacją trzech robotów

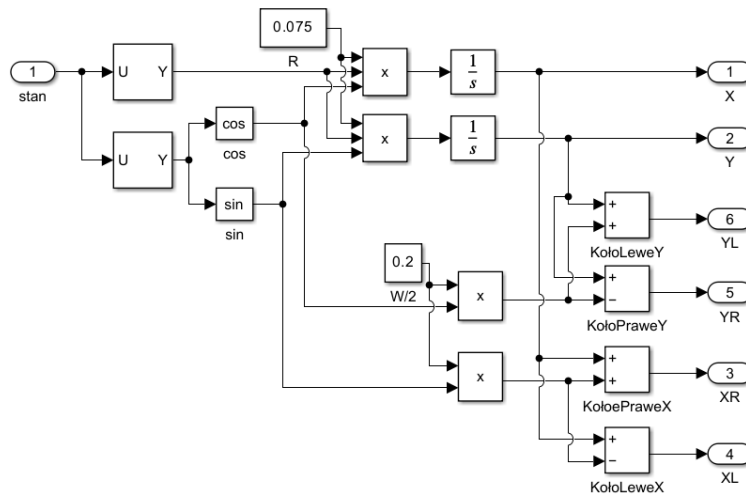
Algorytm sterujący zespołem robotów mobilnych składa się z trzech elementów:

- Wewnętrzna pętla sprzężenia zwrotnego. Składa się na nią regulator utrzymujący pionową postawę i śledzący sygnały referencyjne prędkości liniowej i kątowej 5.22. Dodatkowo w bloku tym obliczane są sygnały referencyjne wyrażone w jednostkach opisanego modelu. Ostatnim elementem jest schemat blokowy obliczający położenie robota na płaszczyźnie (oraz jego kół) 5.23, a także normalizujący orientację robota.
- Przekształcenie sterowania obliczonego dla punktów kinematycznych na kinematykę robota nieholonomicznego. Jest to implementacja przekształcenia opisanego w 4.2. W przeciwieństwie do pojazdów dostępnych w Robotarium, niestabilny robot nie ma wyraźnego przodu i tyłu, w związku z czym dopuszcza się ruch robota w obu kierunkach.
- Zewnętrzna pętla sprzężenia zwrotnego. Zawiera ona algorytm sterujący realizujący zadania sterowania formacją. Ten element zależy od wybranego zadania oraz liczby robotów. Każdy robot posiada swój regulator, który oblicza sterowanie. Postać sterowania to wektor prędkości, które pozwala osiągnąć zadane

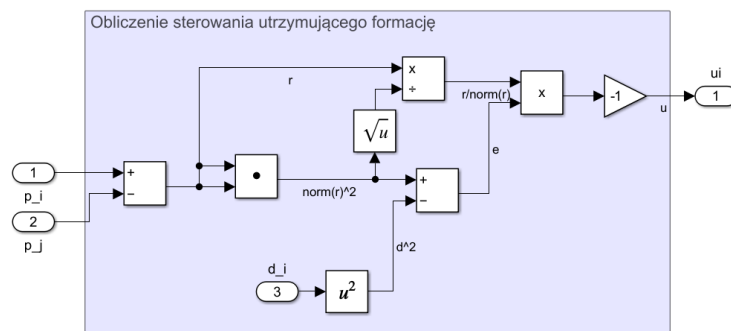
odległości pomiędzy robotami. Pętla ta wykorzystuje informacje o położeniu robotów. Schemat blokowy został przedstawiony na rysunku 5.24.



Rysunek 5.22: Schemat blokowy wewnętrznej pętli sprzężenia zwrotnego oraz modelu robota



Rysunek 5.23: Schemat blokowy modułu obliczającego położenie robota na płaszczyźnie



Rysunek 5.24: Schemat blokowy zewnętrznej pętli sprzężenia zwrotnego dla pojedynczego robota

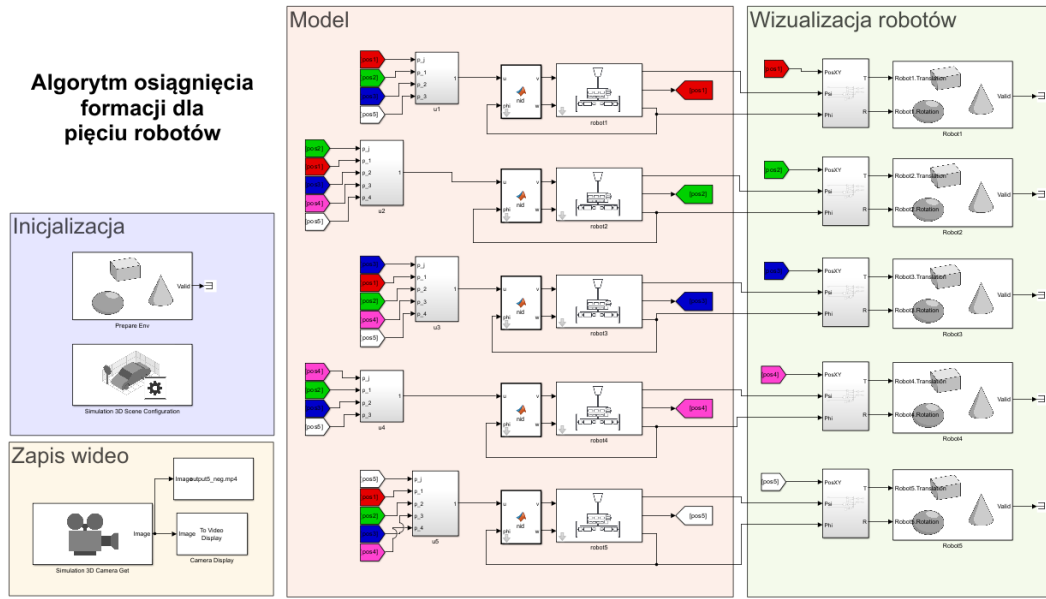
5.3 Eksperymenty symulacyjne

5.3.1 Metodologia badań symulacyjnych

Symulacje zostały wykonane przy pomocy oprogramowania MATLAB/Simulink. Dla każdego eksperymentu przygotowano program w postaci diagramu blokowego, który wykorzystywał kilka instancji modelu pojazdu. Każdy model pojazdu posiada indywidualny regulator od stanu do stabilizacji postawy oraz śledzenia sygnałów referencyjnych. Model matematyczny niestabilnego, dwukołowego pojazdu został zapisany w postaci *S-funkcji*, co pozwoliło skrócić czas wykonywania symulacji. Założono, że wymieniane informacje pomiędzy robotami pozorują posiadanie przez robota idealnego sensora mierzącego kierunek i odległość do sąsiadów. Ponadto przyjęto, że roboty mierzą swoje położenie synchronicznie z częstotliwością 100 Hz. W symulacji założono, że roboty poruszają się po nieograniczonej, nienachylonej płaszczyźnie oraz, że nie występuje poślizg kół.

Do wizualizacji eksperymentów użyto narzędzia 3D Animation™ [110], do którego zaimportowano uproszczony model geometryczny pojazdu. Prezentowane wizualizacje obrazują ruch robotów na płaszczyźnie, orientację robotów oraz ich pochylenie. Kolor robotów jest zgodny z kolorem odpowiadających przebiegów na wykresach. Wizualizacje eksperymentów symulacyjnych umieszczono na platformie YouTube, gdzie można je obejrzeć korzystając z odnośników zebranych w tabeli 6.2.

Przykładowy diagram stanu, wraz z narzędziami do wizualizacji, został przedstawiony na rysunku 5.25. W zaznaczonym obszarze Model znajduje się pięć instancji modelu pojazdu wraz z wewnętrzną pętlą sprzężenia zwrotnego. Kolorowe etykiety oznaczają położenie robota. Na ich podstawie każdy robot dokonuje obliczenia względnego położenia sąsiadów. Informacja ta jest wykorzystywana przez zewnętrzną pętlę sprzężenia zwrotnego, która realizuje algorytm sterowania formacją.



Rysunek 5.25: Diagram stanu dla eksperymentu symulacyjnego opisującego przypadek sterownia formacją pięciu robotów

5.3.2 Przykład wprowadzający

Eksperyment polega na weryfikacji podstawowego przypadku, polegającego na osiągnięciu zadanej odległości przez dwa roboty. Jest to scenariusz podobny do opisywanego w przykładzie wprowadzającym 3.1.1, gdzie dotyczył punktów kinematycznych oraz do opisanego w eksperymencie 4.4.1, gdzie dotyczył robotów klasy $(2,0)$. Istotną różnicą są obiekty sterowania, które w tym eksperymencie stanowią dwa niestabilne pojazdy mobilne.

Podczas eksperymentów symulacyjnych weryfikowano parametry układu regulacji, ograniczenia dla zadanej prędkości oraz wpływ warunków początkowych na przebieg eksperymentu.

Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned}
 u_1 &= -\frac{p_1 - p_2}{\|p_1 - p_2\|} (\|p_1 - p_2\|^2 - d^2) \\
 u_2 &= -\frac{p_2 - p_1}{\|p_1 - p_2\|} (\|p_2 - p_1\|^2 - d^2),
 \end{aligned} \tag{5.92}$$

gdzie $u_1, u_2 \in \mathbb{R}^2$ są wektorami sterującymi odpowiednio dla robota 1 i 2, $p_1, p_2 \in \mathbb{R}^2$ oznacza położenie robotów, a d jest zadaną odległością. W przykładzie przyjęto, bez straty ogólności, odległość $d = 1$ m. Sterowania dla obu robotów są symetryczne i różnią się zadanym kierunkiem ruchu. W stosunku do wcześniej stosowanych postaci sterowania, wprowadzono normalizację wektora kierunku ruchu

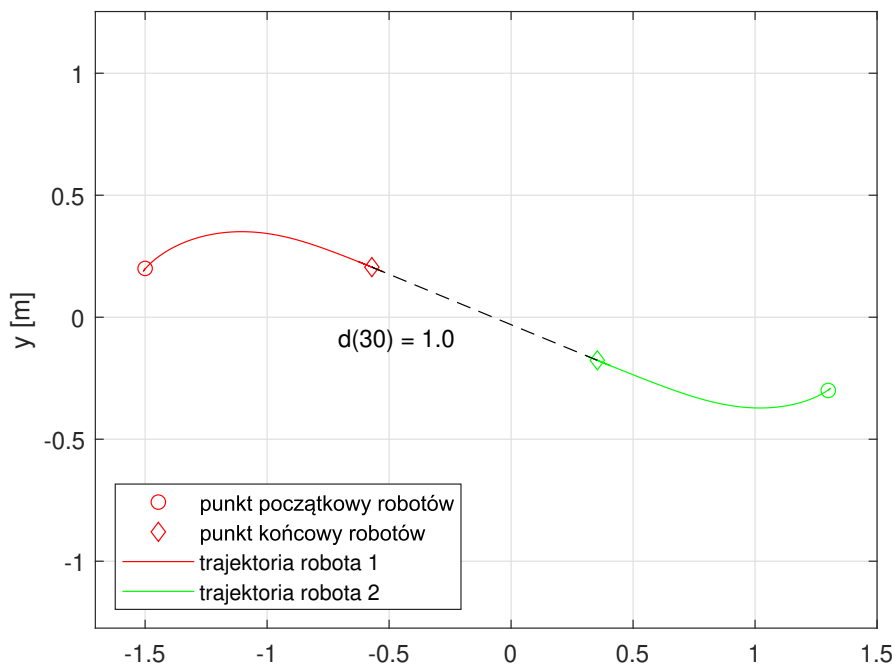
pojazdu (uzasadnienie tej zmiany podano w sekcji 5.3.4).

Zakłada się, że informacja o odległości pomiędzy robotami jest mierzona w jednakowy sposób przez każdego z nich. Pierwszy człon w równaniu (5.92) jest odpowiedzialny za określenie kierunku ruchu. Drugi człon określa różnicę pomiędzy zadaną odległością pomiędzy robotami a wartością aktualną. Jeśli roboty znajdują się zbyt blisko siebie, to zadaniem sterowania jest ich oddalenie na oczekiwaną odległość. Jeśli znajdują się w odległości większej niż zadana, to sterowanie ma za zadanie zbliżyć je na oczekiwaną odległość. Zastosowanie takiego sterowania nie wymaga istnienia globalnego układu współrzędnych. Oznacza to również, że końcowe położenie robotów w globalnym układzie nie jest przedmiotem sterowania.

Obliczone za pomocą równania (5.92) sterowanie jest odwzorowane na kinematykę dwukołowego robota za pomocą przekształcenia opisanego w sekcji 4.2. Zakłada się, że ze względu na symetrię robota, robot może poruszać się w przód lub w tył.

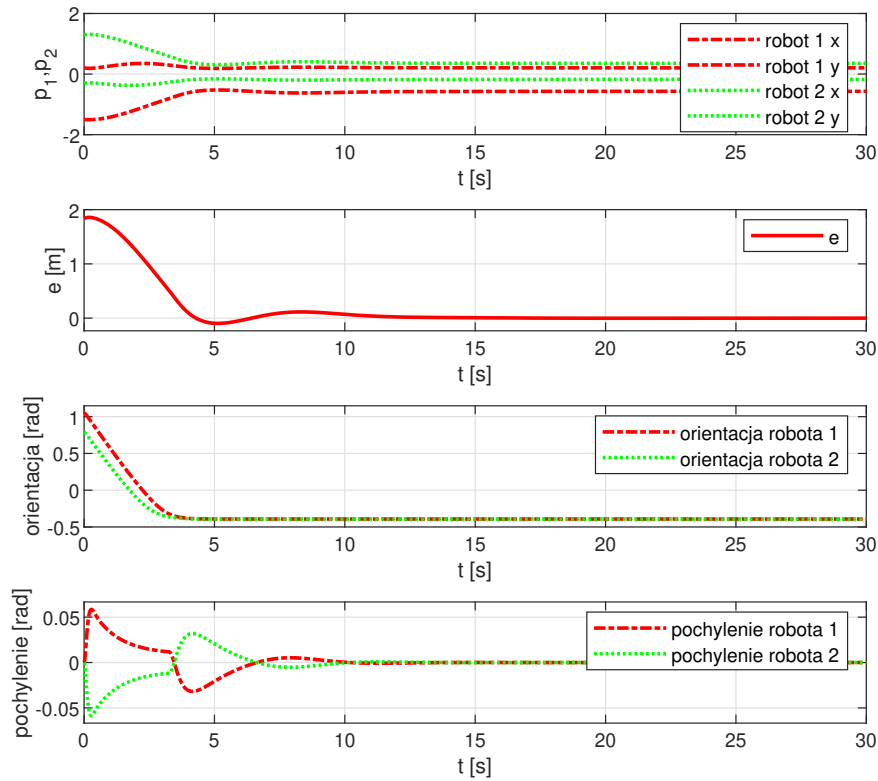
Dla warunków początkowych (5.93) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.26

$$\begin{bmatrix} p_1(0) \\ \varphi_1(0) \end{bmatrix} = \begin{bmatrix} -1,5 \\ 0,2 \\ \frac{\pi}{3} \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}, \quad \begin{bmatrix} p_2(0) \\ \varphi_2(0) \end{bmatrix} = \begin{bmatrix} 1,3 \\ -0,3 \\ \frac{\pi}{4} \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix} \quad (5.93)$$



Rysunek 5.26: Formacja dwóch robotów

Dla warunków początkowych (5.93) odległość początkowa pomiędzy robotami jest większa niż zadana. W związku z powyższym roboty muszą zbliżyć się do siebie. Ich początkowa orientacja powoduje, że roboty muszą wykonać obrót w pierwszej fazie eksperymentu aby móc zbliżyć się do siebie.

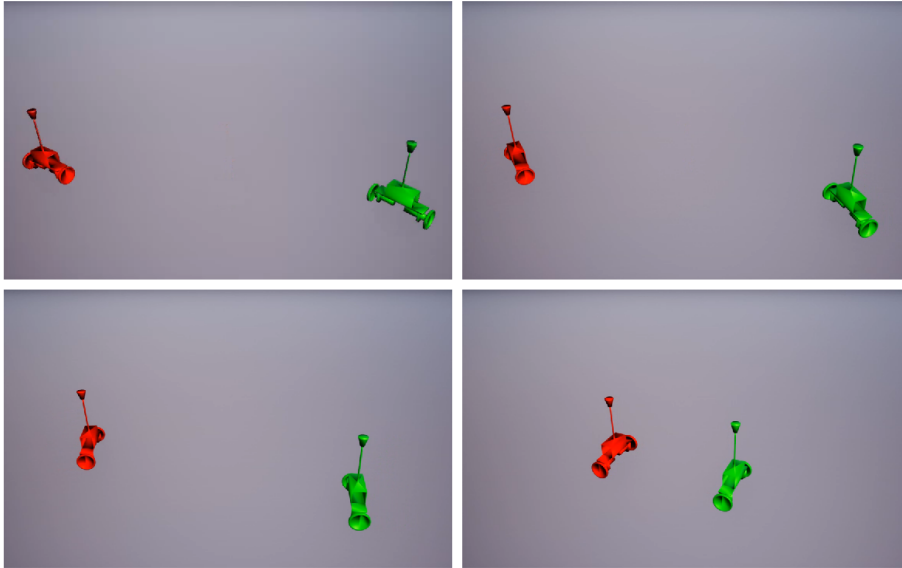


Rysunek 5.27: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Na wykresie 5.27 przedstawiono przebieg położenia robotów, uchyb wzajemnej odległości (czyli wartość $e = \|p_1 - p_2\| - d$), orientację oraz pochylenie robotów.

Przebieg czasowy pokazuje, że po około 10 sekundach pojazdy osiągną zadane względne położenie. W pierwszej fazie eksperymentu orientacja robotów zmienia się w taki sposób aby skierowały się one w swoją stronę. Zmiana orientacji oraz położenia robota powodują niewielkie pochylenia robota od postawy pionowej, wynoszące poniżej 3° , które można zaobserwować na czwartym wykresie.

Na rysunku 5.28 przedstawiono migawki z wizualizacji eksperymentu w początkowej fazie, a następnie w pierwszej, drugiej i siódmej sekundzie eksperymentu.



