

WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ MECHATRONIKI, UZBROJENIA I LOTNICTWA

Instytut Techniki Rakietowej i Mechatroniki



Rozprawa doktorska

Modelowanie ruchu i synchroniczne sterowanie
grupą robotów mobilnych o strukturze rozproszonej

Opracował:

mgr inż. Michał SIWEK

Promotor:

dr hab. inż. Leszek BARANOWSKI, prof. WAT

Promotor pomocniczy:

dr inż. Jarosław PANASIUK

Warszawa 2023

SPIS TREŚCI

Wykaz ważniejszych oznaczeń stosowanych w pracy	7
WSTĘP	9
ANALIZA LITERATURY	13
1.1. Przegląd algorytmów sterowania grupami robotów mobilnych	18
1.1.1. Algorytm Śledzenia Lidera	18
1.1.2. Algorytm Wirtualnej Struktury	20
1.1.3. Algorytm Behawioralny	22
1.1.4. Algorytm Sztucznych Pól Potencjałowych	23
1.1.5. Algorytm Teorii Grafów	24
1.1.6. Algorytm Konsensusu	25
1.2. Analiza kontrolerów stosowanych do realizacji zadania śledzenia trajektorii	27
1.3. Analiza błędów wpływających na dokładność pozycjonowania robotów w grupie rozproszonej	30
1.4. Podsumowanie analizy literatury	32
OBIEKT BADAŃ	35
2.1. Laboratoryjny robot TURTLEBOT 2	35
2.2. Robot Operating System	38
2.3. Opracowanie modelu matematycznego robota TURTLEBOT 2	40
2.3.1. Model kinematyczny	40
2.3.2. Model dynamiczny	43
2.3.3. Identyfikacja parametrów modelu dynamicznego	46
2.4. Walidacja modelu matematycznego robota TURTLEBOT 2	51
2.5. Opracowanie modelu teoretycznego grupy robotów o strukturze rozproszonej	55
BADANIA ROZPOZNAWCZE DOKŁADNOŚCI POZYCJONOWANIA ROBOTÓW W GRUPIE O STRUKTURZE ROZPROSZONEJ	57
3.1. Badania laboratoryjne wpływu ilości robotów w grupie i rozmieszczenia przeszkód środowiskowych na opóźnienia sygnałów sterujących	58
3.1.1. Zakres, metodyka i wyniki badań laboratoryjnych wpływu ilości robotów w grupie o strukturze rozproszonej na opóźnienia sygnałów sterujących	58
3.1.2. Zakres, metodyka i wyniki badań wpływu rozmieszczenia przeszkód w środowisku pracy grupy robotów na opóźnienia sygnałów sterujących	62
3.2. Badania symulacyjne wpływu opóźnień sygnałów sterujących na ruch grupy robotów o strukturze rozproszonej	64
3.2.1. Zakres i metodyka badań wpływu opóźnień sygnałów sterujących na ruch grupy robotów	65
3.2.2. Wyniki badań wpływu opóźnień sygnałów sterujących na ruch grupy robotów	67
3.3. Badania symulacyjne wzajemnego oddziaływania robotów w grupie rozproszonej transportującej ładunek	72

3.3.1. Zakres i metodyka badań wzajemnego oddziaływania robotów w grupie rozproszonej transportującej ładunek	73
3.3.2. Analiza statyczna obciążeń wynikających z fizycznego połączenia robotów	74
3.3.3. Wyniki analizy statycznej obciążeń wynikających z fizycznego połączenia robotów	75
3.3.4. Analiza dynamiczna obciążeń wynikających z opóźnień sygnałów sterujących	76
3.3.5. Wyniki analizy dynamicznej obciążeń wynikających z opóźnień sygnałów sterujących	76
3.4. Podsumowanie badań rozpoznawczych	78
OPRACOWANIE AUTORSKIEGO ALGORYTMU SYNCHRONICZNEGO STEROWANIA GRUPĄ ROBOTÓW O STRUKTURZE ROZPROSZONEJ.....	81
4.1. Generator trajektorii wzorcowej dla grupy	82
4.2. Budowa konfiguracji grupy	83
4.3. Kontroler konfiguracji grupy	84
4.4. Kontroler kinematyczny	87
4.5. Kontroler dynamiczny	90
BADANIA SYMULACYJNE ALGORYTMU SYNCHRONICZNEGO STEROWANIA GRUPĄ ROBOTÓW O STRUKTURZE ROZPROSZONEJ	93
5.1. Zakres i metodyka badań teoretycznych warstwy wewnętrznej W2 algorytmu sterowania	93
5.2. Badania symulacyjne warstwy wewnętrznej W2 algorytmu sterowania	96
5.2.1. Badania kontrolera kinematycznego.....	97
5.2.2. Badania kontrolera kinematyczno-dynamicznego.....	110
5.2.3. Badania kontrolera kinematyczno-dynamicznego z adaptacją.....	122
5.2.4. Podsumowanie badań teoretycznych warstwy wewnętrznej W2 algorytmu sterowania	134
5.3. Zakres i metodyka badań teoretycznych warstwy zewnętrznej W1 algorytmu sterowania	137
5.4. Badania symulacyjne warstwy zewnętrznej W1 algorytmu sterowania.....	138
5.4.1. Badania symulacyjne synchronizacji rozpoczęcia ruchu przez grupę robotów	138
5.4.2. Badania symulacyjne synchronicznego ruchu grupy po zadanej trajektorii..	143
5.4.3. Podsumowanie badań teoretycznych warstwy zewnętrznej W1 algorytmu sterowania	148
5.5. Podsumowanie badań teoretycznych	148

BADANIA DOŚWIADCZALNE ALGORYTMU SYNCHRONICZNEGO STEROWANIA GRUPĄ ROBOTÓW MOBILNYCH O STRUKTURZE ROZPROSZONEJ	151
6.1. Zakres i metodyka badań doświadczalnych warstwy wewnętrznej W2 algorytmu sterowania	151
6.2. Badania doświadczalne warstwy wewnętrznej W2 algorytmu sterowania	154
6.2.1. Badania kontrolera kinematycznego.....	155
6.2.2. Badania kontrolera kinematyczno-dynamicznego.....	168
6.2.3. Badania kontrolera kinematyczno-dynamicznego z adaptacją.....	180
6.2.4. Podsumowanie badań doświadczalnych warstwy wewnętrznej W2 algorytmu sterowania	192
6.3. Zakres i metodyka badań doświadczalnych warstwy zewnętrznej W1 algorytmu sterowania	195
6.4. Badania doświadczalne warstwy zewnętrznej W1 algorytmu sterowania	196
6.4.1. Badania doświadczalne synchronizacji rozpoczęcia ruchu przez grupę robotów	196
6.4.2. Badania doświadczalne synchronicznego ruchu grupy po zadanej trajektorii	201
6.4.3. Podsumowanie badań doświadczalnych warstwy zewnętrznej W1 algorytmu sterowania	206
6.5. Podsumowanie badań doświadczalnych.....	206
PODSUMOWANIE I WNIOSKI KOŃCOWE	209
Załącznik A	215
Załącznik B	217
LITERATURA.....	219

Wykaz ważniejszych oznaczeń stosowanych w pracy

$a_v, a_\omega(t)$ – przyspieszenie liniowe i kątowe robota

d – odległość koła napędowego od środka geometrycznego robota

d_1, d_2, d_3 – wymiary palety transportowej

$e_{ij}(t)$ – uchyb odległości pomiędzy i-tym a j-tym robotem

$e_v(t), e_\omega(t)$ – uchyb prędkości liniowej i kątowej

$e_x(t), e_y(t)$ – uchyby pozycji

$e_{xy}(t)$ – uchyb położenia robota

$e_\theta(t)$ – uchyb orientacji

g_{ij} – waga określająca połączenia komunikacyjne pomiędzy robotami i oraz j

$h_i(t)$ – wektor stanu i-tego robota w grupie o strukturze rozproszonej

k_1, k_2 – wzmocnienia kontrolera kinematycznego

k_a – stała elektryczna silnika pomnożona przez przełożenie przekładni

k_b – stała mechaniczna pomnożona przez przełożenie przekładni

$k_v, k_\omega, l_v, l_\omega$ – wzmocnienia kontrolera dynamicznego

m – masa robota

$q_{di}(t)$ – wektor współrzędnych konfiguracyjnych zadanych i-tego robota

$q_i(t)$ – wektor współrzędnych konfiguracyjnych (pozycji i orientacji) rzeczywistych i-tego robota

r – promień kół napędowych robota

$u_1(t), u_2(t)$ – wzorcowe sygnały sterujące generatorem trajektorii wzorcowej dla grupy

$u_{dv}(t), u_{d\omega}(t)$ – sygnały sterujące kontrolera dynamicznego

$u_i(t)$ – wektor sterowań i-tego robota w grupie o strukturze rozproszonej

$u_{kv}(t), u_{k\omega}(t)$ – sygnały sterujące kontrolera kinematycznego

t – czas

$v_{di}(t), \omega_{di}(t)$ – wzorcowa prędkość liniowa i kątowa i-tego robota

$v_p^s(t), v_l^s(t)$ – prędkości poślizgu wzdłużnego odpowiednio koła prawego i lewego

$v_{ri}(t), \omega_{ri}(t)$ – rzeczywista prędkość liniowa i kątowa robot

$v_x(t), v_y(t)$ – prędkość wzdłużna i boczna robota w modelu dynamicznym

$x(t), y(t)$ – współrzędne położenia robota w globalnym układzie współrzędnych

x_{Gi}, y_{Gi} – współrzędne punktu, za którym podąża i-ty robot w metodzie Śledzenia Lidera

x_L, y_L – współrzędne położenia lidera w metodzie Śledzenia Lidera
 $x_d(t), y_d(t)$ – współrzędne położenia wzorcowego robota
 x_{iF}, y_{iF} – odległości i-tego robota od wirtualnej struktury
 $x_i^{vc}(t), y_i^{vc}(t)$ – współrzędne położenia wirtualnej struktury
 $x_r(t), y_r(t)$ – współrzędne położenia rzeczywistego robota
 $C_{vc}(t)$ – położenie środka geometrycznego wirtualnej struktury
 G – środek obrotu robota
 H – wektor stanu grupy o strukturze rozproszonej
 F_{rlx}, F_{rly} – siła wzdłużna i poprzeczna koła lewego
 F_{rrx}, F_{rry} – siła wzdłużna i poprzeczna koła prawego
 I_z – moment bezwładności względem osi przechodzącej przez punkt G
 L_i – odległość i-tego robota od lidera w metodzie Śledzenia Lidera
 $R_{1x}, R_{1y}, R_{3x}, R_{3y}$ – reakcje w przegubach palety transportowej
 R_a – rezystancja elektryczna uzwojeń silników
 $V_p(t), V_l(t)$ – napięcia zasilania odpowiednio silnika prawego i lewego
 T – współczynnik korelacji
 α_1, α_2 – kąty nachylenia belek zewnętrznych palety
 α_i, k_{ij} – współczynniki wzmocnienia kontrolera grupy
 β_L – orientacja lidera w metodzie Śledzenia Lidera
 γ – współczynnik wzmocnienia estymatora stanu wirtualnej struktury
 $\theta(t)$ – kąt orientacji robota w globalnym układzie współrzędnych
 $\theta_d(t)$ – kąt orientacji wzorcowej robota
 $\theta_r(t)$ – kąt orientacji rzeczywistej robota
 $\theta_i^{vc}(t)$ – kąt orientacji wirtualnej struktury
 $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ – parametry modelu dynamicznego robota
 $\xi_i(t)$ – wektor stanu robota w odniesieniu do kontrolera grupy
 $\xi^{vc}(t)$ – wektor stanu wirtualnej struktury w odniesieniu do kontrolera grupy
 φ_A, φ_B - kąty obrotu własnego kół A i B robota
 $\omega(t)$ – prędkość kątowna robota w modelu kinematycznym i dynamicznym
 $\omega_p(t), \omega_l(t)$ – prędkości kątowe odpowiednio koła prawego i lewego
 Δt_i – opóźnienie czasowe sygnałów sterujących i-tego robota
 $\Delta x_R, \Delta y_R$ – odległości pomiędzy robotami w grupie w odniesieniu do układu współrzędnych grupy