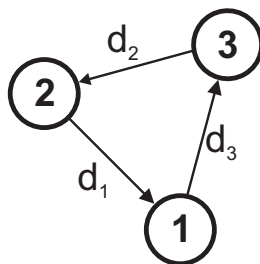
Rysunek 5.28: Migawki z wizualizacji eksperymentu symulacyjnego dla początku eksperymentu, pierwszej, drugiej, i siódmej sekundy

Wykonano eksperymenty dla różnych warunków początkowych, które potwierdziły poprawność proponowanego algorytmu sterującego. Niezależnie od pozycji początkowej robotów oraz ich orientacji odległość pomiędzy robotami asymptotycznie zmierza do zadanej wielkości.

Nagranie z przebiegu eksperymentu zostało zamieszczone pod adresem <https://youtu.be/r64GVAkxASU>.

5.3.3 Algorytm cyklicznej pogoni

Eksperyment polega na badaniu przypadku trzech robotów mobilnych, których zadaniem jest osiągnięcie formacji trójkątnej bazując wyłącznie na śledzeniu jednego sąsiada. Przypadek taki opisuje graf z cyklem przedstawiony na rysunku 5.29. Jest to scenariusz podobny do opisywanego w przykładzie wprowadzającym 3.1.2, gdzie dotyczył punktów kinematycznych oraz opisanego w eksperymencie 4.4.2 i przetestowanego na robotach klasy (2,0). Istotną różnicą są obiekty sterowania, które w tym eksperymencie stanowią trzy niestabilne pojazdy mobilne.



Rysunek 5.29: Digraf opisujący wymianę informacji w algorytmie cyklicznej pogoni dla trzech robotów

Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned} u_1 &= -\frac{p_1 - p_2}{\|p_1 - p_2\|} (\|p_1 - p_2\|^2 - d_1^2) \\ u_2 &= -\frac{p_2 - p_3}{\|p_2 - p_3\|} (\|p_2 - p_3\|^2 - d_2^2) \\ u_3 &= -\frac{p_3 - p_1}{\|p_3 - p_1\|} (\|p_3 - p_1\|^2 - d_3^2) \end{aligned} \quad (5.94)$$

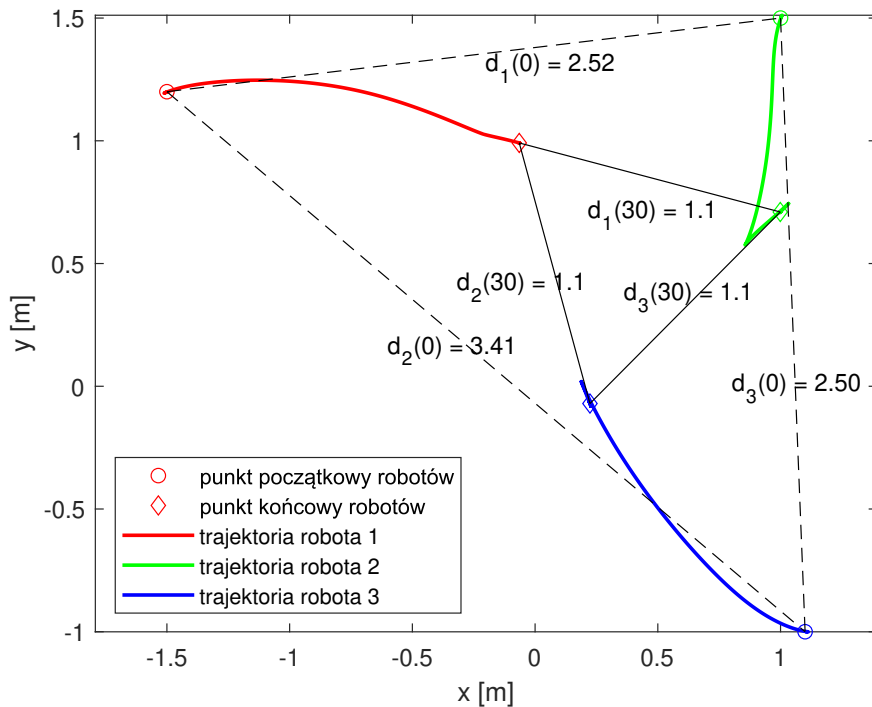
gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, a d_i są zadanymi odległościami. Precyzyjną definicję sąsiedztwa i zadanymi odległościami przedstawiono za pomocą grafu 5.29. W przykładzie przyjęto, bez straty ogólności, $d_1 = d_2 = d_3 = 1,1$ m.

Dla warunków początkowych (5.95) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.30

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} -1,5 & 1,0 & 1,1 \\ 1,2 & 1,5 & -1,0 \\ \frac{\pi}{6} & \frac{\pi}{3} & 0 \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}, \quad (5.95)$$

gdzie

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} p_1(0) & p_2(0) & p_3(0) \\ \varphi_1(0) & \varphi_2(0) & \varphi_3(0) \end{bmatrix}.$$

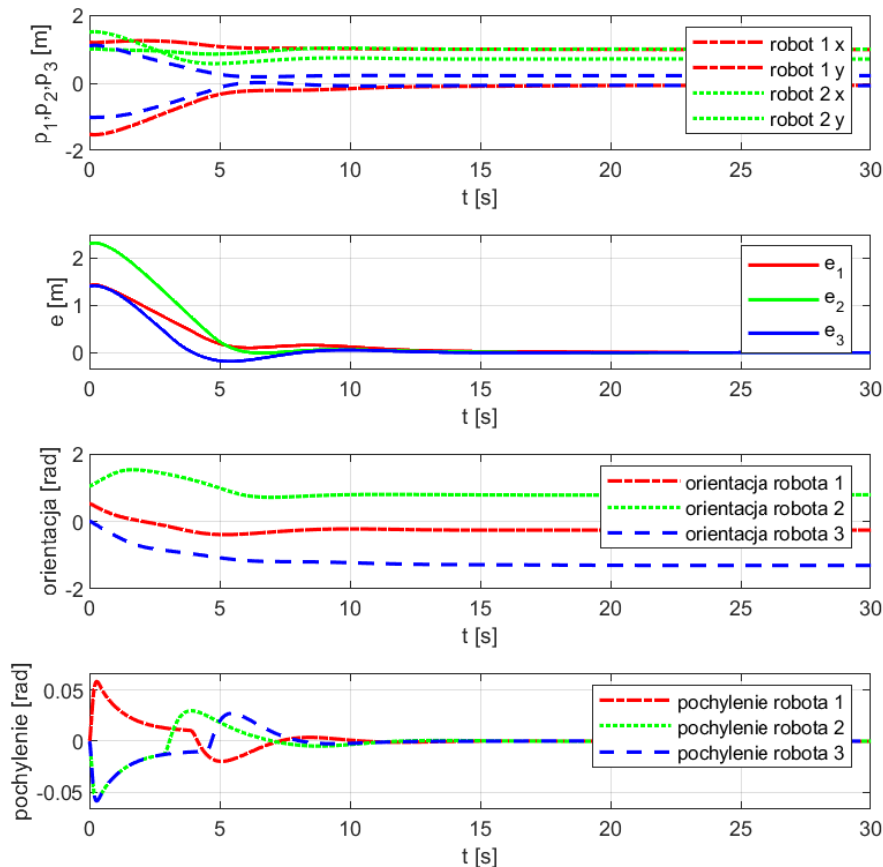


Rysunek 5.30: Formacja robotów w algorytmie cyklicznej pogoni

Na wykresie 5.30 zaznaczono początkowe odległości pomiędzy robotami (czarna, przerywana linia) oraz odległości osiągnięte w 30 s eksperymentu (czarna, ciągła linia). Dla warunków początkowych (5.95) odległości początkowe pomiędzy robotami są większe niż zadane. W związku z powyższym roboty muszą zbliżyć się do siebie. Początkowe orientacje robotów muszą zostać skorygowane przez algorytm sterujący. Poszczególne roboty kierują się w stronę swojego sąsiada. Ze względu na jednokierunkową wymianę informacji trajektorie robotów przypominają łuki okręgów skąd wywodzi się nazwa algorytmu - cykliczna pogon (ang. *cyclic pursuit*).

Na wykresie 5.31 przedstawiono przebieg położenia robotów, uchyb wzajemnej odległości (czyli wartości $e_1 = \|p_1 - p_2\| - d_1$), orientację oraz pochylenie robotów.

Przebieg czasowy pokazuje, że po około 10 sekundach pojazdy osiągają zadane względne położenie. W około 5 sekundzie można zaobserwować przeregulowanie pozycji robotów. Jego korekta wymaga zmiany kierunku ruchu, co wywołuje przechylenie robotów widoczne na czwartym wykresie. W pierwszej fazie eksperymentu orientacja robotów zmienia się w taki sposób, aby skierowały się one w stronę swojego sąsiada. Zmiana orientacji oraz położenia robota powodują niewielkie pochylenia robota od postawy pionowej, wynoszące poniżej 3° , które można zaobserwować na czwartym wykresie.



Rysunek 5.31: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

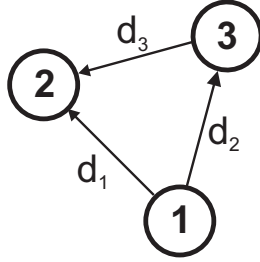
Wykonano eksperymenty dla różnych warunków początkowych, które potwierdziły poprawność proponowanego algorytmu sterującego - uchyby zadanych odległości asymptotycznie zbiegają do wartości 0.

Nagranie z przebiegu eksperymentu zostało zamieszczone pod adresem <https://youtu.be/R--4AacnAJI>.

5.3.4 Algorytm śledzenia stacjonarnego lidera

Eksperyment polega na badaniu formacji trzech robotów mobilnych, w przypadku nierównorzędnych ról. Pierwszemu z robotów zostaje przypisana rola lidera. Pozostałe dwa roboty mają za zadanie osiągnąć zadaną formację, drugi robot na podstawie pomiaru odległości do lidera, a trzeci na podstawie pomiaru odległości do dwóch sąsiadów.

Wymiana informacji w opisywanym zagadnieniu została opisana w sekcji 4.4.3. Wagi zostały zdefiniowane za pomocą grafu na rysunku 5.32.



Rysunek 5.32: Digraf opisujący wymianę informacji w przypadku trzech robotów, z których pierwszy jest liderem, drugi podąża za pierwszym a trzeci za pierwszymi dwoma

Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned}
 u_1 &= 0 \\
 u_2 &= -\frac{p_2 - p_1}{\|p_3 - p_1\|} (\|p_2 - p_1\|^2 - d_1^2) \\
 u_3 &= -\frac{p_3 - p_1}{\|p_3 - p_1\|} (\|p_3 - p_1\|^2 - d_2^2) + \\
 &\quad -\frac{p_3 - p_2}{\|p_3 - p_2\|} (\|p_3 - p_2\|^2 - d_3^2)
 \end{aligned} \tag{5.96}$$

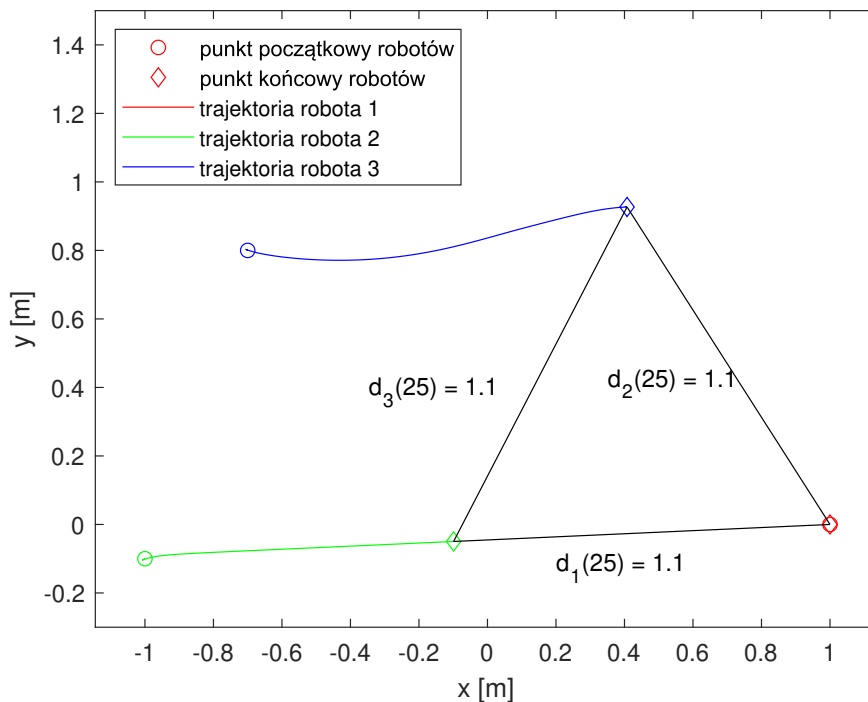
gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, a d_i są zadanymi odległościami. Precyzyjną definicję sąsiedztwa i zadaných odległości przedstawiono za pomocą grafu 5.32. W przykładzie przyjęto, bez straty ogólności, $d_1 = d_2 = d_3 = 1,1$ m.

Dla warunków początkowych (5.97) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.33

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} 1,0 & -1,0 & -0,6 \\ 0,0 & -0,1 & 1,1 \\ 0 & \frac{\pi}{6} & -\frac{\pi}{12} \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \tag{5.97}$$

Na wykresie 5.33 zaznaczono początkową pozycję robotów (koło), pozycję końcową (diament) oraz trajektorie robotów 2 i 3. Pozycja początkowa i końcowa lidera pokrywa się (robot pozostaje w bezruchu). Końcowa odległość pomiędzy robotami została oznaczona czarnymi odcinkami. Trajektorja robota 2 jest podobna do zachowania obserwowanego w przykładzie wprowadzającym. Wynika to ze sterowania, które pod uwagę bierze wyłącznie jednego sąsiada. Trajektorja robota 3 jest bardziej skomplikowana ponieważ celem robota jest utrzymanie zadanej odległości do dwóch sąsiadujących robotów. Jego początkowa orientacja i wzajemne położenie powodują zbliżenie się robota do lidera w pierwszej fazie eksperymentu. W drugiej fazie eks-

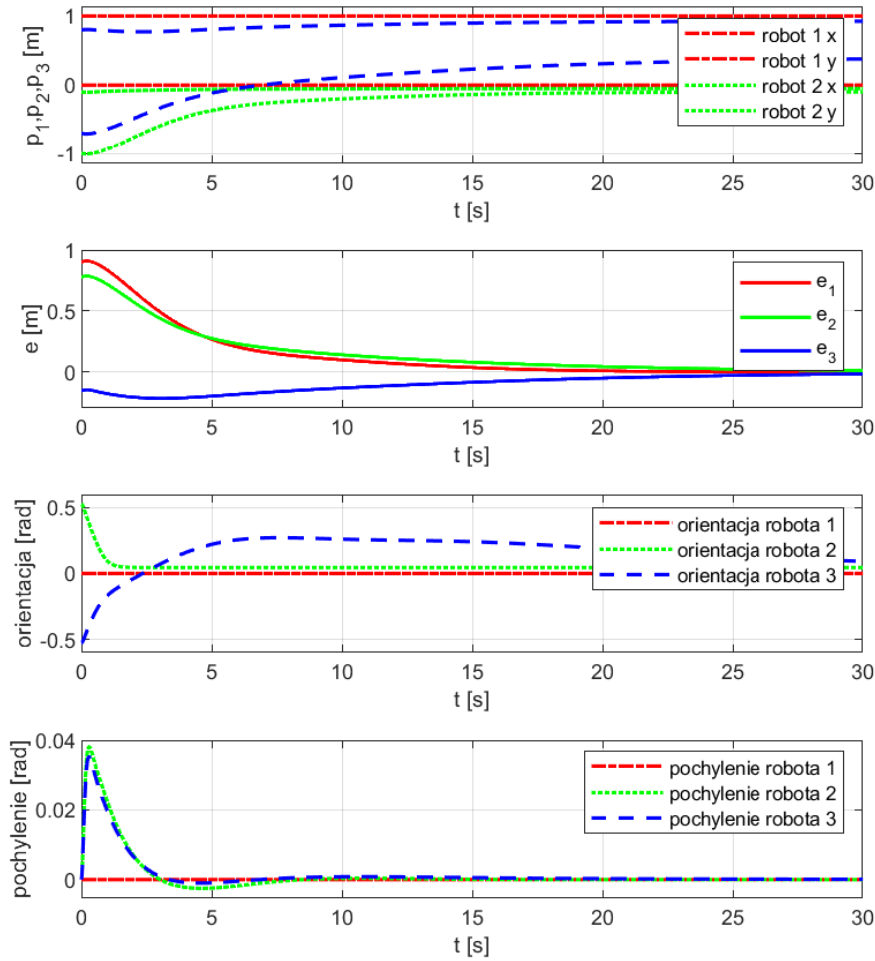
perymentu robot koryguje swoje położenie ze względu na zmianę położenia robota drugiego.



Rysunek 5.33: Formacja robotów w algorytmie śledzenia lidera, przypadek nieruchomego lidera

Na wykresie 5.34 przedstawiono przebieg położenia robotów, uchyb wzajemnej odległości (czyli wartości $e_1 = \|p_1 - p_2\| - d_1$), itd.), orientację oraz pochylenie robotów.

Przebieg czasowy pokazuje, że po około 5 sekundach robot 2 osiąga zadaną odległość (uchyb e_1 znajduje się w pobliżu 0). W podobnym czasie robot 3 zbliża się do lidera (uchyb e_2 znajduje się w pobliżu 0). Początkowa orientacja robota 3 oraz postać sterowania powoduje wzrost uchybu e_3 w pierwszej fazie eksperymentu. Jednak od 5 sekundy robot 3 oddala się od robota 2 i uchyb odległości osiąga wartość bliską 0 w 25 sekundzie eksperymentu. Wymaga to znacznej zmiany orientacji robota trzeciego, która jest widoczna na wykresie. Zmiana orientacji oraz położenia robotów powodują niewielkie pochylenia robotów od postawy pionowej, wynoszące około 3° , które można zaobserwować na czwartym wykresie.