WSTĘP

Według standardów Przemysłu 4.0, transport wewnętrzny staje się jednym z najważniejszych obszarów podlegających przemianom. Z uwagi na coraz bardziej ograniczone zasoby siły roboczej oraz rosnące koszty transportu produktów w procesie produkcyjnym transport jest rozpatrywany w kategorii strat. Pojawia się zatem konieczność minimalizacji kosztów związanych z wewnętrznym transportem produktów polegająca na skróceniu tras i czasu przemieszczania produktu. Jednym z rozwiązań tego problemu jest wprowadzenie autonomicznych robotów transportowych, które w sposób dokładny, szybki i bezpieczny wykonają powierzone im zadania. Zautomatyzowane pojazdy są używane głównie do obsługi operacji przewozu materiałów w magazynie, na produkcji oraz na styku tych obszarów. Ponieważ transport oparty na robotach ułatwia i optymalizuje pracę osób zatrudnionych w fabrykach oraz jest w stanie skrócić czas produkcji nawet o jedną czwartą [69] coraz częściej i zgodnie z zamysłem Przemysłu 4.0 na liniach produkcyjnych pojawiają się roboty typu AGV (ang. Automated Guide Vehicle) lub AMR (ang. Autonomous Mobile Robot), [48], [45] do przewożenia i zarządzania przepływem produktów. Producenci robotów mobilnych dedykowanych dla przemysłu wprowadzili także specyficzne nazwy różnych rodzajów produkowanych maszyn, podkreślających ich właściwości. Technologia AGV dotyczy pojazdów automatycznych poruszających się wewnątrz budynku po fizycznie wyznaczonych ścieżkach, wymagających instalacji linii lub punktów nawigacyjnych. Narzuca to konieczność poruszania się wyłącznie po wyznaczonej trasie, a w przypadku napotkania przeszkody wymaga zatrzymania się i oczekiwania na jej usunięcie. Zmiana trasy wymaga zmiany linii nawigacyjnej i pozostałych oznaczeń. Rozwiązanie typu AMR obejmuje autonomiczne pojazdy, często nazywane inteligentnymi, wyposażone w skanery 2D lub 3D, poruszające się na podstawie samodzielnie utworzonej mapy po samodzielnie wyznaczonej trasie pomiędzy punktami docelowymi, zdolne do omijania przeszkód.

Trend rozwojowy w zakresie robotyzacji procesów transportowych ukierunkowany jest na zastosowanie do realizacji skomplikowanego zadania grupy współpracujących robotów. Szczególną uwagę przykuwają grupy robotów o strukturze rozproszonej, powszechnie stosowane np. w zadaniach transportu ładunków wielkogabarytowych, w których formacja grupy musi zachować zdefiniowany szyk geometryczny bez możliwości zmiany pozycji i kolejności robotów w szyku. Ogromną zaletą zrobotyzowanych systemów rozproszonych jest możliwość ciągłej pracy w przypadku uszkodzenia któregoś elementu systemu. Ze względu na powszechne stosowanie w robotyce mobilnej komunikacji opartej o sieci bezprzewodowe,

transmisja danych pomiędzy robotami grupy rozproszonej jest szczególnie narażona na zakłócenia zewnętrzne takie jak szum elektromagnetyczny pochodzący od innych urządzeń elektronicznych, chwilowy lub całkowity zanik sygnału, niepożądane dane wysyłane do sieci przez uszkodzone systemy innych urządzeń lub wadliwe układy zasilania urządzeń sieciowych. Rozbudowana architektura grup rozproszonych wymusza konieczność przetwarzania bardzo dużej ilości informacji w dynamicznie zmieniającym się środowisku pracy. Pojawiające się błędy i opóźnienia transmisji sygnałów w rozproszonej grupie robotów skutkują błędnym osiągnięciem zadanej pozycji przez poszczególne roboty, a co za tym idzie błędne osiągnięcie pozycji grupy robotów oraz brak zachowania zdefiniowanego szyku. Dokładność zachowania szyku w trakcie ruchu i dokładność końcowego położenia grupy po wykonaniu zadania ma decydujący wpływ na prawidłowy załadunek i rozładunek transportowanego przez grupę towaru. W związku z tym rozwiązanie problemów, o których mowa powyżej, jest kluczowym elementem w procesie projektowania algorytmu sterowania grupą robotów o strukturze rozproszonej oraz stanowi aktualny i ważny kierunek prowadzenia prac badawczych.

Zagadnieniami związanymi z robotami mobilnymi interesuję się i zajmuję od 2013. Pierwsze projekty, analizy i badania związane z robotyką mobilną popełniłem podczas realizacji prac dyplomowych - inżynierskiej i magisterskiej. Dodatkowo swoją wiedzę inżynierską pogłębiałem na praktykach i stażach w Przemysłowym Instytucie Automatyki i Pomiarów PIAP oraz aktualnie będąc pracownikiem Zakładu Mechatroniki w Instytucie Techniki Raketowej i Mechatroniki WML WAT. Prace badawcze z zakresu robotyki w Zakładzie Mechatroniki prowadzone są od dłuższego czasu w postaci projektów badawczo-rozwojowych, uczelnianych grantów badawczych i w ramach rozwoju indywidualnego – w moim przypadku ze szczególnym ukierunkowaniem na roboty mobilne.

Badania dotyczące pojedynczego robota są szeroko spotykane w literaturze jednak nieustannie stanowią aktualny kierunek rozwoju. Podążając za trendem rozwojowym robotyki mobilnej w przemyśle, który ukierunkowany jest na integrację robotów w grupy, swoją uwagę postanowiłem skupić na aspektach sterowania właśnie w tym obszarze. Dodatkowo w instytucie była już realizowana praca doktorska ukierunkowana na sterowanie grupą podążającą za liderem pt. „Leader-Follower Formation Control with Dynamic Leader Changing of Nonholonomic Mobile Robots Under Robot Operating System”. Zapoznając się z metodyką sterowania grupą o strukturze hierarchicznej, przedstawioną we wspomnianej pracy poznałem wady i zalety tego typu sterowania. Aspekty sterowania grupą hierarchiczną postanowiłem porównać z aspektami i problemami sterowania grupą o strukturze rozproszonej. Dodatkowo wady grupy robotów o strukturze hierarchicznej poznałem w praktyce podczas

Międzynarodowych Targów Innowacyjnych Rozwiązań Przemysłowych „Warsaw Industry Week”, gdzie nie udało się uruchomić stanowiska demonstrującego ruch robotów w grupie o strukturze hierarchicznej z prozaicznego powodu – występujących w hali zakłóceń komunikacji Wi-Fi. Zakłócenia wynikały z bardzo dużej ilości sieci uruchomionych na targach i bardzo dużej ilości urządzeń do tych sieci podłączonych. Zastosowany algorytm sterowania nie był odporny na wspomniane błędy. Zauważyłem wtedy potencjał zastosowania grupy robotów o strukturze rozproszonej, uwzględniającej w algorytmie sterowania rozwiązanie wyżej wymienionych problemów, która byłaby w stanie realizować zadania w opisanym środowisku pracy.

Elementem motywującym do podjęcia omawianego tematu były moje zainteresowania dotyczące algorytmów sterowania robotami i grupami robotów, współpracy robotów mobilnych z robotami przemysłowymi (co wymaga wysokiej precyzji pozycjonowania) oraz chęć kontynuacji i rozwoju badań wcześniej prowadzonych przez współpracowników. Wiedzę i doświadczenie w zakresie robotyki mobilnej chciałbym dalej pogłębiać wzbogacając ją i rozwijając aspekty naukowe poruszone w trakcie przygotowywania swojej rozprawy doktorskiej i dalszej pracy zawodowej. Każde postawione przede mną zadanie z zakresu robotyki mobilnej traktuję jako wyzwanie, które podejmuję z przyjemnością.

Celem mojej rozprawy doktorskiej było opracowanie algorytmu synchronicznego sterowania grupą robotów mobilnych o strukturze rozproszonej. Na wstępie do realizacji celu postawiłem tezę, iż **możliwe jest precyzyjne sterowanie grupą robotów mobilnych o strukturze rozproszonej z wykorzystaniem synchronizacji czasowej.**

Proponowane autorskie rozwiązanie jest nowym podejściem do zagadnień minimalizacji uchybów zadanych odległości pomiędzy robotami, powstających podczas sterowania grupą robotów o strukturze rozproszonej. Problem ten jest istotny zarówno z teoretycznego jak i praktycznego punktu widzenia, w związku z rosnącą liczbą autonomicznych platform mobilnych. Proponowaną metodą rozwiązania przedstawionego problemu jest wykorzystanie dynamicznej synchronizacji czasu dla grupy robotów jak również algorytmów sterujących poszczególnymi robotami. Nowość metodyki badawczej polega na połączeniu zagadnień modelowania matematycznego grupy robotów z uwzględnieniem więzów kinematycznych, reonomicznych (jawnie zależnych od czasu) oraz występujących opóźnień (otrzymanych z badań doświadczalnych na laboratoryjnych robotach mobilnych), w celu opracowania algorytmu synchronicznego sterowania grupą robotów mobilnych o strukturze rozproszonej, optymalnego z punktu widzenia funkcji celu, jaką jest minimalizacja uchybu.

Na drodze do realizacji celu rozprawy postawiłem pięć zadań cząstkowych, w których zostały ujęte zadania szczegółowe:

1. Analiza literaturowa algorytmów sterowania grupą robotów mobilnych z uwzględnieniem ich kinematyki i dynamiki;
 - Przegląd i analiza literatury pod kątem modelowania kinematyki i dynamiki oraz sterowania robotów mobilnych, modelowania ruchu;
 - Przegląd i analiza literatury pod kątem zagadnień dotyczących grup robotów mobilnych;
 - Przegląd i analiza literatury pod kątem metod sterowania ruchem grupy robotów mobilnych;
 - Przegląd i analiza literatury pod kątem metod synchronizacji jednostek w grupie robotów;
2. Opracowanie koncepcji synchronicznego sterowania grupą robotów mobilnych o strukturze rozproszonej;
 - Wyznaczenie teoretycznego modelu kinematycznego i dynamicznego dla zaproponowanej konstrukcji robota mobilnego;
 - Opracowanie koncepcji sterowania grupą robotów o strukturze rozproszonej;
 - Identyfikacja zakłóceń mających wpływ na sterowanie grupą robotów mobilnych;
 - Zbadanie wpływu zakłóceń i opóźnień czasowych na ruch grupy robotów testowych o strukturze rozproszonej w wybranym środowisku programistycznym;
3. Opracowanie modelu kinematycznego i dynamicznego grupy robotów mobilnych o strukturze rozproszonej;
 - Wyznaczenie ogólnego modelu kinematycznego grupy robotów mobilnych o strukturze rozproszonej;
 - Wyznaczenie ogólnego modelu dynamicznego grupy robotów mobilnych o strukturze rozproszonej;
4. Opracowanie algorytmu synchronicznego sterowania grupą robotów mobilnych o strukturze rozproszonej z uwzględnieniem ich kinematyki i dynamiki;
 - Opracowanie algorytmu synchronicznego sterowania grupą robotów o strukturze rozproszonej;
 - Implementacja algorytmu w Robot Operating System;
5. Weryfikacja algorytmu na podstawie badań teoretycznych i doświadczalnych.