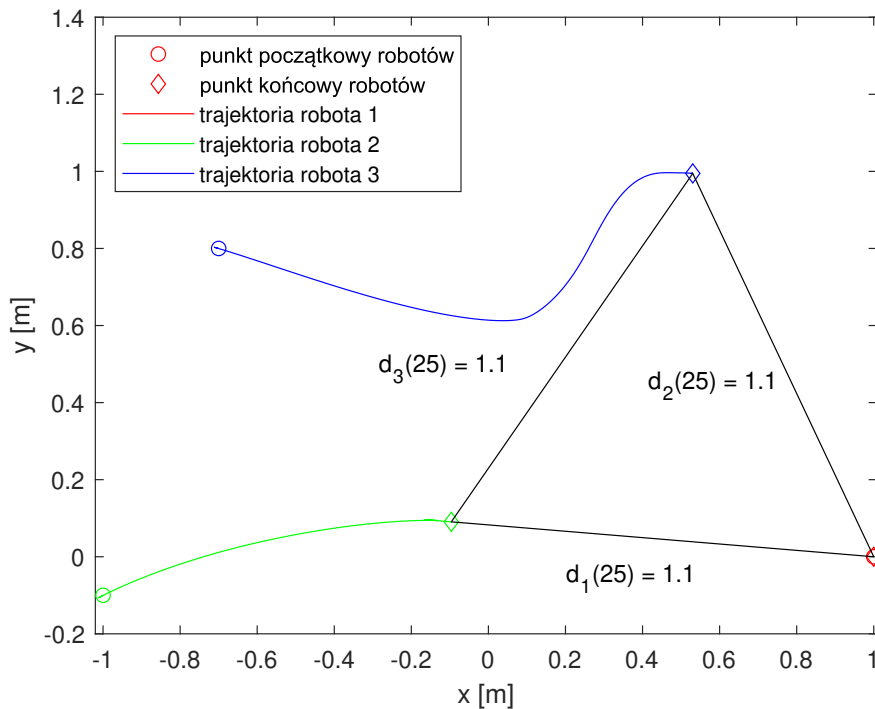


Rysunek 5.34: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Wyniki potwierdzają, że proponowany algorytm śledzenia lidera dla formacji trójkątnej działa poprawnie, gdy lider pozostaje w spoczynku. Roboty śledzące osiągną zadane odległości. Eksperymenty opisane w dwóch kolejnych sekcjach dotyczą sytuacji, w której lider porusza się wzdłuż zadanej trajektorii.

Przykład śledzenie lidera dobrze obrazuje poprawę jakości sterowania poprzez normalizację wektora kierunku. Na wykresach 5.35 i 5.36 przedstawiono wyniki dla takich samych warunków jak w poprzednim eksperymencie, ale przy zastosowaniu sterowania:

$$\begin{aligned}
 u_1 &= 0 \\
 u_2 &= -(p_2 - p_1)(\|p_2 - p_1\|^2 - d_1^2) \\
 u_3 &= -(p_3 - p_1)(\|p_3 - p_1\|^2 - d_2^2) - (p_3 - p_2)(\|p_3 - p_2\|^2 - d_3^2)
 \end{aligned} \tag{5.98}$$



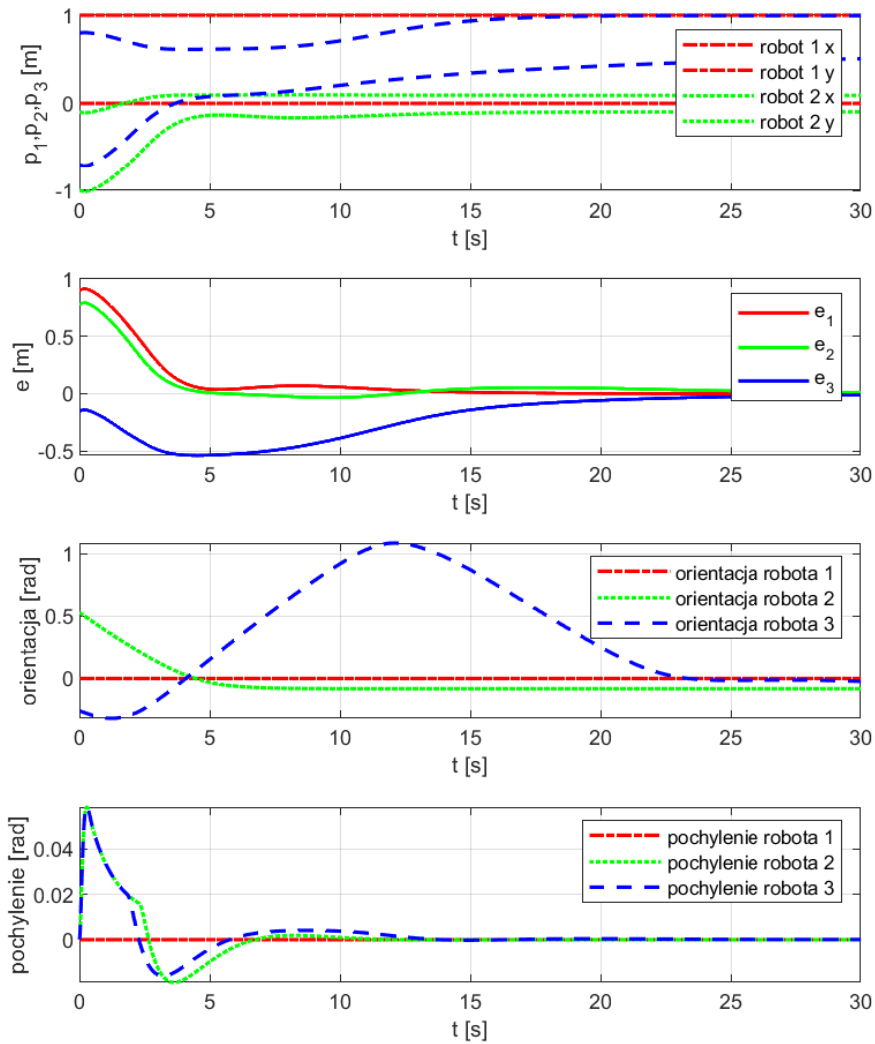
Rysunek 5.35: Formacja robotów w algorytmie śledzenia lidera, przypadek nieruchomego lidera

Modyfikacja sterowania (5.98) względem zastosowanego w opisywanym eksperymencie (5.96) polega na braku normalizacji wektora wskazującego kierunek do sąsiada. Sterowania tej postaci stosowano w przykładach przedstawionych w rozdziale 4.

Szczególne różnice jest widoczne w zachowaniu robota 3. Brak normalizacji wektora kierunku negatywnie wpływa na kontrolę odległości do robota 1 i 2. Ze względu na większą względną pozycję robotów 3 i 1 w stosunku do 3 i 2 śledzenie lidera dominuje sterowanie powodując nadmierne zbliżenie robota 3 do robota 2.

Na przebiegach prezentowanych na wykresie 5.36 można zaobserwować dwukrotnie większy uchyb zadanej odległości e_3 . W konsekwencji nadmiernego zbliżenia do robota 2, robot 3 musi wykonać manewr znacznie zmieniający jego orientację w trakcie trwania eksperymentu.

Mimo gorszej jakości sterowania formacja zostaje osiągnięta. Jest to zgodne z analizami teoretycznymi.

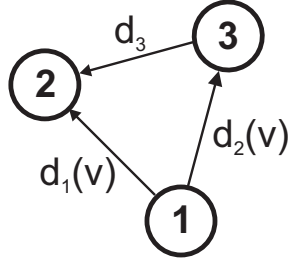


Rysunek 5.36: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Nagranie z przebiegu eksperymentu zostało zamieszczone pod adresem https://youtu.be/F_xm6Fnh8x4.

5.3.5 Algorytm śledzenia lidera poruszającego się wzdłuż prostej, przypadek niestacjonarnych, zadanych odległości

Celem eksperymentu jest analiza algorytmu śledzenia lidera w przypadku ruchu lidera. Ponadto dopuszcza się sytuację, w której odległości pomiędzy robotami są zmienne w czasie. Analizowany jest przypadek, w którym zadane odległości pomiędzy liderem, a robotami 2 i 3 są zależne od prędkości ruchu lidera. Pożądana odległość pomiędzy robotami 2 i 3 pozostaje stała podczas ruchu robotów (rysunek 5.37).



Rysunek 5.37: Digraf opisujący wymianę informacji w przypadku trzech robotów, z których pierwszy jest liderem, drugi podąża za pierwszym a trzeci za pierwszymi dwoma, a odległości pomiędzy robotami są zależne od prędkości

Powyższe założenie wymagają dostosowania algorytmu sterującego formacją robotów. Dla uproszczenia zakłada się, że zadany profil prędkości jest znany wszystkim robotom. Zmodyfikowane sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za ruch wzdłuż zadanej trajektorii oraz doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned}
 u_1 &= v \\
 u_2 &= -k_p \frac{p_2 - p_1}{\|p_2 - p_1\|} (\|p_2 - p_1\|^2 - d_1^2(v)) + k_v v \\
 u_3 &= -k_p \left(\frac{p_3 - p_1}{\|p_3 - p_1\|} (\|p_3 - p_1\|^2 - d_2^2(v)) + \right. \\
 &\quad \left. + \frac{p_3 - p_2}{\|p_3 - p_2\|} (\|p_3 - p_2\|^2 - d_3^2) \right) + k_v v
 \end{aligned} \tag{5.99}$$

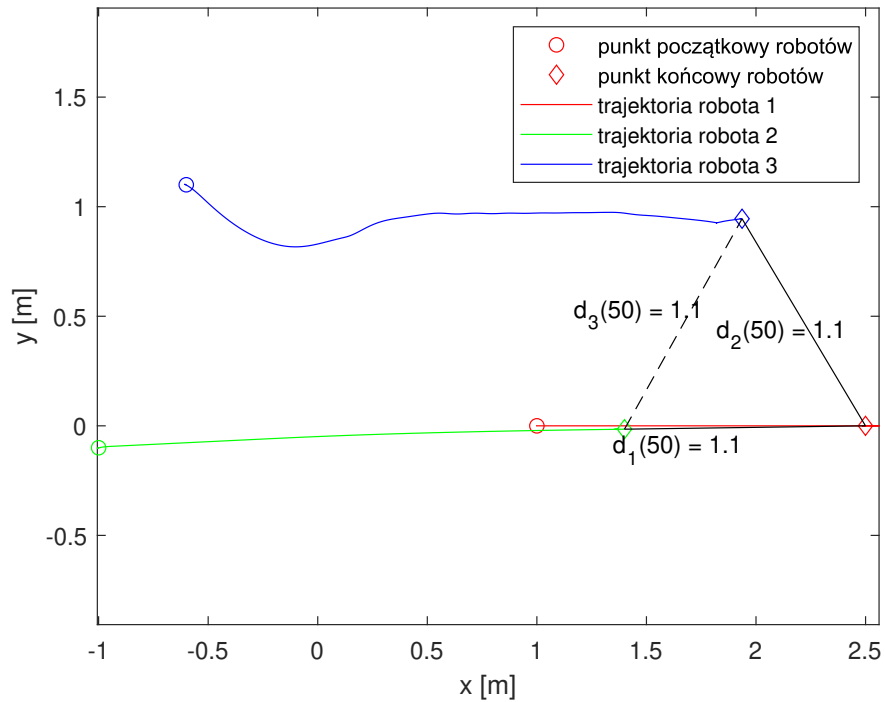
gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, v jest zadany profilem prędkości, d_i są zadanymi odległościami, a k_p, k_v parametrami regulatora. Precyzyjną definicję sąsiedztwa i zadanych odległości przedstawiono za pomocą grafu 5.37. W przykładzie przyjęto, bez straty ogólności, $d_{10} = d_{20} = d_3 = 1,1$ m, $d_1(v) = d_{10} + v$ i $d_2(v) = d_{20} + v$ oraz $k_p = 1$, $k_v = 0,8$.

Dla warunków początkowych (5.100) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.38.

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} 1,0 & -1,0 & -0,6 \\ 0,0 & -0,1 & 1,1 \\ 0 & \frac{\pi}{6} & -\frac{\pi}{12} \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \tag{5.100}$$

Na wykresie 5.38 zaznaczono początkową pozycje robotów (koło), pozycje końcową (diament) oraz trajektorie robotów 1, 2 i 3. Końcowa odległość pomiędzy robotami została oznaczona czarnymi odcinkami. Przerwaną linią oznaczono odległość,

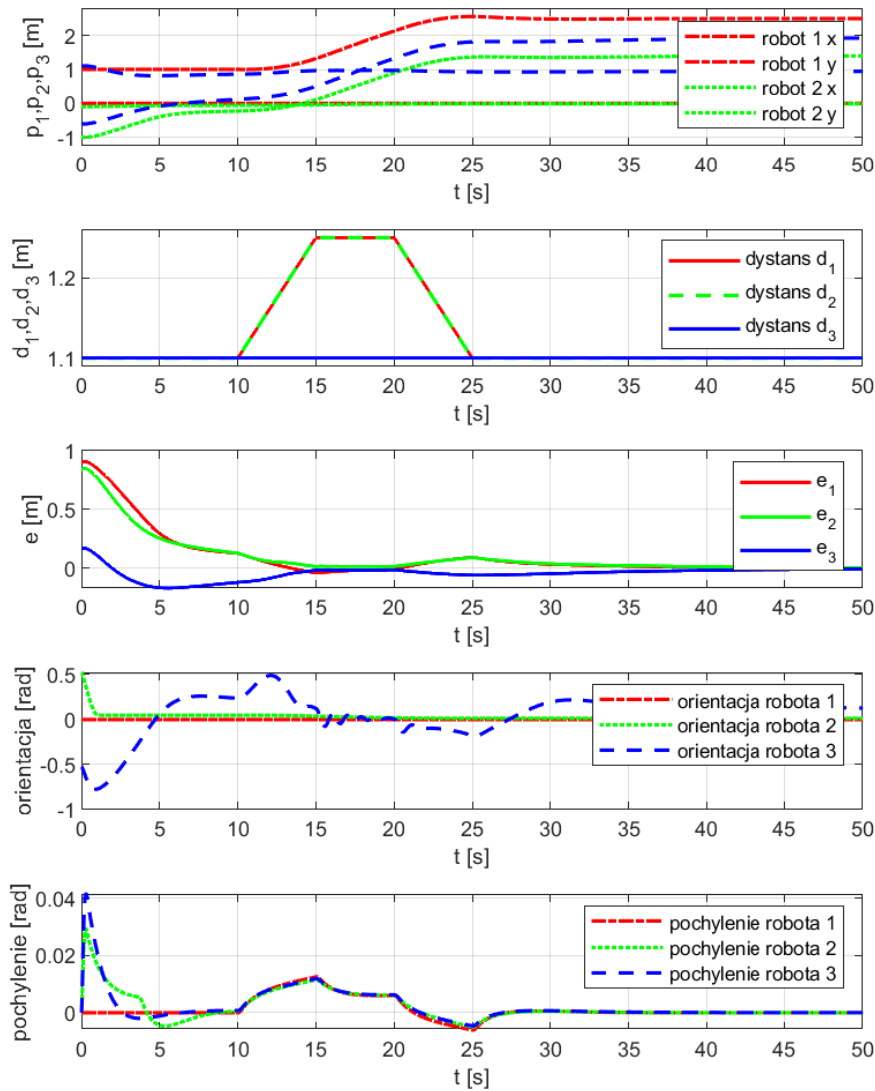
dla której referencja jest stała, linią ciągłą oznaczono odległość, dla której zadana wartość jest zmienna w czasie. W pierwszej fazie eksperymentu (lider w bezruchu) trajektorie robotów 2 i 3 są podobna do zachowania obserwowanego w poprzednich przykładach. W drugiej fazie eksperymentu lider rozpoczyna ruch zgodnie z zadanym sygnałem referencyjnym. Roboty 2 i 3 podążają za liderem utrzymując zadaną formację.



Rysunek 5.38: Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej

Na wykresie 5.39 przedstawiono przebieg położenia robotów, zadane odległości, uchyb wzajemnej odległości (czyli wartości $e_1 = \|p_1 - p_2\| - d_1(v)$, itd.), orientację oraz pochylenie robotów.

Istotną modyfikacją względem poprzednich eksperymentów są zmieniające się w czasie wybrane odległości pomiędzy robotami.



Rysunek 5.39: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Przebieg czasowy zadanej odległości pokazuje, że pomiędzy 10 a 25 sekundą następuje zmiana tego sygnału wynikająca z zadanego profilu prędkości (jest on jednakowego kształtu). Przez pierwsze 10 sekund eksperymentu roboty 2 i 3 wykonują ruch w celu osiągnięcia zadanej formacji. W 10 sekundzie znajdują się one około 20 cm od celu ale w tej chwili lider rozpoczyna ruch. Pomimo tego algorytmy sterujące pozwalają uzyskać zadaną formację, która jest utrzymywana w trakcie trwania ruchu. W fazie ruchu lidera ze stałą prędkością uchyb jest bliski zero. W fazie wytracania prędkości przez lidera (od 20 do 25 sekundy) uchyb wzajemnych odległości wzrasta, po czym osiąga powtórnie wartości bliskie zero w ostatniej fazie eksperymentu. Zmiana orientacji (wykres 4) oraz ruch robotów powodują niewielkie pochylenia od postawy pionowej, wynoszące około 3° , które można zaobserwować na piątym wykresie.

Wyniki potwierdzają, że proponowany algorytm śledzenia lidera poruszającego się wzdłuż linii działa poprawnie. Rozszerzeniem wcześniejszych rozważań jest do-

puszczenie sytuacji zmiennych w czasie zadanych odległości pomiędzy robotami. W analizowanym przypadku dwie z trzech odległości były zależne od prędkości robotów. Modyfikacja algorytmu sterującego formacją, polegająca na wykorzystaniu informacji o prędkości, pozwoliła poprawnie zrealizować cel zadania.

Nagranie z przebiegu eksperymentu zostało zamieszczone pod adresem <https://youtu.be/gXFVxIyTC3g>.