Rozdział 1

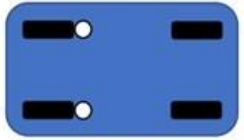
ANALIZA LITERATURY

Słowo robot oznacza w języku czeskim robotnika i pierwszy raz zostało użyte (w odniesieniu do maszyn) w 1920 roku. Robot jest to urządzenie techniczne przeznaczone do realizacji niektórych funkcji manipulacyjnych i lokomocyjnych człowieka, mające pewien poziom inteligencji maszynowej [12]. W odniesieniu do robotyki mobilnej słowo robot oznacza mobilną platformę, której cechą charakterystyczną jest brak operatora znajdującego się na platformie. Sterowanie najczęściej realizowane jest zdalnie, teleoperacyjnie lub autonomicznie [97]. Spektrum zastosowania robotów mobilnych w przemyśle jest bardzo szerokie i obejmuje szereg zadań, takich jak inspekcja [107], [9], [60] mapowanie przestrzeni [2], [110], prace inżynieryjno-montażowe [84], akcje ratunkowe [130], terminowa dostawa części do linii produkcyjnej [42], [70] i transport całodobowy [109], [106]. Pierwsze wózki automatyczne realizujące jazdę bez operatora pojawiły się w latach 50. XX wieku. Były to pojedyncze wdrożenia przede wszystkim w branży motoryzacyjnej. Z czasem zalety płynące z wdrożeń AGV były coraz bardziej dostrzegane i automatyczne wózki zaczęły pojawiać się również w innych branżach, zwłaszcza w zastosowaniach związanych z logistyką wewnątrzzakładową. Od tego czasu technologia robotów AGV dynamicznie się rozwija na co wskazuje fakt istnienia kilku typów maszyn AGV [45]. Wszystkie pojazdy napędzane przeznaczone do automatycznego transportu ładunków są objęte normą ISO 3691-4:2020 i nazywane wózkami jezdniowymi bez operatora (ang. driveless industrial truck).

Podstawową cechą charakterystyczną robotów AGV jest ich układ jezdny. Na Rys. 1.1 przedstawiono kilka podstawowych układów jezdnych stosowanych w mobilnych robotach przemysłowych. Można wyróżnić:

- układy trójkołowe (dwa niezależnie napędzane koła sterowane różnicowo, jedno nienapędzane koło samonastawne),
- układy czterokołowe
 - dwa niezależnie napędzane koła standardowe zamocowane z przodu robota i dwa skrętne koła samonastawne zamocowane z tyłu robota,
 - cztery niezależnie sterowane koła szwedzkie,

- układy sześciokołowe (dwa niezależnie napędzane koła standardowe zamocowane na środku ramy robota i cztery skrętne koła samonastawne zamocowane na rogach ramy robota).

Lp.	Konfiguracja układu jezdnego	Opis układu jezdnego
1.		dwa niezależnie napędzane koła standardowe, które mogą być zamocowane z przodu lub z tyłu robota, jedno nienapędzane koło samonastawne lub kula, zamocowana z tyłu lub przodu robota
2.		dwa niezależnie napędzane koła standardowe zamocowane z przodu robota i dwa skrętne koła samonastawne zamocowane z tyłu robota
3.		dwa niezależnie napędzane koła standardowe zamocowane na środku ramy robota i cztery skrętne koła samonastawne zamocowane na rogach ramy robota
4.		cztery niezależnie sterowane koła szwedzkie

Legenda:

 standardowe koło napędzane

 koło szwedzkie

 samonastawne koło podporowe

 podwozie robota

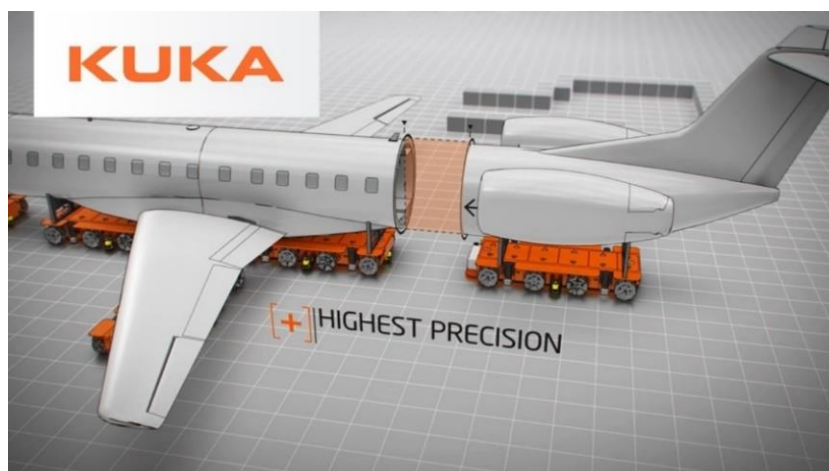
Rys. 1.1. Układy jezdne stosowane w kołowych robotach mobilnych [48]

Obserwując tendencję rozwojową robotyki mobilnej można zauważyć, że trend rozwojowy nie skupia się wyłącznie na projektowaniu i budowaniu maszyn dedykowanych do wykonywania konkretnych zadań, lecz na integracji w grupy robotów. Grupą robotów nazywa się zespół robotów o takich samych lub różnych właściwościach funkcjonalnych i jezdnych, które mogą ze sobą współpracować oraz wymieniać informacje w celu wspólnej realizacji zadania. W przypadku niektórych rodzajów zadań, takich jak np. eksploracja terenu, rozminowywanie terenu lub transport wielkogabarytowy i montaż (biorąc pod uwagę użyteczność cywilną) użycie grupy robotów może skrócić wymagany czas, zminimalizować ryzyko utraty zdrowia lub życia człowieka i zwiększyć szanse na powodzenie wykonania zadania. Różny charakter wykonywanych przez roboty zadań jak i zróżnicowane środowisko (typu indoor- środowisko pracy wewnątrz budynków, typu outdoor- środowisko na przestrzeni otwartej) zachodzi konieczność integracji w grupie robotów o różnej budowie, właściwościach jezdnych i funkcjonalnych (kinematyce i dynamice) [85]. Dodatkowo rozpatrując środowisko indoor roboty mobilne z powodzeniem mogą wspierać i uzupełniać pracę robotów

przemysłowych na liniach produkcyjnych w fabrykach [47]. Do zadań takich można zaliczyć procesy inspekcyjne w zróżnicowanym terenie, manipulacja bądź podejmowanie niebezpiecznego ładunku.



Rys. 1.2. System transportowy KUKA omniMove w fabryce pociągów SIEMENS [58]



Rys. 1.3. Wizualizacja procesu montażu kadłuba samolotu z wykorzystaniem systemu KUKA omniMove [92]

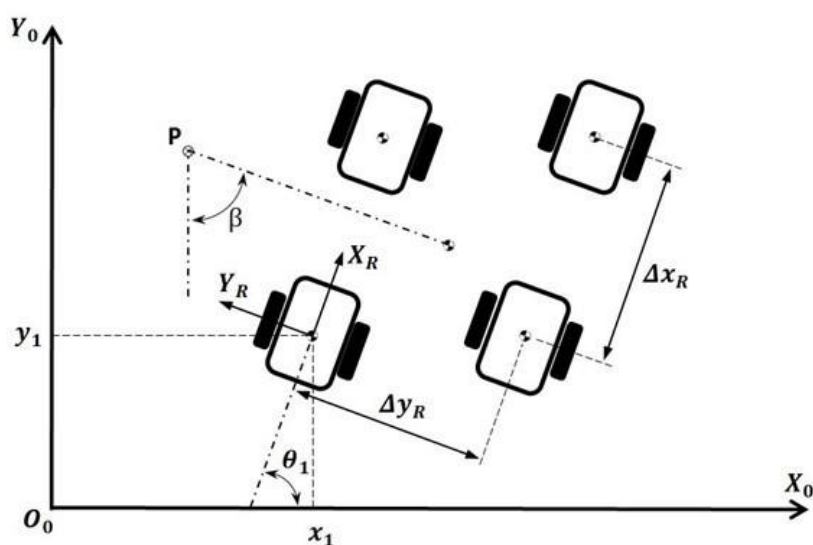
Rozpatrując realizowane przez roboty zadania dotyczące transportu (Rys. 1.2) i montażu (Rys. 1.3) (transport ładunków o dużej masie, dużych gabarytach lub/i nieregularnym kształcie), konieczna jest precyzyjna koordynacja współpracy pomiędzy nimi [57]. Ponadto obszar montażowy jest często zatłoczony, co utrudnia manipulowanie ładunkami jak i zwiększa ryzyko kolizji. Projektowanie, budowa i użytkowanie kilku prostych robotów są łatwiejsze i tańsze niż ten sam proces dotyczący jednego skomplikowanego, multifunkcyjnego urządzenia.

Koncepcja grupy robotów obejmuje sterowanie co najmniej trzema robotami. Trudność sterowania grupą polega na realizacji współdziałania robotów we wspólnym środowisku czego następstwem jest osiągnięcie wspólnego celu [4], [126]. Wyzwania inżynierskie towarzyszące

budowie takiego systemu dotyczą opracowania rozwiązań związanych z wzajemną komunikacją, nawigacją oraz synchronizacją robotów przy sterowaniu autonomicznym [62]. Roboty współpracujące ze sobą narażone są na wzajemne kolizje jak i kolizje z przeszkodami wchodzącymi w skład otoczenia. Sposobem eliminacji kolizji z otoczeniem jest wyposażenie robotów w skanery laserowe, kamery, czujniki zbliżeniowe oraz inne sensory pozwalające na uzyskanie informacji o otoczeniu, które to informacje są następnie przetwarzane przez algorytmy wykrywania przeszkód oraz nawigacji. Rozwiązaniem wyżej wymienionych problemów jest wyposażenie robotów w taki sam system operacyjny i implementacja algorytmów, które przy wprowadzeniu minimalnych modyfikacji związanych z kinematyką i dynamiką danego robota, w łatwy i efektywny sposób umożliwią integrację i sterowanie grupy robotów.

Sterowanie formacją robotów jest ściśle powiązane z rodzajem realizowanego zadania. Wyróżnia się dwie struktury opisujące metodologię sterowania grupą robotów: strukturę centralną (hierarchiczną) oraz strukturę rozproszoną (zdecentralizowaną). Architektura systemów sterowania powiązana jest również z przyjętym rozwiązaniem planowania ruchu czy zadań.

Struktura centralna charakteryzuje się pojedynczym centrum zarządzania informacjami, które przetwarza wszystkie informacje potrzebne do osiągnięcia pożądanego efektów sterowania - realizacji wskazanego zadania. Pozwala to na stosowanie mniej skomplikowanej strategii sterowania formacją, a co za tym idzie łatwiejszą implementację sprzętową. Struktura centralna przedstawiona na Rys. 1.4 polega na wyznaczeniu zadania i przydzieleniu do niego mastera - robota dowodzącego, który oblicza trajektorię i wybiera optymalny sposób jej realizacji.



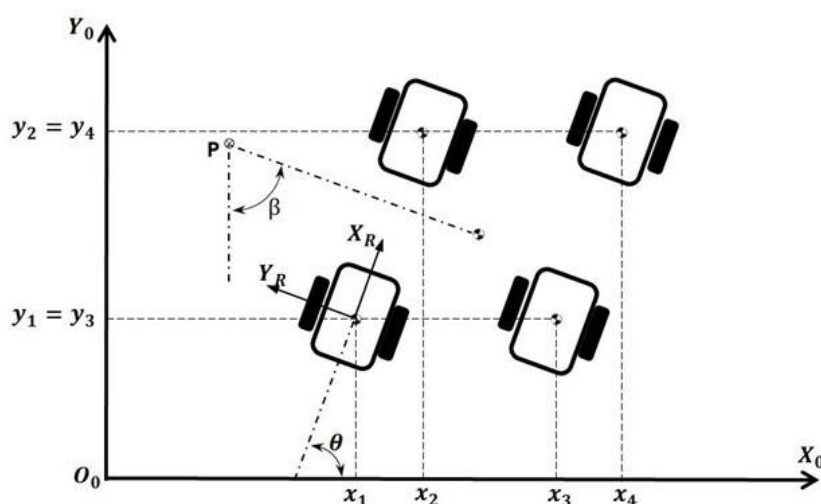
Rys. 1.4. Pozycjonowanie robotów w grupie o strukturze centralnej

Pozycja mastera (x_1, y_1 na Rys. 1.4) jest punktem odniesienia do obliczenia pozycji pozostałych robotów ($x_1 + \Delta x_R, y_1 + \Delta y_R$ na Rys. 1.4) wchodzących w skład grupy. Współrzędne punktów ruchu robotów obliczane są poprzez dodanie przesunięcia Δx_R i Δy_R w wybranym kierunku od pozycji odniesienia robota mastera [12], [44], [116], [19], [98].

Struktura centralna jest charakterystyczna dla systemów, w których planowanie ruchów wynika z narzuconej przez operatora trajektorii lub wygenerowanej na podstawie przyjętych założeń czy bazy wiedzy. Rozwiązanie wykorzystuje trajektorię ruchu lidera narzuconą przez operatora lub pochodzącą z układu sterowania nadrzędnego. Algorytmy sterowania grupą robotów o strukturze centralnej wymagają jednak dużej mocy obliczeniowej jednostki, przepływu bardzo dużej ilości informacji w krótkim czasie, a ich działanie uzależnione jest od działania sterownika centralnego.

Szczególnie ciekawym obiektem rozważań jest struktura rozproszona grupy robotów. W grupie o strukturze rozproszonej poszczególne roboty, chociaż tworzą jedną sieć, w sposób niezależny od siebie interpretują docierające sygnały i w odpowiedni sposób zmieniają swoje działanie. Na podstawie analizy literatury można zauważyć, że za stosowaniem grupy zdecentralizowanej w niektórych zadaniach przemawia wiele zalet. Podstawową zaletą użytkowania grupy robotów o strukturze rozproszonej jest duża odporność na błędy całego systemu, ponieważ w przypadku awarii jednego robota można go wykluczyć z realizacji zadania bez konieczności wyłączania całej grupy. Grupa o takiej strukturze szczególnie nadaje się do wykorzystania w transporcie wielkogabarytowym jak również do współpracy z robotami przemysłowymi na liniach produkcyjnych [85], [57], [80], [99], [21].

Relacje geometryczne pomiędzy robotami w grupie o strukturze rozproszonej przedstawiono na Rys. 1.5.



Rys. 1.5. Pozycjonowanie robotów w grupie o strukturze rozproszonej

Współrzędne położenia poszczególnych robotów x_n , y_n wchodzących w skład grupy rozproszonej obliczane są niezależnie od siebie, lecz względem wspólnego, nieruchomego układu odniesienia $O_0X_0Y_0$. Istotnym w tym przypadku jest to, że każdy robot stanowi część składową grupy, realizującą w sposób spójny określone zadanie. Aby móc to zrealizować konieczne jest na wstępie podanie warunków początkowych konfiguracji – rozmieszczenia grupy robotów. Dopiero tak skonfigurowana grupa może być sterowana w celu realizacji konkretnych zadań. Należy przy tym zaznaczyć, iż wspomniana konfiguracja robotów ma charakter umowny i może być zmieniana w trakcie realizacji zadania, jeśli zajdzie taka konieczność, dokonując segmentacji na mniejsze podgrupy lub też poprzez połączenie kilku podgrup w jedną większą. Niezależnie od konfiguracji i rekonfiguracji grupy, realizacja każdego zadania musi odbywać się w sposób synchroniczny przez wszystkie roboty należące do grupy. Sterowanie grupą o strukturze rozproszonej opiera się na wysyłaniu komend, które są interpretowane indywidualnie przez każdego robota. Dzięki temu istnieje możliwość tworzenia zaawansowanych instalacji transportowych, pozwalających na realizację montażu i demontażu elementów o znacznych gabarytach i masie z wysoką precyzją.

1.1. Przegląd algorytmów sterowania grupami robotów mobilnych

Realizacja zadań wymagających współpracy wielu robotów wymusza konieczność implementacji w systemie sterowania kontrolera grupy, którego zadaniem jest utrzymanie zadanej formacji grupy, polegającym na minimalizacji błędów wzajemnego położenia i orientacji robotów, przy zachowaniu trajektorii wzorcowej grupy.

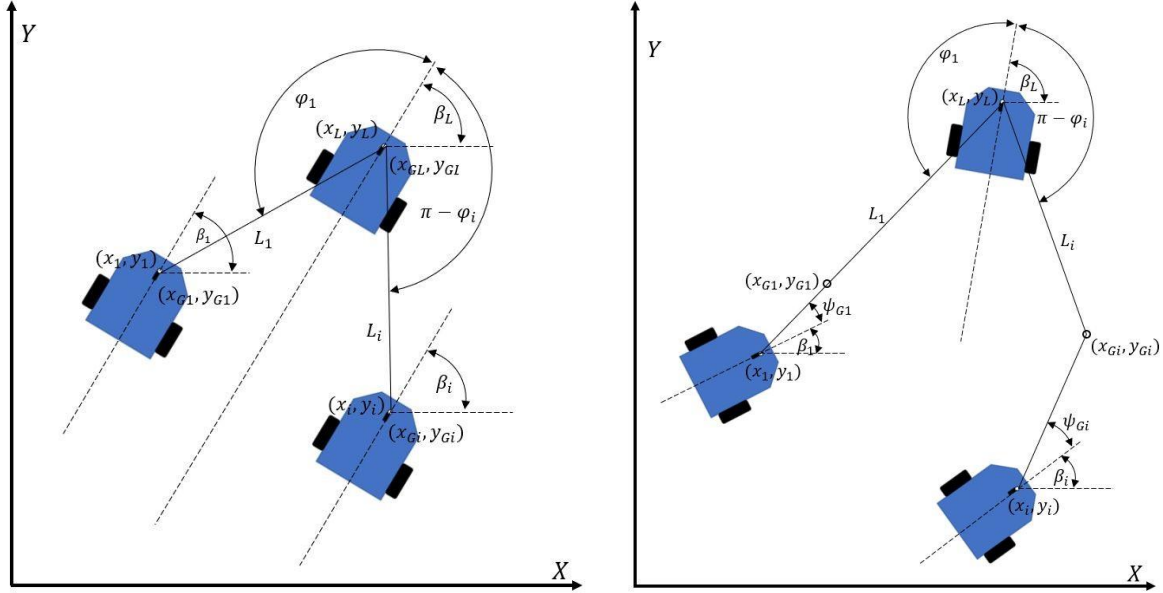
Nawiązując do opisanego w rozdziale 2 podziału struktur grup robotów mobilnych na centralne i rozproszone, jako przykład sterowania grupą o strukturze centralnej przedstawia się algorytm Śledzenia Lidera. W przypadku grupy o strukturze rozproszonej najpopularniejszym rozwiązaniem jest algorytm Wirtualnej Struktury i algorytm Teorii Grafów oraz mniej rozpowszechnione: algorytm Behawioralny i algorytm Sztucznych Pól Potencjałowych.

1.1.1. Algorytm Śledzenia Lidera

W zadaniach patrolowych, bojowych i inspekcyjnych dopuszcza się zmianę położenia robotów w szyku przy jego jednoczesnym zachowaniu. Uzyskuje się to wykorzystując algorytmy sterowania polegające na śledzeniu pozycji lidera [12], [22], [44], [116], [119].

Założenia dla algorytmu Śledzenia Lidera:

1. zdefiniowany kształt formacji,
2. zdefiniowany lider grupy,
3. zdefiniowane początkowe wartości położenia i orientacji robotów.



Rys. 1.6. Idea sterowania metodą Śledzenia Lidera

Idea sterowania metodą Śledzenia Lidera została przedstawiona na Rys. 1.6. Przyjęto współrzędne charakterystycznego punktu robota lidera jako x_L, y_L oraz orientację jako β_L . Zdefiniowano kształt formacji w odniesieniu do lidera jako odległości L_i oraz kąty odchylenia φ_i kolejnych robotów od lidera. Ideą tego algorytmu jest w kolejnych krokach iteracyjnych minimalizowanie różnicy pomiędzy zadeklarowaną a rzeczywistą odległością robotów od lidera.

Analizując Rys. 1.6 oraz prace [15], [10], [11], można zauważyć, że w algorytmie Śledzenia Lidera każdy robot w grupie poza liderem podąża za punktem x_{Gi}, y_{Gi} . Położenie robotów wyznaczone jest na podstawie położenia i orientacji lidera zgodnie z zależnością:

$$\begin{aligned} x_{Gi} &= x_L - L_i \sin \varphi_i \sin \beta_L + L_i \cos \varphi_i \sin \beta_L \\ x_{Gi} &= y_L + L_i \cos \varphi_i \sin \beta_L + L_i \sin \varphi_i \cos \beta_L \end{aligned} \quad (1.1)$$

Prezentowane podejście pozwala na zachowanie kształtu formacji podczas jej przemieszczania (przemieszczanie lidera) przy dowolnej orientacji robotów. Zadeklarowane w systemie sterowania odległości, na którą roboty mogą się do siebie zbliżyć (jest to także parametr określający dopuszczalne odchylenie położenia podczas ruchu grupy) jest sposobem na uniknięcie wzajemnych kolizji. W celu ustabilizowania formacji konieczne jest

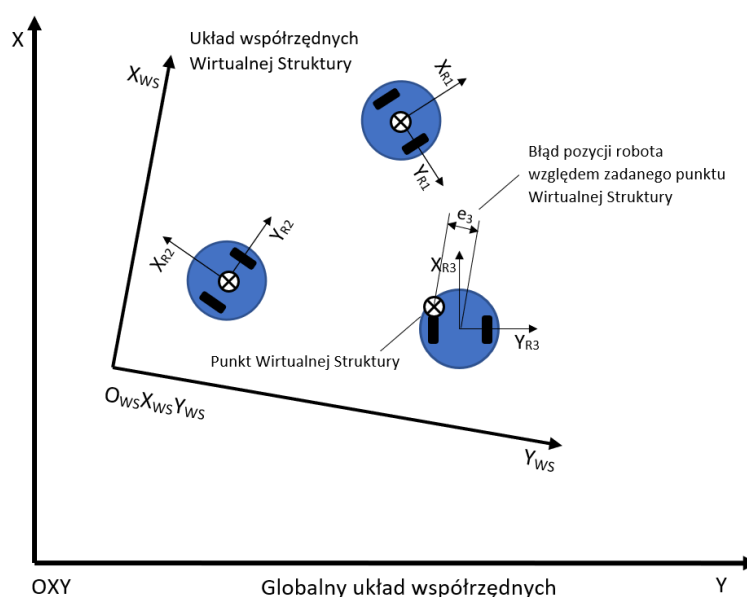
wprowadzenie rozwiązań takich jak np. sterowanie za pomocą sprzężenia zwrotnego sygnału wyjściowego i estymacji stanu układu za pomocą rozszerzonego filtru Kalmana przedstawionych w [73].