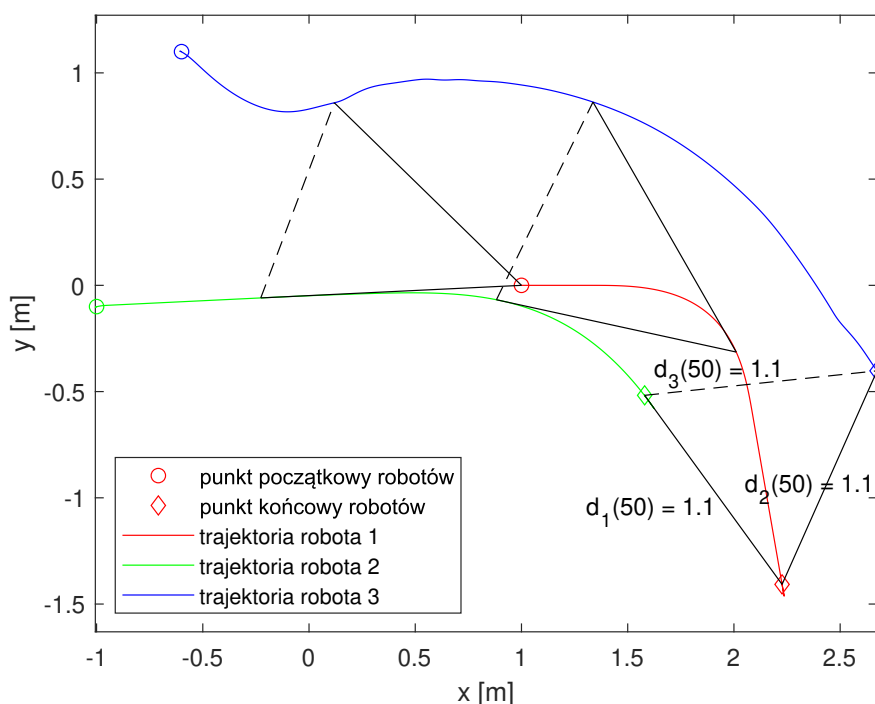
5.3.6 Algorytm śledzenia ruchomego lidera poruszającego się wzdłuż łuku, przypadek niestacjonarnych, zadanych odległości

Eksperyment wykorzystuje założenia opisane w poprzednim przykładzie. Algorytm sterujący formacją robotów jest opisany przez równania (5.99). Zadaniem lidera jest ruch po łuku. Podobnie jak w poprzednim przypadku, odległości d_1 i d_2 są zależne od prędkości robotów.

Dla warunków początkowych (5.101) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.40.

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} 1,0 & -1,0 & -0,6 \\ 0,0 & -0,1 & 1,1 \\ 0 & \frac{\pi}{6} & -\frac{\pi}{12} \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \quad (5.101)$$

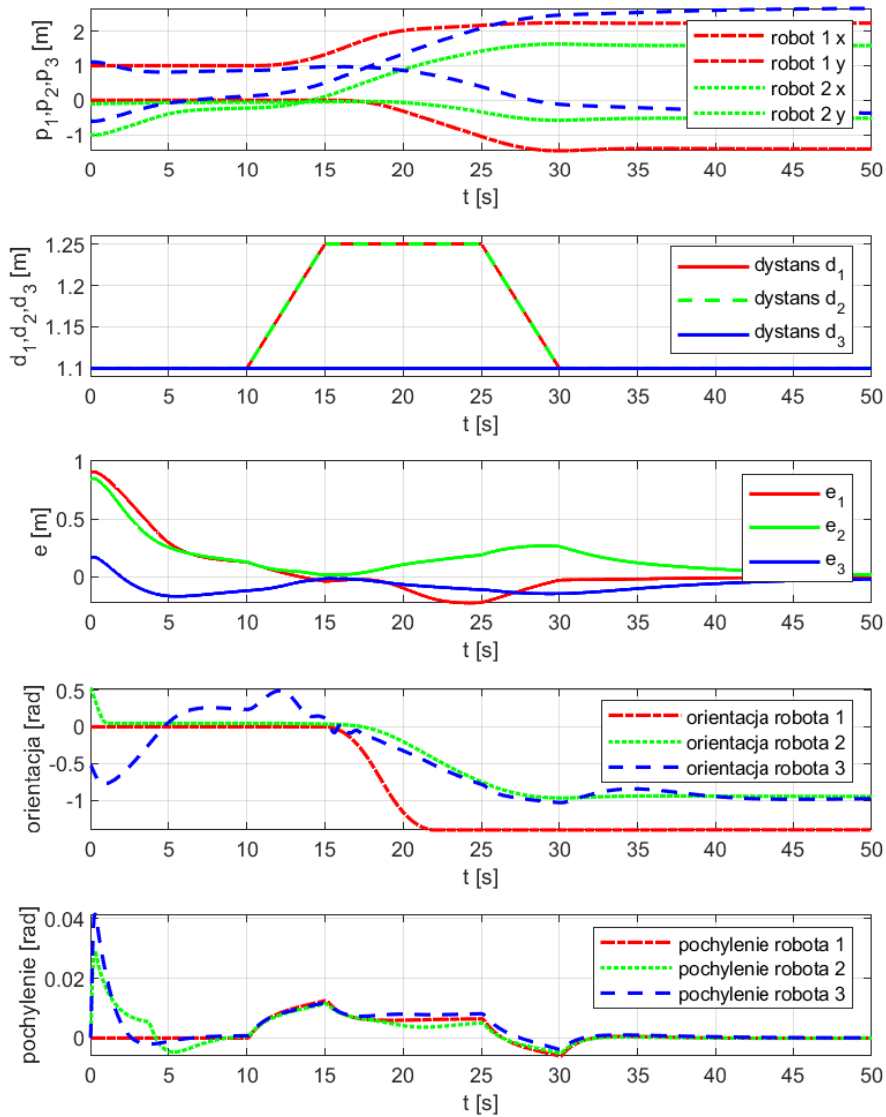
Na wykresie 5.40 zaznaczono początkową pozycje robotów (koło), pozycje końcową (diament) oraz trajektorie robotów 1, 2 i 3. Odległość pomiędzy robotami, w różnych chwilach eksperymentu, została oznaczona czarnymi odcinkami. Przerwaną linią oznaczono odległość, dla której referencja jest stała, linią ciągłą oznaczono odległość, dla której zadana wartość jest zmienna w czasie. Mimo bardziej skomplikowanej trajektorii lidera przebieg eksperymentu jest podobny jak w poprzednim przypadku. W pierwszej fazie eksperymentu roboty 2 i 3 zbliżają się do lidera na zadaną odległość, jednocześnie robot 3 utrzymuje zadaną odległość od robota 2. W drugiej fazie eksperymentu lider rozpoczyna ruch zgodnie z zadanym sygnałem referencyjnym. Roboty 2 i 3 podążają za liderem utrzymując zadaną formację.



Rysunek 5.40: Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej

Na wykresie 5.41 przedstawiono przebieg położenia robotów, zadane odległości, uchyb wzajemnej odległości (czyli wartości $e_1 = \|p_1 - p_2\| - d_1(v)$), itd.), orientację oraz pochylenie robotów.

Przebieg czasowy zadanej odległości pokazuje, że pomiędzy 10 a 25 sekundą następuje zmiana tego sygnału wynikająca z zadanego profilu prędkości (jest on jednakowego kształtu). Przez pierwsze 10 sekund eksperymentu roboty 2 i 3 wykonują ruch w celu osiągnięcia zadanej formacji. W 10 sekundzie znajdują się one około 20 cm od celu ale w tej chwili lider rozpoczyna ruch. Pomimo tego algorytmy sterujące pozwalają uzyskać zadaną formację, która jest utrzymywana w trakcie trwania ruchu. W fazie ruchu lidera ze stałą prędkością uchyb jest bliski zeru. Po 15 sekundach lider zmienia swoją orientację zgodnie z zadanym sygnałem referencyjnym. Roboty 2 i 3 nadążają za zmianą orientacji, ale nie osiągają tej samej orientacji co lider. Wpływa to na wzrost uchybu wzajemnych odległości. W fazie wytracania prędkości przez lidera (od 25 do 30 sekundy) uchyb wzajemnych odległości wzrasta, po czym osiąga powtórnie wartości bliskie zeru w ostatniej fazie eksperymentu. Zmiana orientacji (wykres 4) oraz ruch robotów powodują niewielkie pochylenia od postawy pionowej, wynoszące około 3° , które można zaobserwować na piątym wykresie.



Rysunek 5.41: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Wyniki potwierdzają, że proponowany algorytm śledzenia lidera poruszającego się wzdłuż łuku działa poprawnie. Modyfikacja algorytmu sterującego formacją, polegająca na wykorzystaniu informacji o prędkości, pozwoliła poprawnie zrealizować cel zadania.

Nagranie z przebiegu eksperymentu zostało zamieszczone pod adresem <https://youtu.be/cY9ZHyeEnHo>.

5.3.7 Algorytm sterowania formacją dla pięciu robotów

Celem eksperymentu jest zbadanie algorytmu sterowania formacją dla większej liczby niestabilnych robotów oraz analiza wpływu warunków początkowych na osiągnięty efekt. Zadaniem robotów jest utworzenie formacji pięciokąta foremnego. Zakłada się, że sąsiedztwo pomiędzy robotami opisuje graf przedstawiony na rysunku

4.18. Struktura wymiany informacji jest taka sama jak w eksperymencie opisywanym w sekcji 4.4.5.

Dla opisywanego przypadku macierz sąsiedztwa $A(\mathcal{G})$ i macierz wag mają postać:

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad D(\mathcal{G}) = \begin{bmatrix} 0 & \eta & \gamma & 0 & \eta \\ \eta & 0 & \eta & \gamma & \gamma \\ \gamma & \eta & 0 & \eta & 0 \\ 0 & \gamma & \eta & 0 & \eta \\ \eta & \gamma & 0 & \eta & 0 \end{bmatrix}, \quad (5.102)$$

gdzie η oznacza bok pięciokąta, a γ jego przekątną i wynosi $\gamma = \frac{1+\sqrt{5}}{2}\eta$.

Sterowanie dla poszczególnych robotów, które pozwala osiągnąć zadaną formację ma postać

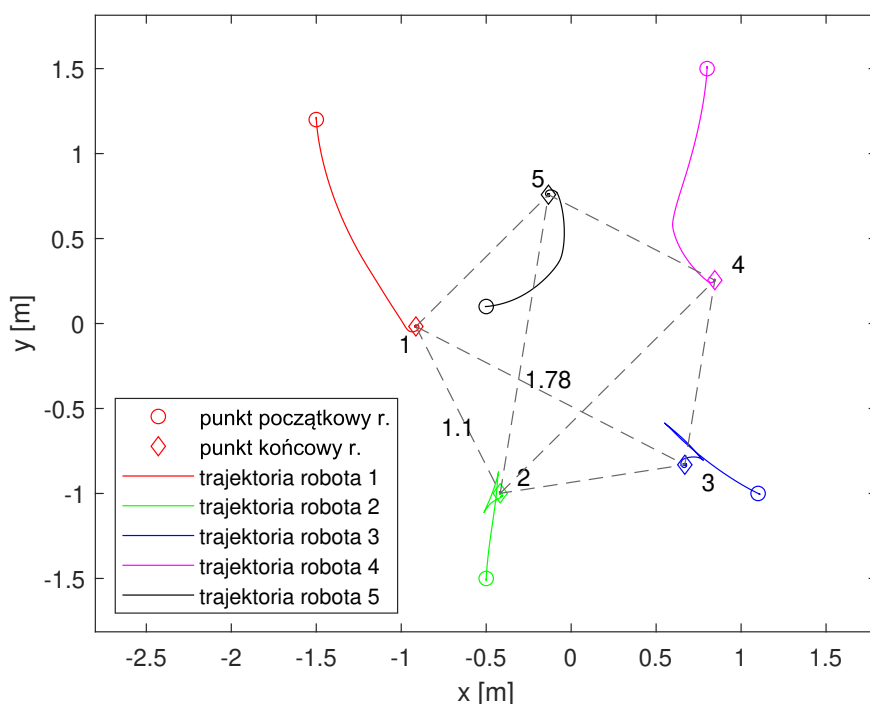
$$u_i = - \sum_{j=1, \dots, 5} a_{ij} (\|p_i - p_j\|^2 - d_{ij}^2) \frac{(p_i - p_j)}{\|(p_i - p_j)\|} \quad (5.103)$$

gdzie a_{ij} są elementami macierzy sąsiedztwa, a d_{ij} oznaczają zadane odległości pomiędzy parami pojazdów i, j (są elementami macierzy wag $D(\mathcal{G})$).

Przeprowadzono eksperymenty z dwoma zestawami warunków początkowych, aby zbadać ich wpływ na osiągniętą formację. Zgodnie z przewidywaniami opisanymi w podrozdziale 4.4.5 oczekuje się, że roboty osiągną formację o orientacji pozytywnej lub negatywnej w zależności od warunków początkowych. Zadana odległość (bok pięciokąta) wynosi $\eta = 1,1$ m.

Dla warunków początkowych (5.104) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.42.

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} -1,5 & -0,5 & 1,1 & 0,6 & -0,6 \\ 1,2 & -1,5 & -1,0 & 1,7 & 0,1 \\ \frac{\pi}{2} & -\frac{\pi}{2} & -\frac{\pi}{8} & -\frac{\pi}{2} & \pi \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \quad (5.104)$$

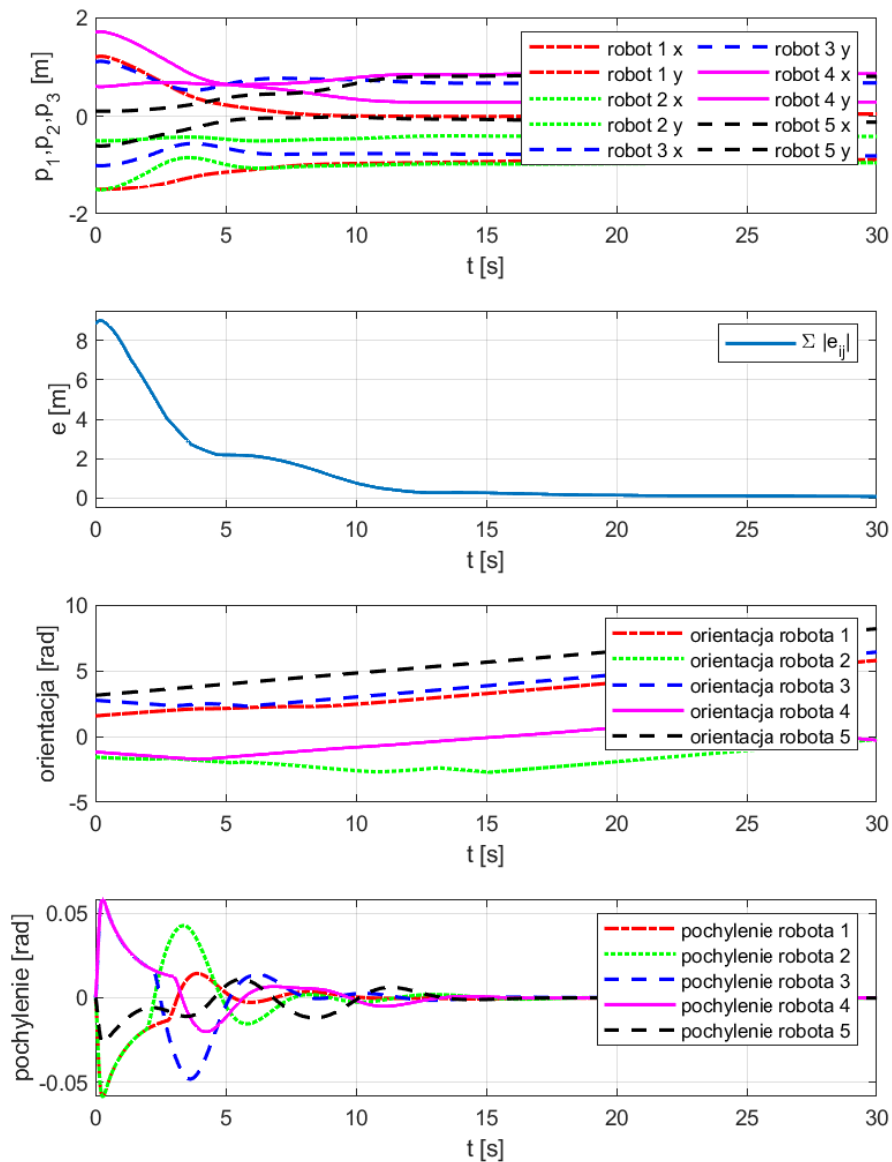


Rysunek 5.42: Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej

Na rysunku 5.42 zaznaczono osiągniętą formację oraz ruch pojazdów na płaszczyźnie. Numery przy robotach odpowiadają wierzchołkom grafu. Koła w odpowiednim kolorze oznaczają początkową pozycję robotów, pozycje końcową oznacza symbol diamentu. Trajektorie robotów oznaczono kolorami (w kolejności): czerwonym, zielonym, niebieskim, różowym oraz czarnym. Końcowe odległości pomiędzy robotami, które podlegają sterowaniu, zostały oznaczone szarymi odcinkami.

Dla warunków początkowych (5.104) roboty osiągają pozytywnie zorientowaną formację.

Na wykresie 5.43 przedstawiono przebieg położenia robotów, zadane odległości, sumę wszystkich wartości bezwzględnych uchybów wzajemnej odległości, orientację oraz pochylenie robotów.



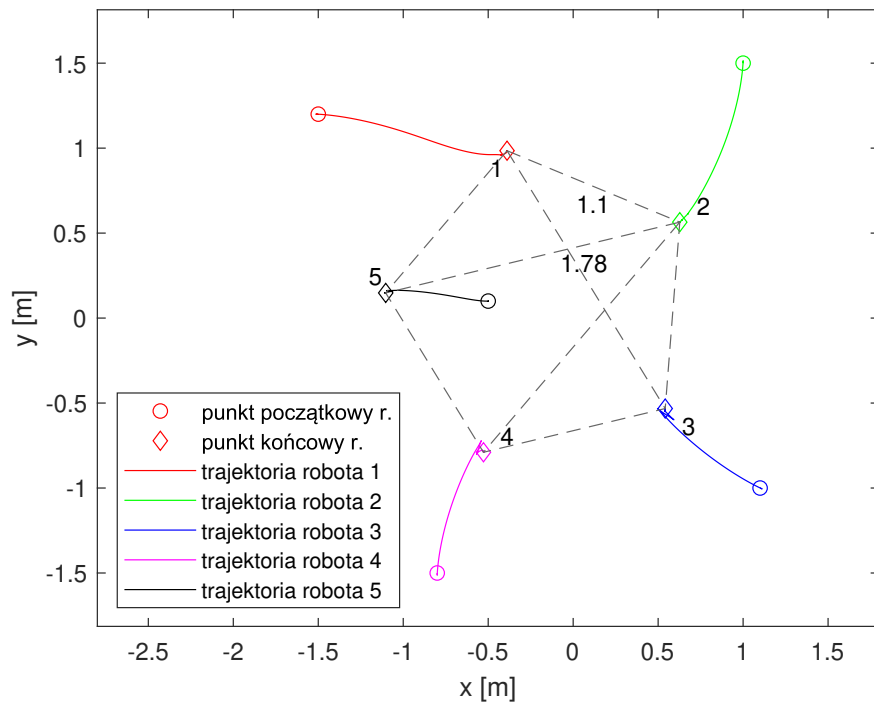
Rysunek 5.43: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Dla warunków początkowych (5.105) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.44.

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} -1,5 & 1,0 & 1,1 & -0,8 & -0,5 \\ 1,2 & 1,5 & -1,0 & -1,5 & 0,1 \\ \pi & -\frac{\pi}{2} & -\frac{\pi}{8} & -\frac{\pi}{2} & \pi \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \quad (5.105)$$

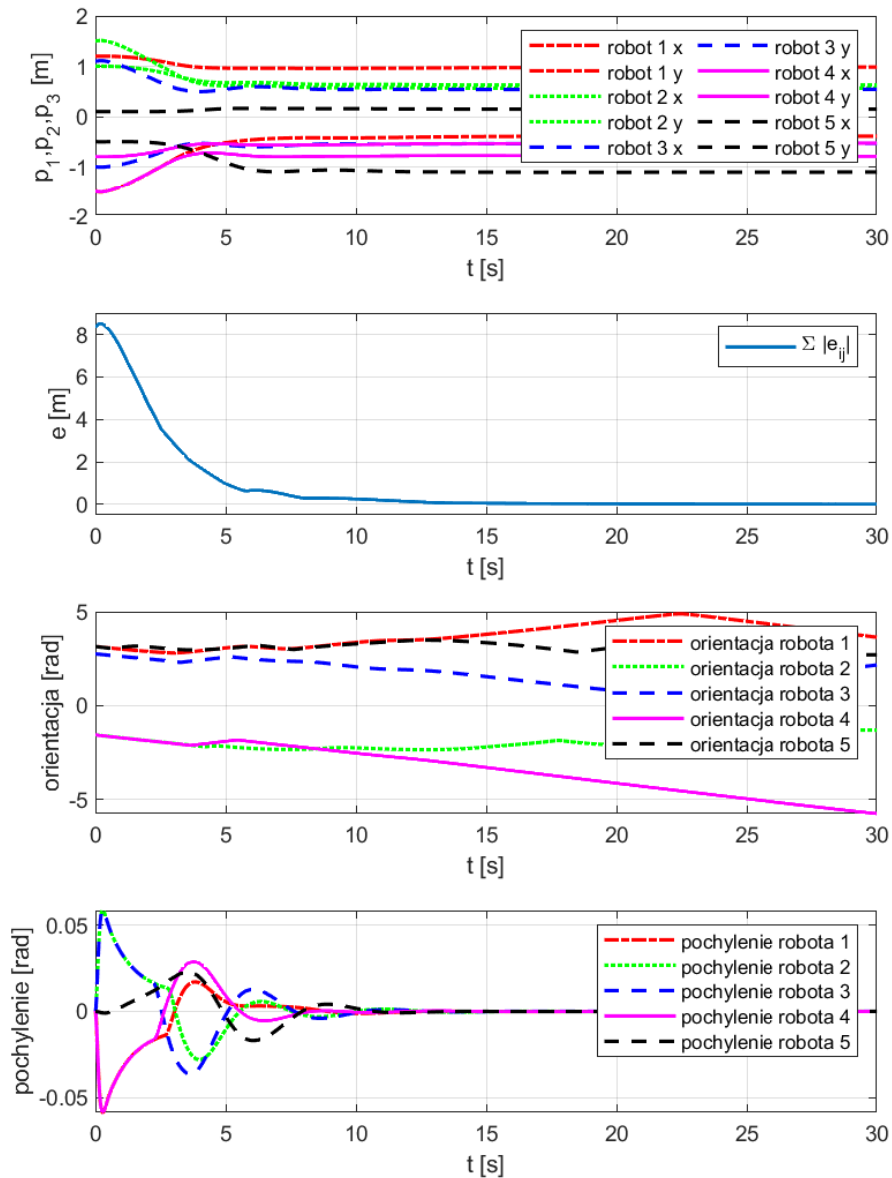
Na rysunku 5.44 zaznaczono osiągniętą formację oraz ruch pojazdów na płaszczyźnie. Numery przy robotach odpowiadają wierzchołkom grafu.

Dla warunków początkowych (5.105) roboty osiągają negatywnie zorientowaną formację.



Rysunek 5.44: Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej

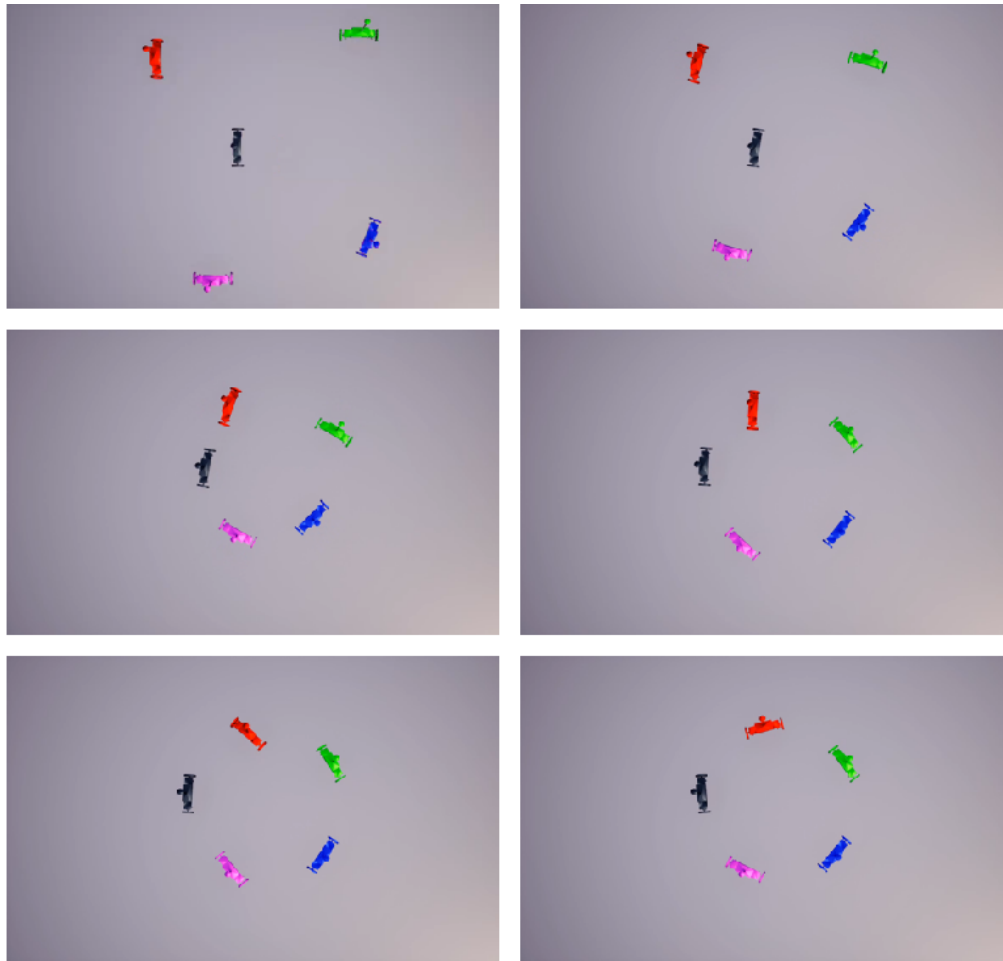
Na wykresie 5.45 przedstawiono przebieg położenia robotów, zadane odległości, sumę wszystkich uchybów wzajemnej odległości, orientację oraz pochylenie robotów.



Rysunek 5.45: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Na rysunku 5.46 przedstawiono migawki z wizualizacji eksperymentu w początkowej fazie, a następnie w pierwszej, drugiej, czwartej, siódmej i dziesiątej sekundzie eksperymentu.

Nagrania z przebiegu eksperymentu zostało zamieszczone pod adresami <https://youtu.be/9yAtdyo7REA> i <https://youtu.be/J0dUaZGwjIY>. Dodatkowo pod adresem <https://youtu.be/tzyoy1XhYTE> został przedstawiony wynik eksperymentu dla sterowania, w którym nie są normalizowane wektory kierunków.



Rysunek 5.46: Migawki z wizualizacji eksperymentu symulacyjnego dla początku eksperymentu, pierwszej, drugiej, czwartej, siódmej i dziesiątej sekundy

5.3.8 Podsumowanie wyników

Przeprowadzone badania symulacyjne z wykorzystaniem modelu matematycznego opisującego dynamikę dwukołowego, niestabilnego pojazdu potwierdziły poprawność układu regulacji oraz algorytmów sterujących zespołem robotów.

Proponowany układ składa się z wewnętrznej pętli sprzężenia zwrotnego, funkcji odwzorowującej sterowanie dla nieholonomicznego robota oraz zewnętrznej pętli do sterowania formacją robotów. Wewnętrzna pętla jest oparta na regulatorze od stanu (LQR), którego zadaniem jest utrzymanie pionowej postawy robota oraz nadążanie za sygnałem referencyjnym dla prędkości liniowej i kątowej. Zewnętrzna pętla sprzężenia zwrotnego jest lokalnym sterowaniem, obliczonym przez każdego robota na podstawie symulowanych lokalnych pomiarów. W zewnętrznej pętli zastosowano wybór algorytmów podobnych do prezentowanych w rozdziałach 3 oraz 4. Użycie tych prostych algorytmów jest możliwe dzięki zastosowaniu funkcji odwzorowującej sterowanie obliczone dla punktu kinematycznego na kinematykę nieholonomicznego robota.

5.4 Eksperymenty z pojazdem fizycznym

5.4.1 Metodologia badań na rzeczywistym robocie

Badania eksperymentalne zostały przeprowadzone przy pomocy rzeczywistego obiektu opisanego w sekcji 5.1. Ze względu na dostępność jednego robota, przetestowano algorytm śledzenia lidera. Sposób wymiany informacji pomiędzy sąsiadami w tym algorytmie umożliwia wykonanie trzech sekwencyjnych eksperymentów z jednym robotem, zamiast jednego eksperymentu z trzema robotami.

Przyjęto następujący scenariusz testowy:

- Robot zostaje umieszczony w pozycji początkowej $p_1(0)$. Zostaje uruchomiony program sterujący dla lidera, który utrzymuje zadaną postawę, a ponadto umożliwia podążanie za referencyjną trajektorią. Dane z położenia robota w trakcie trwania eksperymentu zostają zapisane.
- W kolejnym eksperymencie ten sam robot zostaje umieszczony w pozycji początkowej $p_2(0)$. W tym eksperymencie zostaje uruchomiony program sterujący dla pierwszego podążającego robota (na grafie oznaczony jako wierzchołek 2). Program wykorzystuje informacje zalogowane w poprzednim eksperymencie informujące o położeniu lidera. Na tej podstawie zewnętrzna pętla sprzężenia zwrotnego oblicza sterowanie dla podążającego robota. Obliczenia te symulują posiadanie przez robota lokalnego czujnika, mierzącego odległość i kierunek do sąsiada (przykładem takiego czujnika jest lidar). Wewnętrzna pętla sprzężenia zwrotnego utrzymuje pionową postawę oraz pozwala na nadążanie za sygnałem obliczonym przez zewnętrzną pętlę sprzężenia zwrotnego. Dane z położenia robota w trakcie trwania eksperymentu zostają zapisane.
- W ostatnim eksperymencie robot zostaje umieszczony w pozycji $p_3(0)$. Uruchomiony zostaje algorytm dla drugiego podążającego robota. Zgodnie ze sposobem wymiany informacji w tym algorytmie, dla tego robota dostępne są informacje o położeniu dwóch sąsiadów (lidera i pierwszego podążającego). Na tej podstawie zewnętrzna pętla sprzężenia zwrotnego oblicza sterowanie utrzymujące formację. Podobnie jak poprzednio wewnętrzna pętla sprzężenia zwrotnego utrzymuje pionową postawę oraz pozwala na nadążanie za sygnałem obliczonym przez zewnętrzną pętlę sprzężenia zwrotnego. Dane są logowane w celu dalszej analizy.

W eksperymencie pętle sterujące oraz logowanie danych odbywa się z częstotliwością 100 Hz. Algorytmy są zapisane w postaci schematów blokowych programu Simulink, a do ich uruchomienia na platformie sprzętowo-programowej wykorzystywana

jest technologia automatycznej generacji kodu (opisana w sekcji 5.1.2). Komputer PC, który służy do opracowania algorytmów pozwala również na monitorowanie parametrów prac robota podczas uruchomienia algorytmów.

W eksperymentach zakłada się, że poślizg kół nie występuje. Położenie robotów jest odtwarzane z pomiarów położenia kół (odometria). Algorytm utrzymujący pionową postawę robota, ze względów bezpieczeństwa, zostaje uruchomiony, gdy odchylenie robota jest nie większe niż 10° . W początkowej fazie eksperymentu robot spoczywa na podporze i musi zostać doprowadzony do pozycji pionowej manualnie.

5.4.2 Algorytm śledzenia stacjonarnego lidera

Eksperyment polega na badaniu algorytmu formacji trzech robotów mobilnych w przypadku nierównorzędnych ról oraz wpływu rzeczywistych warunków na jakość sterowania. Wymiana informacji w analizowanym zagadnieniu jest przedstawiona za pomocą grafu na rysunku 5.32, oraz opisana w sekcji 5.3.4.

Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za doprowadzenie i utrzymanie stałej odległości pomiędzy robotami ma postać:

$$\begin{aligned}
 u_1 &= 0 \\
 u_2 &= -\frac{p_2 - p_1}{\|p_2 - p_1\|} (\|p_2 - p_1\|^2 - d_1^2) \\
 u_3 &= -\frac{p_3 - p_1}{\|p_3 - p_1\|} (\|p_3 - p_1\|^2 - d_2^2) + \\
 &\quad -\frac{p_3 - p_2}{\|p_3 - p_2\|} (\|p_3 - p_2\|^2 - d_3^2)
 \end{aligned} \tag{5.106}$$

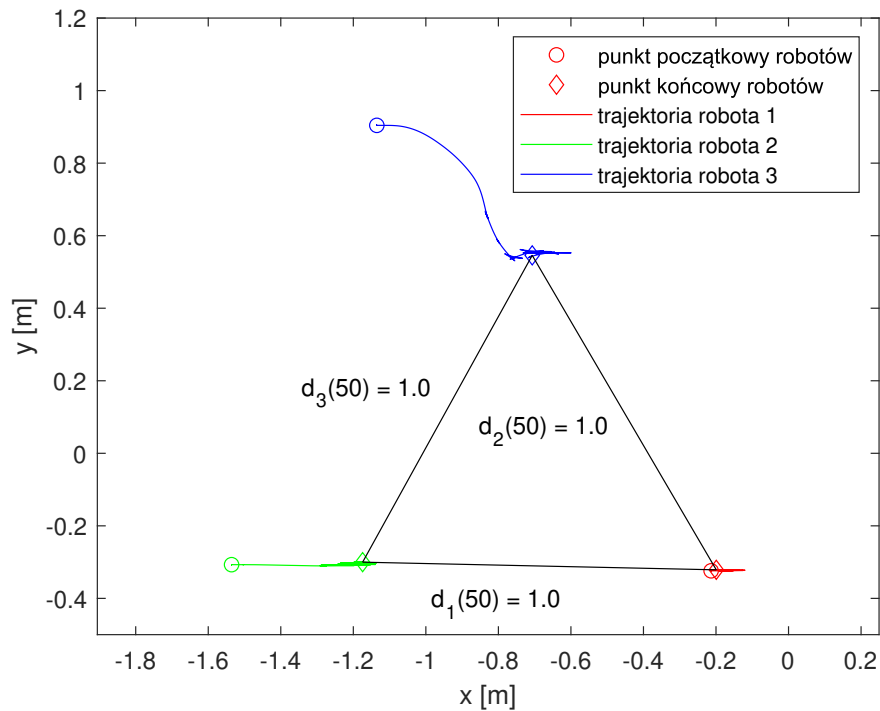
gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, a d_i są zadanymi odległościami. W przykładzie przyjęto, bez straty ogólności, $d_1 = d_2 = d_3 = 1$ m.