Zaletą sterowania metodą Śledzenia Lidera jest łatwość kontrolowania wielu robotów w pożądanej formacji. Metoda doskonale nadaje się do tworzenia opisu matematycznego formacji. Wadą jest trudność w precyzyjnym utrzymaniu zadanych odległości pomiędzy robotami [91].

1.1.2. Algorytm Wirtualnej Struktury

W przypadku innego typu zadań niż eksploracja, gdzie wymagana jest wysoka precyzja pozycjonowania robotów w grupie (np. transport wielkogabarytowy, montaż czy współpraca z robotami przemysłowymi) formacja musi zachować zdefiniowany szyk geometryczny bez możliwości zmiany kolejności robotów. Tego typu zadania są realizowane z wykorzystaniem algorytmu Wirtualnej Struktury [50], [63], [78], [61]. Biorąc pod uwagę grupę złożoną z wielu robotów mobilnych, rozwiązanie problemu poruszania się w formacji musi jednocześnie spełniać dwa cele: postęp w danym kierunku i utrzymanie zgodności geometrycznej z narzuconą Wirtualną Strukturą (ruchomym układem współrzędnych) [50], [63], [51]. Jeżeli jednak na ruch jednego lub więcej robotów zostaną nałożone błędy lub utracona zostanie wydajność nadążania, robot nie będzie w stanie dopasować się idealnie do pozycji w odniesieniu do Wirtualnej Struktury i kształt formacji zostanie zaburzony.

Ideę ruchu grupy sterowanej za pomocą algorytmu Wirtualnej Struktury przedstawiono na Rys. 1.7.



Rys. 1.7. Idea sterowania w metodzie Wirtualnej Struktury

W tej metodzie roboty zawsze utrzymują swoje pozycje względem układu odniesienia $O_{WS}, X_{WS}Y_{WS}$, który porusza się w przestrzeni. Dzięki temu realizowany jest ruch grupy robotów w przestrzeni (w globalnym układzie współrzędnych OXY) bez zmiany ich wzajemnego położenia oraz położenia względem wspólnego układu odniesienia.

W celu dopasowania pozycji i orientacji robotów do Wirtualnej Struktury należy znaleźć wartości przesunięcia i obrotu T_{ij} zgodnie z wyrażeniem [63]:

$$T_{ij} = Rot(\theta_{ij}) * P(x_{ij}, y_{ij}) \quad (1.2)$$

gdzie:

θ_{ij} - orientacja i-tego robota w j-tej chwili czasu

$[x_{ij}, y_{ij}]$ - wektor translacji i-tego robota w j-tej chwili czasu.

Wektor T_{ij} minimalizuje błąd pomiędzy pozycjami każdego robota i odpowiadającymi im pozycjami zadanymi w Wirtualnej Strukturze.

Błąd e_i pomiędzy pozycją poszczególnych robotów, a pozycjami wzorcowymi w algorytmie Wirtualnej Struktury opisany jest wzorem [63]:

$$e_i = f(q_i, q_{d_i}) = \sqrt{(q_{d_i} - q_i)^T (q_{d_i} - q_i)} \quad (1.3)$$

gdzie:

q_i - wektor rzeczywistych współrzędnych konfiguracyjnych i-tego robota

q_{d_i} - wektor współrzędnych konfiguracyjnych zadanych i-tego robota

Sumę błędów L^n wszystkich robotów w algorytmie Wirtualnej Struktury można zapisać wyrażeniem [63]:

$$L(e_1, \dots, e_n) = \sum_{i=1}^n e_i = L^n \quad (1.4)$$

Sterowanie w algorytmie Wirtualnej Struktury sprowadza się do znalezienia takiej macierzy T_{ij} , która minimalizuje koszt funkcji opisanej zależnością [63]:

$$G(T_{ij}) = L^n \quad (1.5)$$

W przypadku oceny jakości sterowania metodą Wirtualnej Struktury zgodnie z [50] i [63] należy zwrócić uwagę na fakt, że pomimo utworzenia przez roboty formacji, każdy robot nadal ma zmiany orientacji. Pomimo wysokiej precyzji pozycjonowania algorytm Wirtualnej Struktury narażony jest na błędy śledzenia trajektorii pochodzące z dwóch źródeł: błędu śledzenia całej Wirtualnej Struktury w globalnym układzie współrzędnych i błędu podążania za Wirtualną Strukturą pojedynczych robotów. Oznacza to, że sterowanie grupą robotów przy wykorzystaniu algorytmu Wirtualnej Struktury nie jest optymalne pod względem

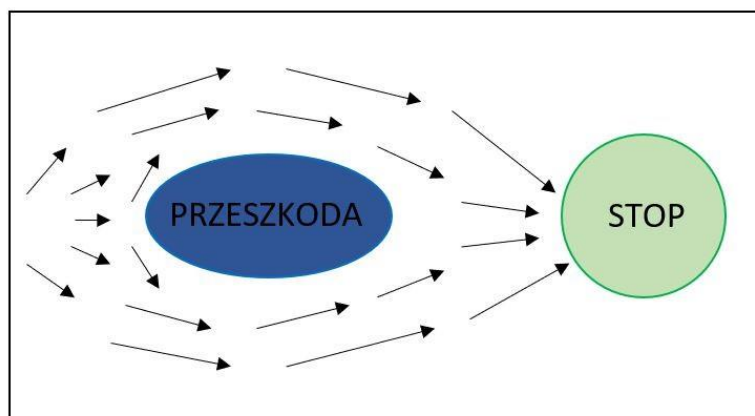
minimalnego czasu wykonania zadania, ponieważ przed rozpoczęciem ruchu grupy, niektóre roboty muszą poczekać aż pozostałe osiągną wymagany stan początkowy.

1.1.3. Algorytm Behawioralny

Metody sterowania formacją robotów, których inspiracją są wzorce pochodzące z obserwacji organizmów żywych, noszą nazwę metod Behawioralnych [3], [8]. Robotyka behawioralna (oparta na zachowaniach) czerpie wzorce z obserwacji zachowań formacyjnych zwierząt stadnych w celu osiągnięcia wspólnej korzyści. Grupując się, zwierzęta łączą swoje zmysły, aby zwiększyć szansę na wykrycie drapieżników lub skuteczniej zdobywać pożywienie. Z obserwacji wynika jednak, że zwierzęta wykazują duże chęci pozostania w grupie, a jednocześnie utrzymują dystans od innych członków grupy zostawiając sobie w pewnym stopniu swobodę. Na tej podstawie badacze robotyki doszli do wniosku, że grupy robotów mogą w podobny sposób korzystać z taktyki tworzenia formacji. Zamiast używać gotowych obliczeń do radzenia sobie z sytuacją, robotyka oparta na zachowaniu opiera się na zdolnościach adaptacyjnych. Postępy te pozwoliły robotom opartym na zachowaniu stać się powszechne w badaniach i gromadzeniu danych. Robot wykorzystuje wszystkie zbierane z czujników informacje, aby stopniowo korygować swoje działania zgodnie ze zmianami w bezpośrednim otoczeniu. Podejście do generowania i utrzymania formacji zgodnie z założeniami BBR (ang. Behaviour-Based Robotics) polega na podziale realizowanego zadania na składowe przypisane dla każdego robota wchodzącego w skład formacji. Przykładami zadań składowych są: omijaj przeszkody, unikaj kolizji, podążaj wzdłuż ścieżki, utrzymuj/dostosuj prędkość, buduj mapę, utrzymuj formację. BBR często popełniają błędy, powtarzają akcje lub chwilowo nie potrafią rozwiązać postawionego zadania, ale wykazują również antropomorficzne podejście do rozwiązywania zadanego problemu (osiągnięcia zadanego położenia, pokonania zadanej trajektorii), którego celem jest sukces realizacji zadania grupowego. Przykładem antropomorficznego podejścia do rozwiązywania problemu jest pozostawienie robota, który utknął za przeszkodą i dalszy ruch pozostałych w celu realizacji zadania grupowego. Zaletą schematów sterowania opartych na zachowaniu jest to, że sprzężenie zwrotne formacji jest niejawnie zintegrowane poprzez sprzężenie wag działań, które zależą od względnych współrzędnych sąsiednich robotów. Podejścia oparte na zachowaniu są przydatne w kierowaniu systemem z wieloma robotami w nieznanym lub dynamicznie zmieniającym się środowisku z wykorzystaniem wyłącznie lokalnych informacji sensorycznych. Trudno jednak opisać dynamikę grupy i zagwarantować stabilność całego systemu.

1.1.4. Algorytm Sztucznych Pól Potencjałowych

Osobną grupę sterowania grupą robotów stanowi metoda oparta na sztucznych funkcjach potencjałowych opisana w pracach [61], [116], [51], [36], [24]. W metodzie tej zakłada się, że przeszkody lokowane w środowisku poruszania się robotów wytwarzają siły odpychające, które odpychają roboty, natomiast punkty docelowe, do których zmierzają roboty wytwarzają siły przyciągające jak przedstawiono na Rys. 1.8.

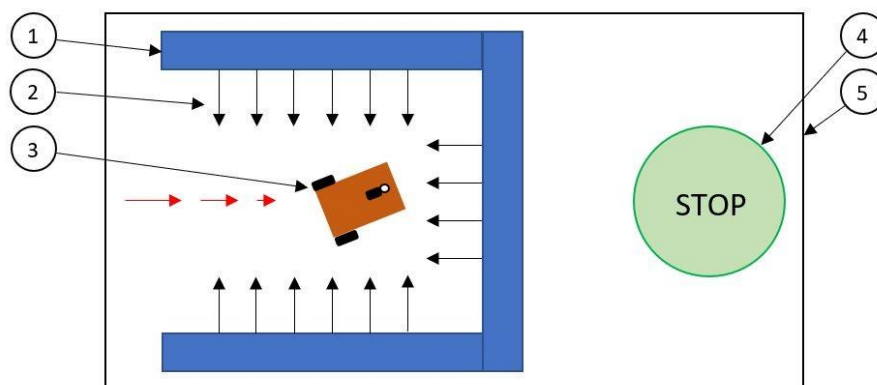


Rys. 1.8. Rozkład wektorów sił wypadkowych działających na robota [48], [24]

Pierwszą koncepcję metody Sztucznych Pól Potencjałowych można znaleźć w pracy Khatiba z 1985 r. [51]. Polega ona na wyborze ogólnej funkcji potencjałowej złożonej z dwóch rodzajów funkcji: przyciągających (APF) i odpychających (RPF). Funkcje APF są formułowane w oparciu o pożądaną odległość pomiędzy robotami i sterują ruchem robotów tak aby utrzymać wymagany kształt formacji. Funkcje RPF są formułowane w oparciu o odległości pomiędzy parami robotów (minimalną odległość na jaką mogą się do siebie zbliżyć). RPF dążą do nieskończoności w przypadku zderzenia się dwóch robotów oraz stopniowo zanikają wraz ze wzrostem odległości pomiędzy robotami. RPF wykorzystuje się do unikania wzajemnych kolizji [36]. Oba opisane wyżej rodzaje funkcji są używane w zastosowaniach praktycznych, aby zapewnić ruch robotów w grupie bez wzajemnych kolizji oraz bez kolizji z przeszkodami. Oznacza to, że w celu poprawnego funkcjonowania algorytmu wartość sił odpychających powinna być mała, gdy robot znajduje się daleko od przeszkody oraz nieskończenie duża w bezpośrednim otoczeniu przeszkody. Siły, które oddziałują na robota są sumowane tworząc siłę wypadkową w danym punkcie przestrzeni. Wektor wypadkowy definiuje nowy kierunek, w którym powinien podążać robot. Zakładając, że punkt materialny, którym opisywany jest każdy robot wchodzący w skład grupy, porusza się na płaszczyźnie, a jego położenie opisane jest poprzez współrzędne x_i, y_i , wypadkowe pole działające na robota jest superpozycją pól przyciągających (pochodzących od punktu

docelowego oraz pozostałych elementów grupy) i pól odpychających (pochodzących od przeszkód). W efekcie robot będzie poruszał się w kierunku, który minimalizuje globalną energię potencjalną.