Dla warunków początkowych (5.107) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.47

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} -0,2 & -1,5 & -1,1 \\ -0,3 & -0,3 & 0,9 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \tag{5.107}$$

Na wykresie 5.47 zaznaczono początkowe pozycje robotów (koło), pozycje końcowe (diament) oraz trajektorie robotów 2 i 3. Pozycja początkowa i końcowa lidera znajdują się w pobliżu (robot utrzymuje swoje położenie). Końcowa odległość po-

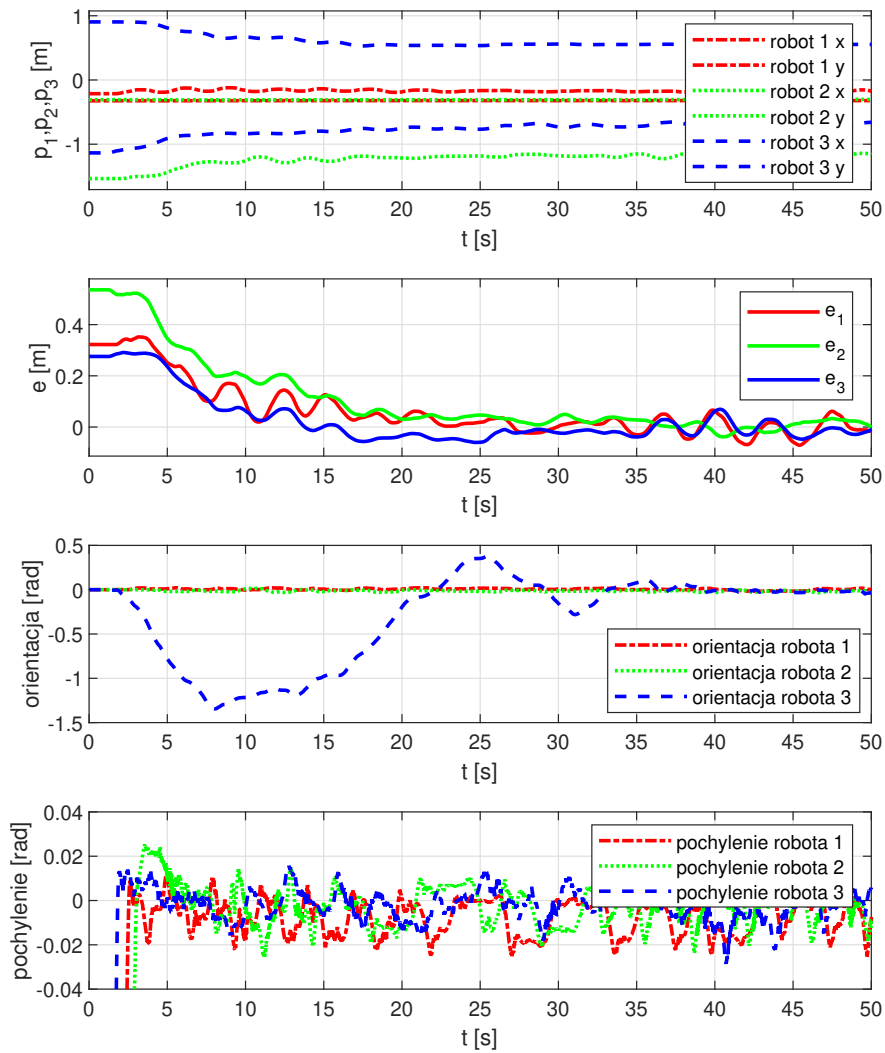
między robotami została oznaczona czarnymi odcinkami. Obserwowane trajektorie robotów mają podobny charakter do wyników otrzymanych w symulacjach. Istotną różnicą jest ciągły ruch robotów po osiągnięciu zadanych odległości wynikający z aktywnego sterowania utrzymującego pionową postawę. W idealizowanych warunkach symulacyjnych roboty osiągały punkt równowagi. W eksperymentach rzeczywistych pojazdy wykonują korygujące ruchy pozwalające zachować pionową postawę.



Rysunek 5.47: Formacja robotów w algorytmie śledzenia lidera, przypadek nieruchomego lidera

Na wykresie 5.48 przedstawiono przebieg położenia robotów, uchyb wzajemnej odległości (czyli wartości $e_1 = \|p_1 - p_2\| - d_1$), itd.), orientację oraz pochylenie robotów.

Przebieg czasowy pokazuje, że po około 25 sekundach robot 2 zbliża się na zadaną odległość (uchyb e_1 oscyluje w okolicach 0). W podobnym czasie robot 3 zbliża się do lidera (uchyb e_2 oscyluje w okolicach 0), a w okolicach 20 sekundy osiąga zadaną odległość od robota 2. Po osiągnięciu zadanych odległości uchyby oscylują w okolicach zera. Wynika to z ciągłych ruchów robota, które mają za zadanie utrzymać pionową postawę. Pochylenie wszystkich robotów zmienia się w czasie trwania eksperymentu. Amplituda tych oscylacji wynosi około $2,3^\circ$. Przebieg ten można zaobserwować na czwartym wykresie. Znaczna zmiana pochylenia robota widoczna w okolicach 3 sekundy wynika z manualnego doprowadzenia robota do pozycji początkowej.



Rysunek 5.48: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

Wyniki potwierdzają, że proponowany algorytm śledzenia lidera dla formacji trójkątnej działa poprawnie, gdy lider pozostaje w spoczynku. Roboty śledzące osiągną zadane odległości.

5.4.3 Algorytm śledzenia lidera poruszającego się wzdłuż prostej

Eksperyment polega na weryfikacji w rzeczywistych warunkach algorytmu śledzenia lidera w przypadku ruchu lidera wzdłuż zadanej prostoliniowej trajektorii.

Sterowanie w zewnętrznej pętli zwrotnej, odpowiedzialne za ruch wzdłuż zadanej trajektorii oraz doprowadzenie i utrzymanie stałej odległości pomiędzy robotami, ma

postać:

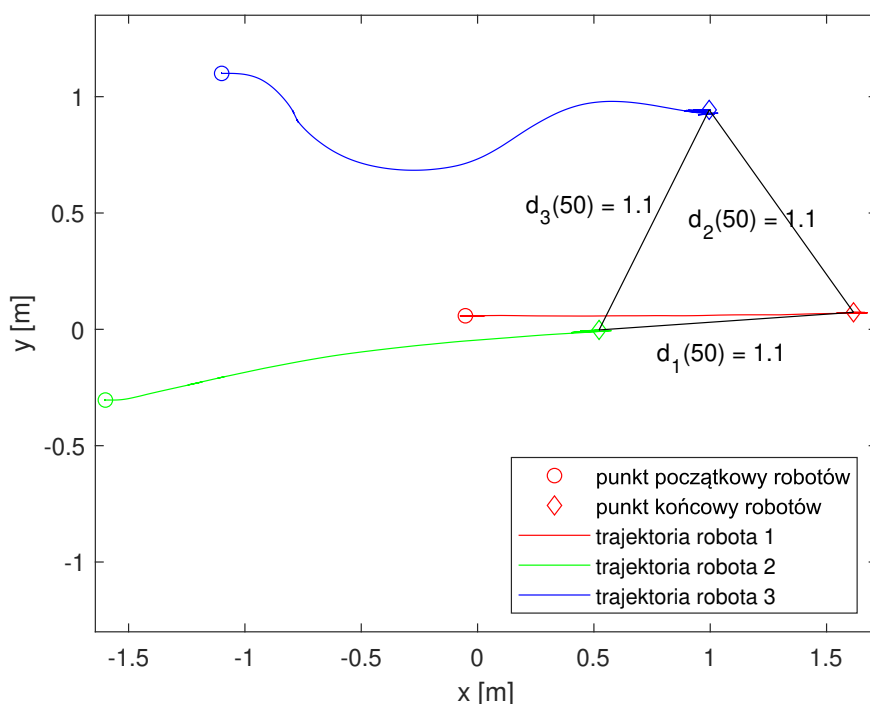
$$\begin{aligned}
u_1 &= v \\
u_2 &= -k_p \frac{p_2 - p_1}{\|p_2 - p_1\|} (\|p_2 - p_1\|^2 - d_1^2) + k_v v \\
u_3 &= -k_p \left(\frac{p_3 - p_1}{\|p_3 - p_1\|} (\|p_3 - p_1\|^2 - d_2^2) + \right. \\
&\quad \left. + \frac{p_3 - p_2}{\|p_3 - p_2\|} (\|p_3 - p_2\|^2 - d_3^2) \right) + k_v v
\end{aligned} \tag{5.108}$$

gdzie $u_i \in \mathbb{R}^2$, $i \in \{1, 2, 3\}$ są wektorami sterującymi, $p_i \in \mathbb{R}^2$ oznacza położenie robotów, v jest zadany profilem prędkości, d_i są zadanymi odległościami, a k_p , k_v parametrami regulatora. W przykładzie przyjęto, bez straty ogólności, $d_1 = d_2 = d_3 = 1,1$ m oraz $k_p = 1$, $k_v = 0,8$.

Dla warunków początkowych (5.109) pozycja robotów na płaszczyźnie podczas eksperymentu została przedstawiona na rysunku 5.38.

$$\begin{bmatrix} p(0) \\ \varphi(0) \end{bmatrix} = \begin{bmatrix} 0,0 & -1,6 & -1,1 \\ 0,0 & -0,3 & 1,1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \text{m} \\ \text{m} \\ \text{rad} \end{bmatrix}. \tag{5.109}$$

Na wykresie 5.49 zaznaczono początkowe pozycje robotów (koło), pozycje końcowe (diament) oraz trajektorie robotów 1, 2 i 3. Końcowa odległość pomiędzy robotami została oznaczona czarnymi odcinkami. W pierwszej fazie eksperymentu (lider w bezruchu) trajektorie robotów 2 i 3 są podobne do zachowania obserwowanego w poprzednim przykładzie. W drugiej fazie eksperymentu lider rozpoczyna ruch zgodnie z zadanym sygnałem referencyjnym. Roboty 2 i 3 podążają za liderem utrzymując zadaną formację.

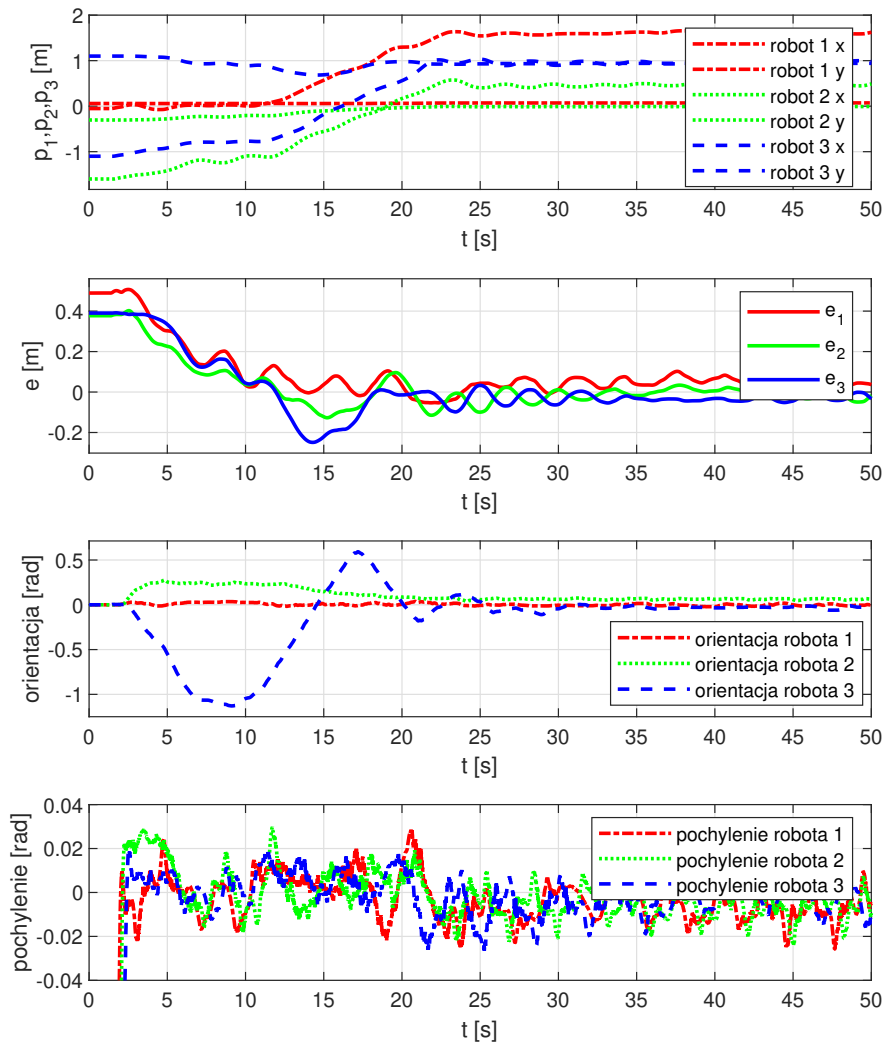


Rysunek 5.49: Formacja robotów w algorytmie śledzenia lidera, lider porusza się po prostej

Na wykresie 5.50 przedstawiono przebieg położenia robotów, uchyb wzajemnej odległości (czyli wartości $e_1 = \|p_1 - p_2\| - d_1$), itd.), orientację oraz pochylenie robotów.

Przebieg czasowy położenia robotów pokazuje, że do 12 sekundy lider pozostaje w bezruchu (poza ruchami korygującymi postawę), pomiędzy 12 a 25 sekundą zmienia swoje położenie a następnie zatrzymuje się. W pierwszej fazie eksperymentu roboty 2 i 3 korygują swoje położenie, aby osiągnąć zadane odległości. Następnie podążają za liderem utrzymując zadane odległości. W końcowej fazie eksperymentu roboty korygują swoje położenie w celu utrzymania pionowej postawy. Powoduje to oscylację uchybu zadanych odległości wokół wartości zero.

Wyniki potwierdzają, że proponowany algorytm śledzenia lidera poruszającego się wzdłuż linii działa poprawnie w rzeczywistych warunkach.



Rysunek 5.50: Przebieg położenia, uchybu wzajemnej odległości, orientacji robotów oraz pochylenia robotów

5.4.4 Podsumowanie wyników

Wykonane badania eksperymentalne z wykorzystaniem rzeczywistego robota pokazują, że proponowane algorytmy sterujące działają poprawnie w realnych warunkach. W badaniach przyjęto szereg uproszczeń, jednak proponowany scenariusz testowy wydaje się dobrze odzwierciedlać sytuację sterowania zespołu robotów.

Badania eksperymentalne pozwoliły przetestować działanie algorytmu i jego odporność na niemodelowane zjawiska. Prezentowane przebiegi pozwalają przypuszczać, że dwa dominujące zjawiska wpływające na jakość sterowania to tarcie statyczne oraz szum pomiarowy. Drugie z tych zjawisk powodują konieczność ciągłej korekcy położenia robota w celu utrzymania pionowej postawy, pierwsze utrudnia realizację zadania. Mimo oscylacyjnego ruchu robota na płaszczyźnie algorytmy sterowania formacją pozwalają zbliżyć uchyb regulacji do wartości bliskich zeru.

6 Podsumowanie

Przedmiotem badań opisanych w rozprawie były rozproszone algorytmy sterowania zespołem współpracujących robotów mobilnych. Analizowano metody sterowania bazujące na wzajemnych odległościach pomiędzy pojazdami, przy założeniu, że globalny układ współrzędnych jest nieznany dla poszczególnych pojazdów. Punktem wyjścia prowadzonych badań były prace dotyczące sterowania formacją robotów mobilnych, w których wykorzystano sztuczne potencjały do definicji algorytmu osiągniętego zadania formację [79, 15, 53] (rozdział 3).

Głównym celem pracy było opracowanie algorytmów sterowania rozproszonego dla układów zbudowanych z wielu autonomicznych robotów mobilnych. Dodatkowym celem była budowa środowiska sprzętowo — programowego umożliwiającego prowadzenie badań eksperymentalnych algorytmów sterowania dla niestabilnych robotów mobilnych.

Pierwszy cel pracy został osiągnięty przez opracowanie algorytmów sterujących dla uproszczonych modeli pojazdów, a następnie ich adaptację dla robotów mobilnych, których kinematyka i dynamika jest opisana bardziej złożonymi modelami. W przypadku robotów klasy (2,0) wprowadzono przekształcenie sterowania dostosowane do ograniczeń nieholonomicznych (rozdział 4). W przypadku niestabilnych robotów zaprojektowano kaskadowy układ regulacji, gdzie wewnętrzna pętla sprzężenia zwrotnego odpowiedzialna była za utrzymanie robota w górnym, niestabilnym punkcie równowagi oraz nadążanie za sygnałami referencyjnymi, a zewnętrzna pętla sprzężenia zwrotnego realizowała zadanie sterowania formacją (rozdział 5). Skonstruowane algorytmy posiadają cechy sterowania lokalnego umożliwiając ich rozproszoną implementację.

Weryfikacja proponowanych metod sterowania została przeprowadzona z wykorzystaniem symulacji komputerowych oraz badania na rzeczywistych stanowiskach robotów mobilnych. Wyniki eksperymentów potwierdziły skuteczność proponowanych algorytmów sterowania w różnych scenariuszach, co podkreśla ich aplikacyjny charakter. Uzyskane rezultaty są spójne z przewidywaniami opisanymi w [60] oraz [105].

Dodatkowy cel pracy został osiągnięty przez budowę stanowiska laboratoryjnego złożonego z niestabilnych robotów mobilnych (sekcja 5.1). Integracja jednostki sterującej robotem z platformą sprzętowo – programową RTDAQ/Zynq zapewniła możliwość użycia graficznych narzędzi modelowania i symulacji (MATLAB/Simulink) oraz odwoływanie się do modelu obiektu podczas projektowania i weryfikacji algorytmu sterowania. Takie podejście jest zgodne z metodologią projektowania algorytmów sterowania korzystającą z modelu matematycznego

obiektu (MBD). Przyjęte rozwiązanie znacznie ułatwiło przeprowadzenie procesów modelowania, identyfikacji i syntezy regulatorów. Zadanie to zostało zrealizowane we współpracy z firmą INTECO w ramach projektu badawczo-rozwojowego NCBIIR (INNOTECH-K1/HI1/12/158460/ NCBR/12), a powstałe urządzenie zostało wdrożone do oferty firmy INTECO.

Na podstawie sformułowanych celów pracy do jej głównych rezultatów należy zaliczyć:

1. Opracowanie i implementacja rozproszonych algorytmów sterujących dla zespołu niestabilnych robotów mobilnych. Głównymi składnikami proponowanego algorytmu są stabilizujący regulatora od stanu, regulator sterujący formacją i funkcja odwzorowująca kinematykę obiektu bez ograniczeń na sterowanie na kinematykę nieholonomicznego pojazdu.
2. Opracowanie algorytmu śledzenie lidera, w którym odległości pomiędzy robotami są zależne od prędkości lidera oraz weryfikacja eksperymentalna proponowanego algorytmu.
3. Implementacja i weryfikacja algorytmów sterujących dla zespołu robotów klasy (2,0).
4. Formalny opis układu, w tym wykazanie asymptotycznej stabilności oraz obszaru atrakcji punktów równowagi układu dla 2 obiektów i symetrycznej komunikacji oraz 3 obiektów i kierunkowej wymiany informacji.
5. Opracowanie modelu matematycznego dwukołowego, niestabilnego pojazdu i identyfikacja jego parametrów.
6. Budowa robota mobilnego i integracja z platformą sprzętowo-programową umożliwiającą prowadzenie badań eksperymentalnych.

6.1 Kierunki dalszych badań

Przedstawione w pracy rozwiązania nie wyczerpują wszystkich możliwości sterowania zespołem robotów mobilnych. Dotyczy to zarówno zaproponowanych algorytmów sterowania jak i założonego celu sterowania. Uzyskane wyniki mogą stanowić wstęp do dalszych badań.

Szczególnie interesujące jest przeprowadzenia badań eksperymentalnych z wykorzystaniem większej liczby niestabilnych robotów mobilnych. Otwiera to dodatkowe możliwości badawcze dotyczące, na przykład:

- opracowania i implementacji algorytmów wykrywania przeszkód i unikania kolizji,
- czasowej synchronizacji pomiarów i sterowania rozproszonego układu regulacji.

Inny ciekawy kierunek badań związany jest z wyposażeniem pojazdów w zestaw sensorów dostarczających informacji o względnym położeniu sąsiadów oraz otoczeniu. Wykorzystanie czujników takich jak kamera, lidar czy czujniki zbliżeniowe, a więc sensory dostępne we współczesnych samochodach, zwiększy aplikacyjny charakter prowadzonych badań oraz pozwoli na analizę wpływu zakłóceń pomiarowych na algorytmy sterowania formacją robotów.

Zagadnienia te będą stanowić temat przyszłych badań autora.

6.2 Wnioski

Przeprowadzone badania symulacyjne i eksperymentalne pozwoliły zrealizować wyznaczone zadania naukowe i przedstawione cele pracy. Na podstawie uzyskanych wyników można stwierdzić, że:

1. Rozproszone algorytmy sterowania, oparte na sztucznych potencjałach, są skuteczne w koordynacji zespołu robotów mobilnych.
2. Algorytmy bazujące na lokalnych pomiarach i sterowaniu pozwalają uzyskać globalny cel sterowania.
3. Przekształcenie sterowania obliczonego dla punktów kinematycznych na kinematykę nieholonomicznego robota zachowuje właściwość asymptotycznej stabilności układu.
4. Można sterować formacją robotów o niestabilnej dynamice pod warunkiem zastosowania stabilizującego regulatora.
5. Metodologia projektowania algorytmów oparta na modelowaniu matematycznym pozwala na ich weryfikację na wczesnych etapach. Dzięki temu przyczynia się do skrócenia czasu potrzebnego na wykonanie zadania.

Przedstawione rozwiązania mają potencjałe zastosowania w różnych dziedzinach, wśród których można wymienić: logistykę (np. zrobotyzowane magazyny), transport (np. systemy wspomagające kierowcę) czy technologie kosmiczne (np. satelitarne loty formacyjne). Podsumowując, praca wnosi istotny wkład w rozwój technologii sterowania rozproszonego zespołem współpracujących robotów mobilnych.

Załącznik A - Skrypt do przykładu 3.1.1

```
1 %% symulacja układu 2 punktów
2 opts = odeset('RelTol',1e-9,'AbsTol',1e-10); % ustawienia solwera
3
4 tSys=0:0.02:10; % definicja wektora czasu
5 % p0 = [.1; -.1; -.1; .2]; % definicja warunku początkowego BLISKO
6 p0 = [.8; -.8; -.8; 1.6]; % definicja warunku początkowego DALEKO
7 d = 0.8; % definicja zadanej odległości
8
9 % rozwiązanie równania różniczkowego opisującego dynamikę robotów
10 [t,p] = ode45(@(t,p) robotSys(t,p,d),tSys,p0,opts);
11
12 % przygotowanie danych
13 p1x = p(:,1); p1y = p(:,2);
14 p2x = p(:,3); p2y = p(:,4);
15
16 %% wykres XY
17 labCorr1 = 0.1; labCorr2 = 0.05;
18 figure
19 % trajektorie
20 hp{1} = plot(p1x,p1y,'r',LineWidth=3);
21 hold on, grid on
22 hp{2} = plot(p2x,p2y,'g',LineWidth=3);
23 % pkt. początkowe
24 hpsp{1} = plot(p1x(1),p1y(1),'ro', LineWidth=1.5);
25 hpsp{2} = plot(p2x(1),p2y(1),'go', LineWidth=1.5);
26 % pkt. końcowe
27 hpep{1} = plot(p1x(end),p1y(end),'rd', LineWidth=1.5);
28 hpep{2} = plot(p2x(end),p2y(end),'gd', LineWidth=1.5);
29
30 xlim([min([p1x; p2x])-labCorr1,max([p1x; p2x])+labCorr1])
31 ylim([min([p1y; p2y])-labCorr1,max([p1y; p2y])+labCorr1])
32 text(p1x(end)+labCorr2,p1y(end)+labCorr2,'1')
33 text(p2x(end)+labCorr2,p2y(end)+labCorr2,'2')
34
35 plot([p1x(end) p2x(end)], [p1y(end) p2y(end)], 'k--')
36 text(mean([p1x(end) p2x(end)])+labCorr2,mean([p1y(end) p2y(end)])+
37 labCorr2, ...
38 ['d(10) = ' num2str(norm(p(end,1:2)-p(end,3:4)))]])
```



```

39
40 handles = [hpsp{1} hpsp{2} hpep{1} hpep{2} hp{1} hp{2}];
41 legend(handles, 'punkt poczatkowy robota 1', 'punkt poczatkowy robota 2',
    ...
42     'punkt koncowy robota 1', 'punkt koncowy robota 2', ...
43     'trajektoria robota 1', 'trajektoria robota 2', ...
44     'Location','southwest')
45
46 %% wykresy p(t), e(t)
47 figure
48 subplot(3,1,1)
49 plot(t, p1x, t,p1y,'LineWidth', 1.5)
50 xlabel('t[s]'), ylabel('p_{1}[m]'), ylim('padded')
51 legend('wspolrzeczna x','wspolrzeczna y')
52 grid on
53 subplot(3,1,2)
54 plot(t, p2x, t,p2y,'LineWidth', 1.5)
55 xlabel('t[s]'), ylabel('p_{2}[m]'), ylim('padded')
56 legend('wspolrzeczna x','wspolrzeczna y')
57 grid on
58 subplot(3,1,3)
59 plot(t, vecnorm(p(:,1:2)-p(:,3:4),2,2).^2-d^2 , 'LineWidth', 1.5)
60 xlabel('t[s]'), ylabel('e'), ylim('padded')
61 legend('uchybnosc')
62 grid on
63
64 function dpdt = robotSys(~,p,d)
65 % rownania ruchu ukladu robotow
66     r = p(1:2) - p(3:4);
67     dp1 = -r * (norm(r)^2 - d^2);
68     dp2 = r * (norm(r)^2 - d^2);
69     dpdt=[dp1; dp2];
70 end

```

Załącznik B - Skrypt do przykładu 3.1.2

```
1 %% symulacja układu 3 punktów, cykliczna pogon
2 opts = odeset('RelTol',1e-9,'AbsTol',1e-10); % ustawienia solwera
3
4 tSys=0:0.02:10; % definicja wektora czasu
5 p0 = [0; -.1; -.1; .2; .2; .1]; kp=1; % definicja warunku początkowego
   BLISKO
6 % p0 = [.8; -.8; -.8; 1.6; -2; 0]; kp=0.25 % definicja warunku początkowego
   DALEKO
7 % p0 = [-.8; .8; 0; 0; .8; -.8]; kp=0.1 % definicja warunku początkowego
   WSPOLINIOWO
8 d = [1; 1; 1]; % definicja zadanych odleglosci
9
10 % rozwiązanie równania różniczkowego opisującego dynamikę robotów
11 [t,p] = ode45(@(t,p) robotSys(t,p,d,kp),tSys,p0,opts);
12
13 % przygotowanie danych
14 p1x = p(:,1); p1y = p(:,2);
15 p2x = p(:,3); p2y = p(:,4);
16 p3x = p(:,5); p3y = p(:,6);
17
18 %% wykres XY
19 labCorr1 = 0.1; labCorr2 = 0.05;
20 figure
21 % trajektorie
22 hp{1} = plot(p1x,p1y,'r',LineWidth=3);
23 hold on, grid on
24 hp{2} = plot(p2x,p2y,'g',LineWidth=3);
25 hp{3} = plot(p3x,p3y,'b',LineWidth=3);
26 % pkt. początkowe
27 hpsp{1} = plot(p1x(1),p1y(1),'ro', LineWidth=1.5);
28 hpsp{2} = plot(p2x(1),p2y(1),'go', LineWidth=1.5);
29 hpsp{3} = plot(p3x(1),p3y(1),'bo', LineWidth=1.5);
30 % pkt. końcowe
31 hpep{1} = plot(p1x(end),p1y(end),'rd', LineWidth=1.5);
32 hpep{2} = plot(p2x(end),p2y(end),'gd', LineWidth=1.5);
33 hpep{3} = plot(p3x(end),p3y(end),'bd', LineWidth=1.5);
34
35 axis equal, xlim('padded')
```

```

36 text(p1x(end)+labCorr2,p1y(end)+labCorr2,'1')
37 text(p2x(end)-labCorr2,p2y(end)+labCorr2,'2')
38 text(p3x(end)+labCorr2,p3y(end)-labCorr2,'3')
39
40 plot([p1x(end) p2x(end)], [p1y(end) p2y(end)], 'k--')
41 plot([p2x(end) p3x(end)], [p2y(end) p3y(end)], 'k--')
42 plot([p3x(end) p1x(end)], [p3y(end) p1y(end)], 'k--')
43
44 text(mean([p1x(end) p2x(end)])-2.5*labCorr2,mean([p1y(end) p2y(end)])+2.5*
    labCorr2, ...
45     ['d_1(10) = ' num2str(norm(p(end,1:2)-p(end,3:4)), '%.2f')])
46 text(mean([p1x(end) p3x(end)])-0*labCorr2,mean([p1y(end) p3y(end)])-2*
    labCorr2, ...
47     ['d_2(10) = ' num2str(norm(p(end,1:2)-p(end,5:6)), '%.2f')])
48 text(mean([p2x(end) p3x(end)])-2*labCorr2,mean([p2y(end) p3y(end)])+2*
    labCorr2, ...
49     ['d_3(10) = ' num2str(norm(p(end,3:4)-p(end,5:6)), '%.2f')])
50
51 handles = [hpsp{1} hpep{1} hp{1} hp{2} hp{3}];
52 legend(handles, 'punkt poczatkowy robotow', 'punkt koncowy robotow', ...
53     'trajektoria robota 1', 'trajektoria robota 2', 'trajektoria robota 3'
54     , ...
55     'Location','southwest')
56 %% wykresy p(t), e(t)
57 figure
58 subplot(4,1,1)
59 plot(t, p1x, t,p1y,'LineWidth', 1.5)
60 xlabel('t[s]'), ylabel('p_{1}[m]'), ylim('padded')
61 legend('wspolrzeczna x','wspolrzeczna y')
62 grid on
63 subplot(4,1,2)
64 plot(t, p2x, t,p2y,'LineWidth', 1.5)
65 xlabel('t[s]'), ylabel('p_{2}[m]'), ylim('padded')
66 legend('wspolrzeczna x','wspolrzeczna y')
67 grid on
68 subplot(4,1,3)
69 plot(t, p3x, t,p3y,'LineWidth', 1.5)
70 xlabel('t[s]'), ylabel('p_{3}[m]'), ylim('padded')
71 legend('wspolrzeczna x','wspolrzeczna y')

```

```

72 grid on
73 subplot(4,1,4)
74 plot(t, vecnorm(p(:,1:2)-p(:,3:4),2,2).^2 - d(1)^2, 'r', 'LineWidth', 1.5)
75 hold on;
76 plot(t, vecnorm(p(:,3:4)-p(:,5:6),2,2).^2 - d(2)^2, 'g', 'LineWidth', 1.5)
77 plot(t, vecnorm(p(:,5:6)-p(:,1:2),2,2).^2 - d(3)^2, 'b', 'LineWidth', 1.5)
78 xlabel('t[s]'), ylabel('e'), ylim('padded')
79 legend('e_1', 'e_2', 'e_3')
80 grid on
81
82 function dpdt = robotSys(~,p,d,kp)
83 % rownania ruchu ukladu robotow
84     r1 = p(1:2) - p(3:4);
85     r2 = p(3:4) - p(5:6);
86     r3 = p(5:6) - p(1:2);
87     dp1 = -r1*kp * (norm(r1)^2 - d(1)^2);
88     dp2 = -r2*kp * (norm(r2)^2 - d(2)^2);
89     dp3 = -r3*kp * (norm(r3)^2 - d(3)^2);
90     dpdt=[dp1; dp2; dp3];
91 end

```

Załącznik C - Nagrania eksperymentów

Tabela 6.1: Wykaz nagrań przeprowadzonych eksperymentów na rzeczywistych robotach

Opis	Warunek początkowy	Odnośnik
Dwa roboty, przykład wprowadzający 3.1.1, początkowa odległość większa niż zadana	$\begin{bmatrix} -0,8 & 1,1 \\ -0,4 & -0,5 \end{bmatrix}$	youtu.be/VsKDNuH4Sn0
Formacja trzech robotów, przykład 3.1.2 graf skierowany z cyklem	$\begin{bmatrix} -0,8 & 1,1 & 1,2 \\ -0,4 & -0,5 & 0,6 \end{bmatrix}$	youtu.be/svab5bG34kY
Algorytm śledzenia lidera, formacja trzech robotów, graf skierowany bez cyklu	$\begin{bmatrix} -0,5 & 0,9 & 1,2 \\ 0,1 & 0,7 & -0,6 \end{bmatrix}$	youtu.be/rNwqAlyOook
Algorytm śledzenie lidera, formacja trzech robotów, eksperyment ze zmiennymi wagami	$\begin{bmatrix} -0,7 & -1,4 & -1,3 \\ -0,3 & -0,4 & -0,7 \end{bmatrix}$	youtu.be/hLx63xc1iW4
Algorytm śledzenie lidera, formacja trzech robotów, eksperyment ze zmiennymi wagami	$\begin{bmatrix} -0,7 & -1,4 & -1,3 \\ 0,2 & 0,2 & -0,2 \end{bmatrix}$	youtu.be/8hi7yfHi9HE
Formacja pięciu robotów, orientacja negatywna	$\begin{bmatrix} -1,1 & 0,9 & 1,2 & -1,0 & 0,0 \\ 0,6 & 0,7 & -0,6 & 0,5 & 0,0 \end{bmatrix}$	youtu.be/mCrDJKqkFnY
Formacja pięciu robotów, orientacja pozytywna	$\begin{bmatrix} -0,8 & 1,1 & 1,2 & -1,0 & 0,0 \\ -0,4 & -0,5 & 0,6 & 0,5 & 0,0 \end{bmatrix}$	youtu.be/BV-GqQnSnS8

Tabela 6.2: Wykaz wizualizacji eksperymentów przeprowadzonych na modelach niestabilnych pojazdów mobilnych

Opis	Warunek początkowy	Odnosićnik
Przykład wprowadzający, dwa roboty zbliżające się na zadaną odległość	$\begin{bmatrix} -1,5 & 1,3 \\ 0,2 & -0,4 \\ \frac{\pi}{3} & \frac{\pi}{4} \end{bmatrix}$	youtu.be/r64GVAkxASU
Algorytm cyklicznej pogoni dla trzech robotów	$\begin{bmatrix} -1,5 & 1,0 & 1,1 \\ 1,2 & 1,5 & -1,0 \\ \frac{\pi}{6} & \frac{\pi}{3} & 0 \end{bmatrix}$	youtu.be/R-4AacnAJI
Algorytm śledzenie lidera dla trzech robotów, stacjonarny lider	$\begin{bmatrix} 1,0 & -1,0 & -0,6 \\ 0,0 & -0,1 & 1,1 \\ 0 & \frac{\pi}{6} & -\frac{\pi}{12} \end{bmatrix}$	youtu.be/F_xm6Fnh8x4
Algorytm śledzenie lidera dla trzech robotów, lider porusza się po prostej	$\begin{bmatrix} 1,0 & -1,0 & -0,6 \\ 0,0 & -0,1 & 1,1 \\ 0 & \frac{\pi}{6} & -\frac{\pi}{12} \end{bmatrix}$	youtu.be/gXFVxIyTC3g
Algorytm śledzenie lidera dla trzech robotów, lider porusza się po łuku	$\begin{bmatrix} 1,0 & -1,0 & -0,6 \\ 0,0 & -0,1 & 1,1 \\ 0 & \frac{\pi}{6} & -\frac{\pi}{12} \end{bmatrix}$	youtu.be/cY9ZHyeEnHo
Algorytm sterowania formacją pięciu robotów, formacja o orientacji pozytywnej	$\begin{bmatrix} -1,5 & -0,5 & 1,1 & 0,6 & -0,6 \\ 1,2 & -1,5 & -1,0 & 1,7 & 0,1 \\ \frac{\pi}{2} & -\frac{\pi}{2} & -\frac{\pi}{8} & -\frac{\pi}{2} & \pi \end{bmatrix}$	youtu.be/9yAtdyo7REA
Algorytm sterowania formacją pięciu robotów, formacja o orientacji negatywnej	$\begin{bmatrix} -1,5 & 1,0 & 1,1 & -0,8 & -0,5 \\ 1,2 & 1,5 & -1,0 & -1,5 & 0,1 \\ \pi & -\frac{\pi}{2} & -\frac{\pi}{8} & -\frac{\pi}{2} & \pi \end{bmatrix}$	youtu.be/JOdUaZGWjIY
Algorytm sterowania formacją pięciu robotów, formacja o orientacji negatywnej, wersja 2 (brak normalizacji wektora kierunku)	$\begin{bmatrix} -1,5 & 1,0 & 1,1 & -0,8 & -0,5 \\ 1,2 & 1,5 & -1,0 & -1,5 & 0,1 \\ \pi & -\frac{\pi}{2} & -\frac{\pi}{8} & -\frac{\pi}{2} & \pi \end{bmatrix}$	youtu.be/tzyoy1XhYTE

Literatura

- [1] Abdollah Amirkhani and Amir Hossein Barshooi. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5):3897–3935, 2022.
- [2] Brian D. O. Anderson, Changbin Yu, Soura Dasgupta, and A. Stephen Morse. Control of a three-coleader formation in the plane. *Systems & Control Letters*, 56(9-10):573–578, 2007.
- [3] Brian D.O. Anderson, Changbin Yu, Baris Fidan, and Julien M. Hendrickx. Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine*, 28(6):48–63, 2008.
- [4] Ronald C Arkin. *Behavior-based robotics*. MIT press, 1998.
- [5] K.J. Åström and R. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.
- [6] K.J. Åström and K. Furuta. Swinging up a pendulum by energy control. *IFAC Proceedings Volumes*, 29(1):1919–1924, 1996. 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July.
- [7] J. Baillieul and A. Suri. Information patterns and hedging brockett’s theorem in controlling vehicle formations. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 1, pages 556–563 Vol.1, 2003.
- [8] Jared B. Bancroft and Gérard Lachapelle. Estimating mems gyroscope g-sensitivity errors in foot mounted navigation. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, pages 1–6, 2012.
- [9] Piotr Bania. Estymacja stanu w układach liniowych przy skwantowanych pomiarach wyjść. *Automatyka/Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie*, 15:15–28, 2011.
- [10] J. Baranowski, K. Hajduk, W. Mitkowski, A. Korytowski, and A.J. Tutaj. *Teoria sterowania: materiały pomocnicze do ćwiczeń laboratoryjnych*. Skrypty Uczelniane - Akademia Górniczo-Hutnicza im. Stanisława Staszica. Wydawnictwa AGH, 2015.
- [11] A. Belabbas, S. Mou, A. S. Morse, and B. D. O. Anderson. Robustness issues with undirected formations. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 1445–1450, 2012.