W celu określenia stopnia realizacji zadania, czyli określenia dokładności osiągnięcia zadanej pozycji, koniecznym jest przyjęcie kryterium w postaci określenia wymaganej wartości siły wypadkowej. Jeżeli wartość siły spadnie poniżej ustalonej granicy, przyjmuje się, że punkt końcowy został osiągnięty.



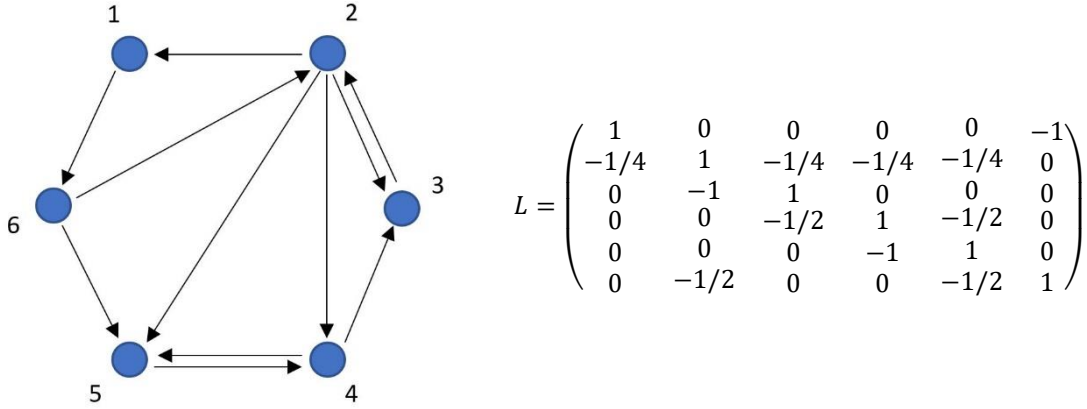
Rys. 1.9. Położenie robota w lokalnym minimum pól potencjałowych [48], [24]
1 – przeszkoda, 2 – wektory sił potencjałowych, 3 – robot, 4 – pozycja docelowa, 5 – granica obszaru roboczego

Algorytm Sztucznych Pól Potencjałowych nie wymaga dużych mocy obliczeniowych procesora, z tego względu doskonale nadaje się do sterowania w czasie rzeczywistym. Ogromną wadą metody Sztucznych Pól Potencjałowych jest prawdopodobieństwo pojawienia się minimum lokalnego - położenia, w którym siła wypadkowa będzie równa zero, a robot nie znajduje się w celu. Wyjście robota z minimum lokalnego jest niemożliwe przy sterowaniu wykorzystującym wyłącznie metodę pól potencjałowych. Zastosowanie dodatkowych metod sterowania wspierających metodę Sztucznych Pól Potencjałowych pozwala na wyjście robota z problematycznej sytuacji. Przykład położenia robota w minimum lokalnym pól potencjałowych przedstawiono na Rys. 1.9.

1.1.5. Algorytm Teorii Grafów

Kolejna metoda sterowania grupą robotów stosuje podejście oparte na Teorii Grafów do projektowania układu sterowania w układzie zamkniętym z pętlą sprzężenia zwrotnego i modelowania wzajemnego połączenia robotów w grupie. Na podstawie [30] graf jest zbiorem wierzchołków (inaczej węzłów) oraz krawędzi łączących pary wierzchołków. W uproszczeniu można przyjąć, że graf jest figurą geometryczną jak przedstawiono na Rys. 1.10.

Jeżeli przyjmuje się, że nie ma różnicy pomiędzy dwoma wierzchołkami związanymi z daną krawędzią wtedy graf określany jest jako nieukierunkowany. Jeżeli przyjmuje się, że krawędź jest skierowana od jednego wierzchołka do drugiego wtedy komunikacja między dwoma robotami w grupie może odbywać się wyłącznie jednokierunkowo.



Rys. 1.10. Przykładowy graf i jego Laplacian [48], [30]

Analizując literaturę pod kątem zastosowania algorytmu Teorii Grafów można zauważyć, że metoda ta jest szeroko rozwijana i często stosowana w przypadku sterowania rozproszonego. W pracy [26] metodę Teorii Grafów zastosowano do rozproszonego sterowania grupą nieholonomicznych robotów kołowych w oparciu o ich modele nieliniowe. W przedstawionym rozwiązaniu roboty tworzą wymagany wzór geometryczny, którego środek geometryczny porusza się po zadanej trajektorii wzorcowej, a informacje o niej dostępne są tylko dla niektórych robotów w grupie. Z kolei w pracy [82] poruszono problem sterowania formacją nieholonomicznych robotów w skończonym czasie. W prezentowanym rozwiązaniu wprowadzono ujednolicony system błędów dla grupy rozproszonej, a następnie sformułowano prawo sterowania dla zadania śledzenia trajektorii przez każdego robota. Zaletą tej metody jest łatwość przedstawienia dowolnej formacji za pomocą wykresu, a co za tym idzie prostota opisu matematycznego. Jednak takie podejście wymaga wymiany bardzo dużej ilości informacji pomiędzy robotami co jest ograniczone własnościami sprzętu komunikacyjnego.

1.1.6. Algorytm Konsensu

Do realizacji sterowania grupą, rozwiązującego opisane we wprowadzeniu problemy, przy występujących zakłóceniach i opóźnieniach transmisji danych jest algorytm Konsensu. Według strategii konsensu jednym ze sposobów kontrolowania formacji grupy jest optymalizacja profilu prędkości dla poszczególnych robotów tak aby utrzymać zadaną formację grupy w trakcie realizacji trajektorii wzorcowej. Rozwiązanie takie przedstawiono w pracy

[67], gdzie autorzy utrzymanie formacji grupy zamodelowali jako zależne od prędkości, a następnie sprowadzili problem koordynacji ruchu do problemu optymalizacji prędkości poszczególnych robotów, a do minimalizacji funkcji celu zastosowali metodę Lingo (ang. Linear interactive and general optimizer).

Innym przykładem są prace [66] i [57], w których zaprezentowano system sterowania grupy robotów poruszającej się w nieznanym środowisku z przeszkodami. W pracach zastosowano dwupoziomową architekturę systemu sterowania - sterowanie lokalne dla każdego robota oraz sterowanie globalne dla utrzymania formacji grupy. Jako sterowanie lokalne, polegające na zaplanowaniu trajektorii dla każdego robota zastosowano zdecentralizowaną metodę planowania Local-Constant-Path Planner. Na poziomie sterowania globalnego ograniczenia kinematyczne grupy i wymianę informacji pomiędzy robotami przedstawiono za pomocą diagramu „Constraint tree”. Następnie, do koordynacji ruchu poszczególnych robotów zaimplementowano metodę sterowania nazwaną Dynamic Priority Strategy.

Wspólną cechą wyżej wymienionych metod jest implementacja regulatorów kinematycznych, w celu osiągnięcia i utrzymania zadanej formacji, opartych o wzajemne śledzenie prędkości poszczególnych robotów grupy. Ponieważ w rzeczywistości na dynamikę pojedynczego robota jak i relacje w grupie może mieć wpływ wiele czynników (opisanych w rozdziale 0) generujących zakłócenia, założenie idealnego śledzenia prędkości nie sprawdza się w praktyce, zwłaszcza w realizacji zadań, wymagających precyzyjnego pozycjonowania tj. transport wielkogabarytowy. Dlatego w tym przypadku należy wdrożyć strategię sterowania grupą z uwzględnieniem kinematyki i dynamiki robotów. Jednak w praktyce nieosiągalna jest doskonała znajomość dynamicznego modelu robota mobilnego, a uzyskanie dokładnych wartości parametrów modelu jest bardzo trudne [86]. Do rozwiązania wyżej przedstawionego problemu stosuje się algorytmy konsensusu. Jeden ze sposobów rozwiązania problemu sterowania grupą robotów o nieznanej dynamice przedstawiono w pracach [86] i [59], w których opisano transformację sterowania formacją do problemu konsensusu. Następnie opracowano regulatory kinematyczne oparte na rozproszonym konsensusie, których sterowanie polega na asymptotycznym zbieganiu się robotów w pożądaną wzór geometryczny, i kontrolowaniu wirtualnego środka grupy tak, by poruszał się po wyznaczonej trajektorii. W algorytmie założono lokalną wymianę informacji pomiędzy robotami - tylko roboty sąsiadujące mają informacje na temat stanu, w którym się aktualnie znajdują. Opisane rozwiązanie zaimplementowano z wykorzystaniem modelu dynamiki robotów, gdzie sygnałami wyjściowymi są momenty obrotowe. Utrzymanie formacji grupy

realizowane jest przy wykorzystaniu identyfikacji online parametrów modelu dynamiki, za pomocą sieci neuronowych.

Kolejnym przykładem sprowadzenia zadania sterowania formacją do problemu konsensusu jest praca [79], w której zaproponowano zdecentralizowany, zmienny w czasie i ciągły kontroler grupy oraz praca [64], w której przedstawiono adaptacyjne algorytmy śledzenia trajektorii dla problemu konsensusu. Dodatkowo zwrócono szczególną uwagę na topologię komunikacji pomiędzy robotami.

Wyżej wymienione prace prezentują podejście sterowania asymptotycznego. W przypadku realizacji precyzyjnego śledzenia trajektorii wzorcowej skuteczniejszym podejściem jest sterowanie w skończonym czasie. Szczegóły tych metod można znaleźć w pracach [83] i [43]. Charakteryzują się one krótkim czasem regulacji i dobrą odpornością na niepewności i zakłócenia.

Podejście realizacji zadania śledzenia trajektorii przez grupę robotów z wykorzystaniem algorytmu konsensusu wymusza przetwarzanie bardzo dużej ilości informacji o stanie robotów w grupie. Wraz z wzrostem ilości robotów w grupie oraz zmiennością parametrów transmisji sygnałów, wydajność sterowanego systemu ma tendencję do zmniejszania się, a nawet do niestabilności systemu. W związku z tym, badanie opóźnień czasowych w układach sterowania grupami robotów przy użyciu algorytmu konsensusu staje się ważnym kierunkiem badawczym w dziedzinie sterowania czego potwierdzenie znajdziemy między innymi w pracach [71], [128] i [31].

1.2. Analiza kontrolerów stosowanych do realizacji zadania śledzenia trajektorii

Sterowanie nieholonomicznymi robotami, czyli takimi, na które nałożone są więzy prędkościowe, jest bardzo ważnym zagadnieniem w robotyce mobilnej od wielu lat. Nieholonomiczne ograniczenia robotów wynikają z ich kinematyki, która wyklucza ruch w niektórych kierunkach. W literaturze spotyka się wiele rozwiązań algorytmów sterowania, których zadanie sprowadza się do śledzenia trajektorii referencyjnej. Częstym rozwiązaniem tego zadania są kontrolery opracowane na podstawie zadania odwrotnego kinematyki np. opisane w pracach [76], [40]. Tego typu kontrolery w bardzo prosty i szybki sposób są implementowane dla robotów, w przypadku których punkt śledzący trajektorię nie jest umieszczony na osi łączącej koła. Kontrolery takie jednak nie mogą być zrealizowane w przypadku robotów o środku obrotu umieszczonym na osi łączącej koła, jak np. TURTLEBOT 2, ponieważ zadanie odwrotne kinematyki jest nierozwiązywalne.

W przypadku robotów o napędzie różnicowym, ze środkiem obrotu umieszczonym na osi łączącej koła, stosowane są kontrolery proporcjonalne, poślizgowe, predykcyjne. Przykład kontrolera proporcjonalnego jest opisany w pracach [52] i [94] gdzie autorzy przedstawiają kontrolery z pętlą sprzężenia zwrotnego w zadaniach śledzenia trajektorii przez nieholonomiczne roboty kołowe. Autorzy zastosowali nowatorską metodę optymalizacji Teaching-Learning-Based Optimization (TLBO), która jest wzbogaceniem tradycyjnego kontrolera PD. Sterowanie poślizgowe przedstawiono w pracach [49] i [53] gdzie na podstawie równań kinematycznych robota zaprojektowano system sterowania, w którym robot jest sterowany tak, aby podążał za dowolną ścieżką odniesienia z predefiniowanym profilem prędkości. Autorzy wykazali stabilność zaproponowanego algorytmu i odporność na szумы wejściowe i wyjściowe oraz inne zakłócenia. Działanie opracowanego kontrolera przedstawiono na prostym przykładzie trajektorii, jednak dla bardziej ogólnych zastosowań należy rozwinąć przedstawioną metodę sterowania. Sterowanie predykcyjne przedstawiono w pracach [115] i [112] gdzie zastosowano strategię sterowania predykcyjnego opartą na nieliniowym modelu robota kosmicznego w zadaniach śledzenia trajektorii w czasie rzeczywistym i unikania kolizji. Do optymalizacji sterowania autorzy zastosowali metody quadratic programming (QP).

Sterowanie robotami uwzględniające wyłącznie własności kinematyczne robotów jest bardzo popularne w literaturze ze względu na prostotę implementacji. W przypadku bardzo precyzyjnego śledzenia trajektorii zadanej koniecznym jest uwzględnienie dynamiki obiektu. W tym celu wprowadza się kontrolery kompensujące dynamikę robotów, szerzej omówione w pracach [23], [32], [74], [68].

Analizując literaturę pod kątem kontrolerów dynamicznych można zauważyć, że powszechnym rozwiązaniem są kontrolery, które generują sygnały sterujące w postaci momentu obrotowego (przedstawione w pracach [23], [50], [123]) lub kontrolery napięcia zasilania silników napędowych (przedstawione w pracach [77], [29], [28]). Strategia kontroli momentu uwzględnia momenty obrotowe silników jako wejścia sterujące, podczas gdy strategia kontroli napięcia koncentruje się na napięciach silnika jako wejściach sterujących.

Z kolei w pracy [68] pokazano, na przykładzie zadania śledzenia trajektorii, działanie adaptacyjnego kontrolera prędkości opartego na sztucznych sieciach neuronowych. W pierwszej kolejności sygnały sterujące generowane są przez kontroler prędkości, a następnie traktowane są jako sygnały wejściowe dla kontrolera adaptacyjnego. Adaptacja parametrów realizowana jest za pomocą sztucznych sieci neuronowych. Ostatecznie kontroler generuje sygnały sterujące w postaci momentów obrotowych dla silników napędzających koła.

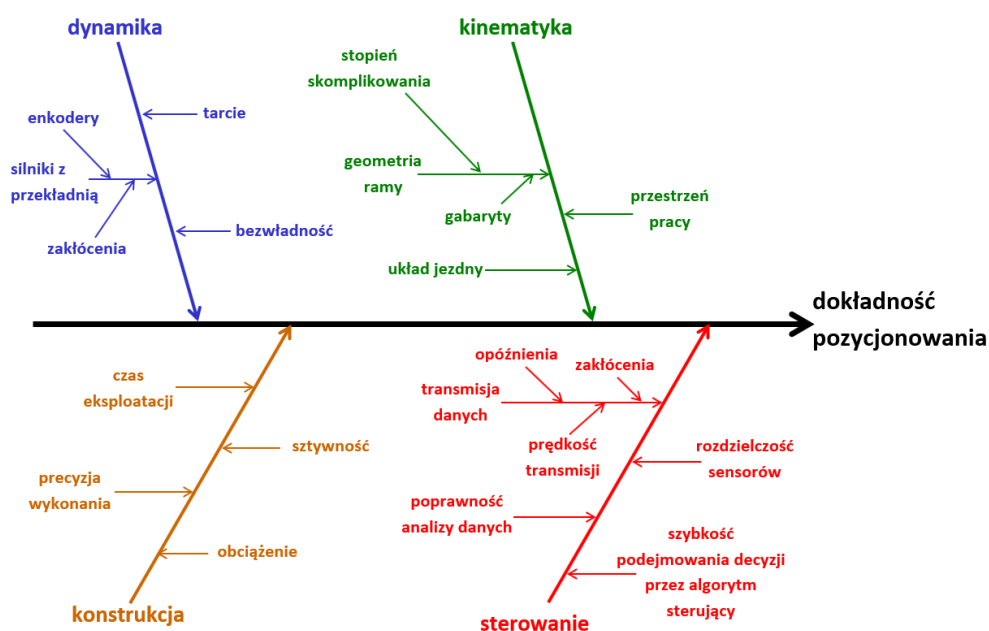
W pracy [123] przedstawiono metodę projektowania adaptacyjnego regulatora nadążnego dla dynamicznego modelu nieholonomicznego robota mobilnego o nieznanach parametrach metodą Adaptive Backstepping. Metoda Backstepping jest rekurencyjną procedurą projektowania kontrolera dla układu nieliniowego. Zgodnie z tą procedurą, w procesie projektowania złożony system można podzielić na n podsystemów (wychodząc od najprostszego, a w każdym kolejnym podsystemie uwzględniając stan poprzedniego), dla których łatwe będzie opracowanie kontrolera i określenie stabilności wykorzystując funkcję Lapunova. Postępując rekurencyjnie, czyli uwzględniając podczas tworzenia każdego nowego n - tego podsystemu stabilny stan podsystemu $n - 1$, zagwarantowana jest stabilność całego złożonego systemu. Metoda łączy wybór funkcji Lapunova (za pomocą, której określana jest stabilność systemu) z regulatorem ze sprzężeniem zwrotnym. Metoda projektowania Adaptive Backstepping jest zmodyfikowaną formą metody sterowania Backstepping, która wykorzystuje oszacowania dla nieznanach parametrów w systemach.

W pracy [77] opisano projekt algorytmu, który obejmuje dwie pętle sterowania, pętlę sterowania kinematycznego i pętlę sterowania dynamicznego. Pętla sterowania dynamicznego wykorzystuje strategię kontroli napięcia zasilania silników zamiast strategii kontroli momentu. Do systemu kontroli napięcia przekazywane są informacje zwrotne dotyczące orientacji robota, jego pozycji, prędkości liniowej i kątowej oraz prądu silników. W pracach [29] i [28] wyżej opisane podejście rozwinięto o estymację niepewności pomiarowych informacji przekazywanych w formie sprzężenia zwrotnego.

Dostępne na rynku laboratoryjne roboty mobilne zwykle jako sygnały sterujące przyjmują prędkość liniową i kątową. W związku z tym do sterowania takimi robotami zasadne jest zastosowanie kontrolerów generujących właśnie takie sygnały. Przykład kontrolera generującego sygnały sterujące w postaci prędkości liniowej i kątowej przedstawiono w pracach [24] i [76]. W tych pracach model dynamiczny robota mobilnego został zmodyfikowany w taki sposób, że liniowa i kątowa prędkość robota są traktowane jako sygnały sterujące. Następnie model dynamiczny został zlinearyzowany. Równanie regulatora dynamicznego wyprowadzono na podstawie zadania odwrotnego dynamiki. Otrzymany w ten sposób regulator dynamiczny poszerzono o prawo adaptacji do wyznaczenia nieznanach parametrów modelu dynamicznego robota.

1.3. Analiza błędów wpływających na dokładność pozycjonowania robotów w grupie rozproszonej

Wybierając robota do realizacji konkretnego zadania zazwyczaj kierujemy się jego dokładnością, funkcjonalnością oraz możliwościami jezdny. Parametry takie jak funkcjonalność oraz możliwości jezdne są cechami charakterystycznymi danego robota i nie ulegają zmianie w czasie eksploatacji. Biorąc pod uwagę stopień skomplikowania budowy robotów mobilnych, dokładność pozycjonowania jest parametrem, na który wpływa najwięcej czynników. Dokładność pozycjonowania nie jest stałą cechą, zależy od czynników związanych z charakterem pracy wykonywanej przez robota, długością i intensywnością eksploatacji, a także sposobem sterowania.



Rys. 1.11. Źródła pochodzenia błędów dokładności pozycjonowania robota mobilnego [27]

W literaturze spotyka się prace jak np. [121] i [27], których autorzy prowadzą badania dotyczące wpływu czynników zewnętrznych na dokładność pozycjonowania robotów. Jednym ze sposobów zobrazowania źródeł błędów wpływających na dokładność pozycjonowania robota jest opracowany w [27] diagram Ishikawy (Rys. 1.11), na którym uwzględniono kluczowe składowe mające wpływ na całkowitą dokładność pozycjonowania robota przemysłowego. Źródła błędów podzielono na cztery grupy: kinematyka, dynamika, sterowanie oraz konstrukcja. Dla każdej z grup wymieniono czynniki wpływające na ostateczny efekt, jakim jest dokładność pozycjonowania.

Rozważając wymienione na diagramie czynniki wpływające na dokładność pozycjonowania robota mobilnego, znaczenie części z nich można zredukować na etapie

projektowania robota i jego systemu. Są to przede wszystkim czynniki związane z dynamiką czy kinematyką, gdzie konstruktor w początkowym etapie projektowania może przeprowadzić symulacje i na ich podstawie zweryfikować poprawność doboru napędów czy gabarytów robota. Również w przypadku czynników związanych z konstrukcją ich znaczenie w dokładności pozycjonowania można zredukować bazując na doświadczeniu konstruktorskim, wiedzy inżynierskiej oraz dokładności podczas montażu. Dodatkowym aspektem pozwalającym na eliminację błędów, wynikających z długiej eksploatacji urządzeń, jest ciągle monitorowanie stanu maszyny i wprowadzanie nowych technologii diagnostyki usterek [124], [96], [90] oraz regularna kalibracja zastosowanych elementów pomiarowych [111], [20], [56].

Szczególną uwagę należy poświęcić grupie błędów związanych ze sterowaniem. Rozbudowana architektura systemu wymusza konieczność przetwarzania bardzo dużej ilości informacji z sensorów co wymaga niezawodnej pracy algorytmu sterowania. Dodatkowym i bardzo istotnym czynnikiem jest dynamicznie zmieniające się środowisko pracy. Faktem jest, że obecnie transmisja danych coraz częściej oparta jest o sieci bezprzewodowe, które pozwalają na pełną mobilność urządzeń. Ich systemy transmisji narażone są jednak na duże zakłócenia zewnętrzne takie jak np. szum elektromagnetyczny pochodzący od innych urządzeń elektronicznych, chwilowy lub całkowity zanik sygnału, niepożądane dane wysyłane do sieci przez uszkodzone systemy innych urządzeń, wadliwe układy zasilania urządzeń sieciowych.