- [12] Witold Byrski. *Obserwacja i sterowanie w systemach dynamicznych*. Wydawnictw AGH, 2007.
- [13] Mitch Campion, Prakash Ranganathan, and Saleh Faruque. Uav swarm communication and control architectures: a review. *Journal of Unmanned Vehicle Systems*, 7(2):93–106, 2019.
- [14] M. Cao, B. D. O. Anderson, A. S. Morse, and C. Yu. Control of acyclic formations of mobile autonomous agents. In *47th IEEE Conference on Decision and Control*, pages 1187–1192, 2008.
- [15] M. Cao, A. S. Morse, C. Yu, B. D. O. Anderson, and S. Dasguyta. Controlling a triangular formation of mobile autonomous agents. In *2007 46th IEEE Conference on Decision and Control*, pages 3603–3608, 2007.
- [16] Ming Cao, Changbin Yu, A Stephen Morse, BDO Anderson, and Soura Dasgupta. Generalized controller for directed triangle formations. *IFAC Proceedings Volumes*, 41(2):6590–6595, 2008.
- [17] Krzysztof Cetnarowicz, Pablo Gruer, Vincent Hilaire, and Abder Koukam. A formal specification of m-agent architecture. In *International Workshop of Central and Eastern Europe on Multi-Agent Systems*, pages 62–72. Springer, 2001.
- [18] L. Chen, M. Shi, H. G. d. Marina, and M. Cao. Stabilizing and maneuvering angle rigid multiagent formations with double-integrator agent dynamics. *IEEE Transactions on Control of Network Systems*, 9(3):1362–1374, 2022.
- [19] Liangming Chen, Ming Cao, Zhiyong Sun, Brian D.O. Anderson, and Chuanjiang Li. Angle-based formation shape control with velocity alignment. *IFAC-PapersOnLine*, 53(2), 2020.
- [20] Liangming Chen, Hector Garcia de Marina, and Ming Cao. Maneuvering formations of mobile agents using designed mismatched angles. *IEEE Transactions on Automatic Control*, 67(4):1655–1668, 2022.
- [21] Hector Garcia de Marina, Zhiyong Sun, Ming Cao, and Brian D. O. Anderson. Controlling a triangular flexible formation of autonomous agents. *IFAC-PapersOnLine*, 50(1):594–600, 2017.
- [22] Carlos Canudas de Wit, Bruno Siciliano, and Georges Bastin. *Theory of robot control*. Springer Science & Business Media, 2012.

- [23] J.P. Desai, J.P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.
- [24] Dimos V. Dimarogonas and Karl H. Johansson. On the stability of distance-based formation control. In *47th IEEE Conference on Decision and Control*. IEEE, 2008.
- [25] Xiwang Dong, Bocheng Yu, Zongying Shi, and Yisheng Zhong. Time-varying formation control for unmanned aerial vehicles - theories and applications. *IEEE Transactions on Control Systems Technology*, 46, 2016.
- [26] F. Dörfler and B. Francis. Formation control of autonomous robots based on cooperative behavior. In *European Control Conference (ECC)*, pages 2432–2437, 2009.
- [27] Tolga Eren, Peter N Belhumeur, Brian DO Anderson, and A Stephen Morse. A framework for maintaining formations based on rigidity. *IFAC Proceedings Volumes*, 35(1):499–504, 2002.
- [28] Pieter Eykhoff. *System identification*, volume 14. Wiley London, 1974.
- [29] Andrzej Firlit, Krzysztof Kołek, and Krzysztof Piątek. Heterogeneous active power filter controller. In *2017 International Symposium ELMAR*, pages 241–244, 2017.
- [30] Andrzej Firlit, Krzysztof Kołek, and Krzysztof Piątek. Rapid controller development for a dynamic voltage restorer. In *2017 International Symposium ELMAR*, pages 237–240, 2017.
- [31] Hector Garcia de Marina, Ming Cao, and Bayu Jayawardhana. Controlling rigid formations of mobile agents under inconsistent measurements. *IEEE Transactions on Robotics*, 31(1):31–39, February 2015.
- [32] Xiaohua Ge, Qing-Long Han, Derui Ding, Xian-Ming Zhang, and Boda Ning. A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems. *Neurocomputing*, 275:1684–1701, 2018.
- [33] A. Gmitterko and M. Vacková. Dynamic model of vehicle with two coaxial parallel wheels. In *2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 303–305, 2011.

- [34] P. Gorczyca, K. Kołek, M. Rosół, and D. Knapik. Rapid development of real-time applications for ARM Cortex A9. In Marcin W. Mastalerz Leszek Trybus, editor, *Design, development and implementation of real-time systems*, pages 145–155. Polish Information Processing Society, 2013.
- [35] P. Grabowski. *Ćwiczenia komputerowe z teorii sterowania*. Skrypty Uczelniane - Akademia Górniczo-Hutnicza im. Stanisława Staszica. Wydawnictwa AGH, 1996.
- [36] F. Grasser, A. D’Arrigo, S. Colombi, and A.C. Rufer. Joe: a mobile, inverted pendulum. *IEEE Transactions on Industrial Electronics*, 49(1):107–114, 2002.
- [37] Mohinder S Grewal and Angus P Andrews. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.
- [38] Henryk Górecki. *Optymalizacja i sterowanie systemów dynamicznych*. Wydawnictwa AGH, 2006.
- [39] David Halliday, Robert Resnick, and Jearl Walker. *Fundamentals of physics*. John Wiley & Sons, 2013.
- [40] Philip Hartman. A lemma in the theory of structural stability of differential equations. *Proceedings of the American Mathematical Society*, 11(4):610–620, 1960.
- [41] Min Hu, Guoqiang Zeng, and Hong Yao. Coordination control of distributed spacecraft system. In Rushi Ghadawala, editor, *Advances in Spacecraft Systems and Orbit Determination*, chapter 6. IntechOpen, Rijeka, 2012.
- [42] A. Jadbabaie, Lin Jie, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [43] Meng Ji and Magnus Egerstedt. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.
- [44] Mohamed A. Kamel, Xiang Yu, and Youmin Zhang. Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review. *Annual Reviews in Control*, 49:128–144, 2020.

- [45] Abhilash Kantamneni, Laura E Brown, Gordon Parker, and Wayne W Weaver. Survey of multi-agent systems for microgrid control. *Engineering applications of artificial intelligence*, 45:192–203, 2015.
- [46] Soumya Kar and José MF Moura. Distributed consensus algorithms in sensor networks: Quantized data and random link failures. *IEEE Transactions on Signal Processing*, 58(3):1383–1400, 2009.
- [47] Hassan K Khalil. *Control of nonlinear systems*. Prentice Hall, New York, NY, 2002.
- [48] Yeonhoon Kim, Soo Hyun Kim, and Yoon Keun Kwak. Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot. *Journal of Intelligent and Robotic Systems*, 44(1):25–46, 2005.
- [49] Dawid Knapik, Krzysztof Kołek, Maciej Rosół, and Andrzej Turnau. Efektywne kształcenie inżynierów z wykorzystaniem laboratoryjnych systemów mechatronicznych i stosowaniem metodologii szybkiego prototypowania algorytmów sterowania. *Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej*, 48:35–40, 2016.
- [50] W. Kowalczyk and K. Kozłowski. Trajectory tracking and collision avoidance for the formation of two-wheeled mobile robots. *Bulletin of the Polish Academy of Sciences Technical Sciences*, No. 5:915–924, 2019.
- [51] Krzysztof Kołek and Andrzej Firlit. A new optimal current controller for a three-phase shunt active power filter based on karush–kuhn–tucker conditions. *Energies*, 14(19), 2021.
- [52] Krzysztof Kołek and Maciej Rosół. Velocity estimation for slow motion devices equipped with encoder. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 809–814, 2017.
- [53] Laura Krick, Mireille E Broucke, and Bruce A Francis. Stabilisation of infinitesimally rigid formations of multi-robot networks. *International Journal of control*, 82(3):423–439, 2008.
- [54] Katie Laventall and Jorge Cortes. Coverage control by robotic networks with limited-range anisotropic sensory. In *2008 American Control Conference*, pages 2666–2671, 2008.

- [55] Yongfu Li, Chuancong Tang, Srinivas Peeta, and Yibing Wang. Nonlinear consensus-based connected vehicle platoon control incorporating car-following interactions and heterogeneous time delays. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2209–2219, 2019.
- [56] J. Lin, A.S. Morse, and B.D.O. Anderson. The multi-agent rendezvous problem. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 1508–1513 Vol.2, 2003.
- [57] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multi-agent Networks*, volume 33. Princeton University Press, 2010.
- [58] Wojciech Mitkowski. *Zarys teorii sterowania*. Wydawnictwa AGH, 2019.
- [59] S. Mou, M. Cao, and A.S. Morse. A distributed control law for acyclic formations. *IFAC Proceedings Volumes*, 44(1):7818–7823, 2011. 18th IFAC World Congress.
- [60] P. Ogren, M. Egerstedt, and X. Hu. A control lyapunov function approach to multi-agent coordination. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 2, pages 1150–1155 vol.2, 2001.
- [61] P. Ogren, E. Fiorelli, and N.E Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic control*, 49(8):1292–1302, 2004.
- [62] Kwang-Kyo Oh and Hyo-Sung Ahn. Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space. *International Journal of Robust and Nonlinear Control*, 24(12):1809–1820, 2014.
- [63] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.
- [64] R. Olfati-Saber. Near-identity diffeomorphisms and exponential ϵ -tracking and ϵ -stabilization of first-order nonholonomic $se(2)$ vehicles. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 6, pages 4690–4695 vol.6, 2002.
- [65] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

- [66] R. Olfati-Saber and R.M. Murray. Distributed structural stabilization and tracking for formations of dynamic multi-agents. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 1, pages 209–215 vol.1, 2002.
- [67] Reza Olfati-Saber and Richard M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. *IFAC Proceedings Volumes*, 35(1):495–500, 2002.
- [68] Adam Owczarkowski and Dariusz Horla. Robust LQR and LQI control with actuator failure of a 2DOF unmanned bicycle robot stabilized by an inertial wheel. *International Journal of Applied Mathematics and Computer Science*, 26(2):325–334, 2016.
- [69] Myoung-Chul Park, Kyungmin Jeong, and Hyo-Sung Ahn. Formation stabilization and resizing based on the control of inter-agent distances. *International Journal of Robust and Nonlinear Control*, 25(14):2532–2546, 2015.
- [70] Daniel Pickem, Myron Lee, and Magnus Egerstedt. The GRITSBot in its natural habitat - a multi-robot testbed. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4062–4067, 2015.
- [71] Ayush Rai and Shaoshuai Mou. Safe region multi-agent formation control with velocity tracking. *Systems & Control Letters*, 186:105776, 2024.
- [72] Jaime Luis Ramirez Riberos. *New decentralized algorithms for spacecraft formation control based on a cyclic approach*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [73] Wei Ren and Ella Atkins. Distributed multi-vehicle coordinated control *via* local information exchange. *International Journal of Robust and Nonlinear Control*, 17(10-11):1002–1033, 2007.
- [74] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, 1987.
- [75] B. Roth. Rigid and flexible frameworks. *The American Mathematical Monthly*, 88(1):6–21, 1981.
- [76] Farid Sahebsara and Marcio de Queiroz. Distance-based formation maneuvering of mobile robots with static obstacles. In *2022 IEEE Conference on Control Technology and Applications (CCTA)*, pages 470–475, 2022.

- [77] D.P. Scharf, F.Y. Hadaegh, and S.R. Ploen. A survey of spacecraft formation flying guidance and control. part ii: control. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 2976–2985 vol.4, 2004.
- [78] Andrzej MJ Skulimowski. Anticipatory network models of multicriteria decision-making processes. *International Journal of Systems Science*, 45(1):39–59, 2014.
- [79] Stephen L Smith, Mireille E Broucke, and Bruce A Francis. Stabilizing a multi-agent system to an equilateral polygon formation. In *Proc. of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 2415–2424, 2006.
- [80] Aakash Soni and Huosheng Hu. Formation control for a fleet of autonomous ground vehicles: A survey. *Robotics*, 7(4), 2018.
- [81] Haifan Su, Cailian Chen, Ziwen Yang, Shanying Zhu, and Xinping Guan. Bearing-based formation tracking control with time-varying velocity estimation. *IEEE Transactions on Cybernetics*, 53(6), 2023.
- [82] K.R. Symon. *Mechanics*. Addison-Wesley series in physics. Addison-Wesley Publishing Company, 1971.
- [83] Torsten Söderström and Petre Stoica. *Identyfikacja systemów*. Wydawnictwo Naukowe PWN, 1997.
- [84] Anam Tahir, Jari Böling, Mohammad-Hashem Haghbayan, Hannu T. Toivonen, and Juha Plosila. Swarms of unmanned aerial vehicles — a survey. *Journal of Industrial Information Integration*, 16:100106, 2019.
- [85] Leszek Trybus, Feliks Bator, Wiesław Pecka, Maciej Walaszek, and Marian Wysocki. *Teoria sterowania i teoria systemów: laboratorium*. Dział Wydawnictw Politechniki Rzeszowskiej, 1982.
- [86] Andrzej Turnau. *Sterowanie docelowe układami nieliniowymi w czasie rzeczywistym – algorytmy inteligentne i optymaln czasowe*. Wydawnictwa AGH, 2002.
- [87] Thijs H.A. van den Broek, Nathan van de Wouw, and Henk Nijmeijer. Formation control of unicycle mobile robots: a virtual structure approach. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8328–8333, 2009.

- [88] Tamas Vicsek, Andras Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Sochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6), 1995.
- [89] Bang Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 43(4):1–53, 2011.
- [90] Qin Wang, Yu-Ping Tian, and Yao-Jin Xu. Globally asymptotically stable formation control of three agents. In *Proceedings of the 2011 American Control Conference*, pages 4008–4013, 2011.
- [91] Gerhard Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.
- [92] Wikipedia. Relaxed stability. https://en.wikipedia.org/wiki/Relaxed_stability. Dostęp: 2024-06-15.
- [93] Robin J. Wilson. *Wprowadzenie do teorii grafów*. Wydawnictwo Naukowe PWN, 2007.
- [94] Sean Wilson and Magnus Egerstedt. The Robotarium: A remotely-accessible, multi-robot testbed for control research and education. *IEEE Open Journal of Control Systems*, 2:12–23, 2023.
- [95] Sean Wilson, Paul Glotfelter, Siddharth Mayya, Gennaro Notomista, Yousef Emam, Xiaoyi Cai, and Magnus Egerstedt. The Robotarium: Automation of a remotely accessible, multi-robot testbed. *IEEE Robotics and Automation Letters*, 6(2):2922–2929, 2021.
- [96] Sean Wilson, Paul Glotfelter, Li Wang, Siddharth Mayya, Gennaro Notomista, Mark Mote, and Magnus Egerstedt. The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Systems Magazine*, 40(1):26–44, 2020.
- [97] Peter R Wurman, Raffaello D’Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1):9–9, 2008.
- [98] Huikai Xie and Gary K. Fedder. Integrated microelectromechanical gyroscopes. *Journal of Aerospace Engineering*, 16(2):65–75, 2003.

- [99] Yoriyisa Yamamoto. NXTway-GS (self-balancing two-wheeled robot) controller design. Technical report, MATLAB Central File Exchange, 2004.
- [100] P. C. Young and J. C. Willems. An approach to the linear multivariable servomechanism problem†. *International Journal of Control*, 15(5):961–979, 1972.
- [101] Wenwu Yu, Guanrong Chen, and Ming Cao. Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica*, 46(6):1089–1095, 2010.
- [102] Paweł Zagórski. *Nanosatellite attitude estimation*. PhD thesis, AGH University of Krakow, 2017.
- [103] S.H. Żak. *Systems and Control*. Oxford series in electrical and computer engineering. Oxford University Press, 2003.
- [104] Daniel Zelazo, Antonio Franchi, Heinrich H. Bühlhoff, and Paolo Robuffo Giordano. Decentralized rigidity maintenance control with range measurements for multi-robot systems. *The International Journal of Robotics Research*, 34(1):105–128, November 2014.
- [105] Shiyu Zhao and Zhiyong Sun. Defend the practicality of single-integrator models in multi-robot coordination control. In *2017 13th IEEE International Conference on Control & Automation (ICCA)*, pages 666–671, 2017.
- [106] Yajun Zheng, Qingling Wang, Dongpu Cao, Baris Fidan, and Changyin Sun. Distance-based formation control for multi-lane autonomous vehicle platoons. *IET Control Theory & Applications*, 15(11):1506–1517, 2021.
- [107] Yongkun Zhou, Bin Rao, and Wei Wang. Uav swarm intelligence: Recent advances and future trends. *IEEE Access*, 8:183856–183878, 2020.
- [108] Quanser. <https://www.quanser.com/products/qbot-platform/>. Dostęp: 2024-04-30.
- [109] ROSbot. <https://husarion.com/manuals/rosbot/>. Dostęp: 2024-04-30.
- [110] Mathworks, simulink 3D Animation. <https://www.mathworks.com/products/3d-animation.html>. Dostęp: 2024-05-31.
- [111] Turtlebot. <https://clearpathrobotics.com/turtlebot-4/>. Dostęp: 2024-04-30.