Dryft czasu w systemach rozproszonych jest istotną kwestią, szczególnie w przypadku sieci mobilnych, gdyż może przerwać synchronizację między wewnętrznymi zegarami węzłów. W konsekwencji protokoły komunikacyjne w takich sieciach powinny uwzględniać ten problem w przypadku, gdy muszą korzystać z informacji o czasie. Pomimo zalet, systemy rozproszone stają przed ważnym wyzwaniem, jakim jest synchronizacja czasu. Rzeczywiście, nawet jeśli każda jednostka grupy ma swój własny zegar, aspekt rozproszony implikuje, że te zegary mogą dryfować, a tym samym nie mogą być zsynchronizowane [5]. W praktyce systemy wieloagentowe (grupy robotów) pracujące w dynamicznie zmiennych środowiskach napotykać nieoczekiwane zmiany w swoim otoczeniu, a ich skuteczne działanie zależy od ich szybkiej zbiorowej reakcji w dostosowywaniu się do zmieniających się okoliczności. Pomyślane zaprojektowanie tej klasy systemu wymaga zrównoważenia trzech czynników związanych z transmisją danych: wymaganego minimalnego przepływu danych, topologii sieci oraz stabilności lub możliwości stabilizacji działania zbiorowego. W pracy [81] przedstawiono przegląd licznych badań poświęconych skutkom ograniczonej prędkości transmisji danych, topologii sieci oraz opóźnień czasowych w systemach wieloagentowych. Jednym z przykładów

próby rozwiązania wyżej wymienionych problemów w systemach dynamicznych jest metoda sterowania grupą oparta na quasi-konsensusie, opisana w pracy [75]. Wykluczenie błędów powodowanych poprzez przedstawione czynniki jest nieodłącznym elementem budowy algorytmu sterowania robotem i ma istotny wpływ na ocenę jakości realizacji zadań przez robota.

1.4. Podsumowanie analizy literatury

Analizując liczną literaturę dotyczącą metod sterowania i wymiany informacji w grupie robotów można zauważyć, że obecnie najczęściej stosowaną jest struktura hierarchiczna systemu. Idea sterowania takim systemem polega na wyznaczeniu zadania i przydzieleniu do niego mastera - robota dowodzącego, który oblicza trajektorię i wybiera optymalny sposób jej realizacji. Pozycja mastera jest punktem odniesienia do obliczenia pozycji pozostałych robotów wchodzących w skład grupy. Współrzędne punktów ruchu robotów obliczane są poprzez dodanie przesunięcia w wybranym kierunku od pozycji odniesienia robota mastera. Algorytmy Śledzenia Lidera są łatwe w implementacji jednak nie zapewniają wysokiej dokładności ruchu.

Część zadań realizowanych przez grupy robotów wymaga jednak wysokiej precyzji i realizacji w określonym czasie. W tym przypadku wykorzystywane są algorytmy sterowania zdecentralizowanego, w którym roboty tworzą jedną sieć jednak w sposób niezależny od siebie interpretują przetwarzają sygnały sterujące i w odpowiedni sposób zmieniają swoje położenie lub działanie, czego efektem jest wspólna realizacja zadania [4], [25]. Ogromną zaletą systemów zdecentralizowanych w stosunku do systemów hierarchicznych jest ich niezawodność i skalowalność. Pomimo tych korzyści systemy rozproszone stoją przed ważnym wyzwaniem jakim jest synchronizacja czasu. Każdy robot w grupie ma swój własny zegar, a aspekt rozproszenia oznacza, że zegary te mogą dryfować wprowadzając błędy poruszania się robotów, a co za tym idzie błędy całej ich grupy [29], [30].

Niezależnie od przyjętej strategii sterowania grupą robotów, ostatecznie realizacja zadania przez grupę, sprowadza się do śledzenia trajektorii pojedynczego robota. Analizując literaturę pod kątem systemów sterowania można zauważyć, że większość proponowanych kontrolerów realizujących zadanie śledzenia trajektorii uwzględnia wyłącznie kinematykę robota. Jest to rozwiązanie dopuszczalne, jednak w przypadku zastosowania grupy robotów do zadań transportowych, zadań wymagającej dużej precyzji pozycjonowania oraz poruszania się z dużą prędkością, konieczne jest uwzględnienie w sterowaniu dynamiki robota.

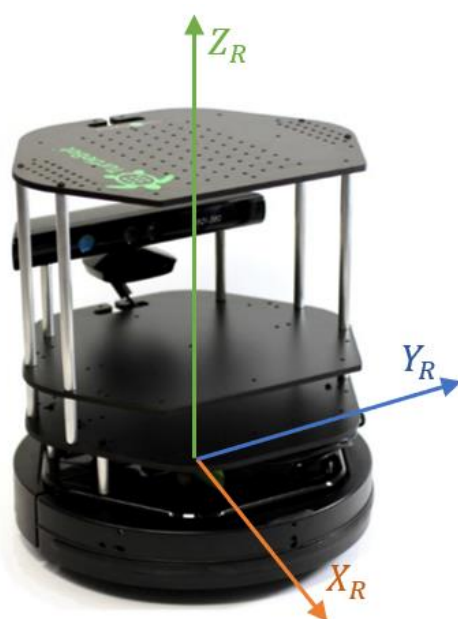
W literaturze popularne są modele dynamiczne robotów uwzględniające jako sygnały sterujące momenty obrotowe lub napięcia silników. Zdecydowanie łatwiejszym do implementacji rozwiązaniem jest zastosowanie modelu dynamiki, który jako sygnały sterujące przyjmuje prędkość liniową i kątową robota. W taki właśnie sposób sterowane są zwykle dostępne na rynku roboty mobilne.

Na podstawie analizy metod sterowania zauważono możliwość poprawy parametrów ruchu grupy robotów, wprowadzając w systemie sterowania synchronizację czasową jednostek. W systemie takim, w taktowaniu sygnałów sterujących jednostkami, należy uwzględnić dryft czasu, czyli różnicę w taktowaniu zegarów poszczególnych jednostek. Innymi słowy poprzez komunikację pomiędzy robotami należy badać wartość odchylenia czasu poszczególnych jednostek i uwzględniać ją w algorytmie sterowania grupą robotów.

Rozdział 2

OBIEKT BADAŃ

Prezentowane w niniejszej pracy rozważania obejmują grupę robotów TURTLEBOT 2 (Rys. 2.1) wyposażonych w system operacyjny ROS. Roboty te są na wyposażeniu Laboratorium Zakładu Mechatroniki Instytutu Techniki Rakietowej i Mechatroniki. Robot TURTLEBOT 2 jest dwukołowym robotem mobilnym, napędzanym w układzie różnicowym, z dwoma kołami podporowymi. Dodatkową zaletą przemawiającą za zastosowaniem tego robota jest jego układ jezdny stanowiący bazę do rozbudowy do układów jezdnych przedstawionych na Rys. 1.1. W związku z powyższym opracowanie algorytmu sterowania i uruchomienie na wspomnianym robocie umożliwia prostą implementację na innych stosowanych w przemyśle robotach wymienionych na Rys. 1.1.



Rys. 2.1. Laboratoryjny robot TURTLEBOT 2 z bazowym układem współrzędnych [70]

2.1. Laboratoryjny robot TURTLEBOT 2

Robot TURTLEBOT 2 jest niskobudżetowym zestawem dydaktycznym, bo budowy i testowania systemów sterowania kołowych robotów mobilnych. Platforma wyposażona jest w dwukołowy napęd różnicowy oraz podstawowe czujniki takie jak: skaner laserowy, kamera wizyjna RGB o rozdzielczości 640x480 (wykorzystywana do przetwarzania obrazu,

rozpoznawania kolorów i tekstur), kamera IR (przetwarzanie informacji o głębokości, pomiar odległości w zakresie 0.4 - 6.5 m), cztery mikrofony kierunkowe oraz akcelerometr. Zaletą przemawiającą za zastosowaniem omawianego robota jest jego otwarty system sterowania ROS, który umożliwia szybkie uruchomienie platformy i integrację z czujnikami [69].

Dane techniczne dotyczące robota TURTLEBOT 2 zestawiono w tabelach poniżej:

Tabela 1. Gabaryty i masa robota TURTLEBOT 2 [69]

Wymiary (długość × szerokość × wysokość)	354 × 354 × 420 mm
Masa własna	6,3 kg
Średnica kół (2r)	76 mm
Rozstaw kół (2d)	230 mm

Tabela 2. Parametry funkcjonalne i jezdne robota TURTLEBOT 2 [69]

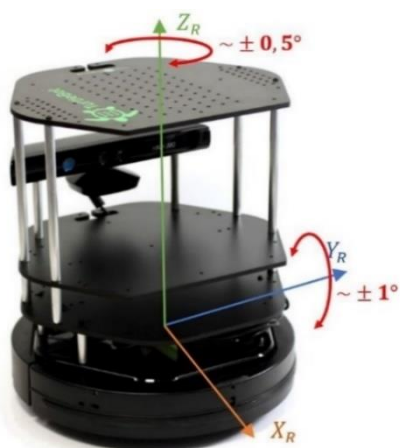
Max. prędkość liniowa	$0,7 \frac{\text{m}}{\text{s}}$
Max. prędkość kątowna	$180 \frac{\text{deg}}{\text{s}}$
Ładowność	5 kg
Max. wysokość pionowej przeszkody	12 mm
Rozdzielczość odometrii	$52 \frac{\text{impuls}}{\text{obrót osi enkodera}}, 2578,33 \frac{\text{impuls}}{\text{obrót koła}}, 11,7 \frac{\text{impuls}}{\text{mm}}$
Giroskop	1 oś, $110 \frac{\text{deg}}{\text{s}}$

Dokumentacja techniczna nie zawiera jednak informacji dotyczących momentów bezwładności robota. Do pozyskania tych danych wykorzystano model CAD robota TURTLEBOT 2 opracowany w środowisku SOLIDWORKS, który z dużą dokładnością odwzorowuje jego geometrię. Po weryfikacji gęstości każdego z elementów modelu (zadeklarowania materiału, z jakiego wykonany jest dany element) przeprowadzono analizę pozwalającą na wyznaczenie momentów bezwładności odniesionych do zadeklarowanego układu współrzędnych. Wymagane momenty bezwładności wyznaczono dla układu współrzędnych umieszczonego w środku geometrycznym podstawy robota, zorientowanym jak na Rys. 2.1. Wyznaczone momenty bezwładności robota TURTLEBOT 2 przedstawiono w tabeli poniżej:

Tabela 3. Momenty bezwładności robota TURTLEBOT 2

Moment robota względem osi X_R	$0,1593 \text{ kg} \cdot \text{m}^2$
Moment robota względem osi Y_R	$0,1574 \text{ kg} \cdot \text{m}^2$
Moment robota względem osi Z_R	$0,0896 \text{ kg} \cdot \text{m}^2$

Podczas badań laboratoryjnych robota TURTLEBOT2 stwierdzono wady konstrukcyjne, które wprowadzały błędy odczytów prędkości kątowej przez żyroskop, co przekładało się na błędy położenia kąowego w dwóch osiach jak wskazano na Rys. 2.2. Ze względu na obniżony montaż kół napędowych względem kół podporowych występuje możliwość kołysania się robota podczas jazdy – robot może być odchylony do przodu lub tyłu. Wynikało to z faktu, że z podłożem w jednej chwili miało kontakt tylko jedno koło podporowe. Wskutek powyższego, w trakcie badań doświadczalnych zidentyfikowano błąd odczytu położenia kąowego względem osi Y_R o wartości około $\pm 1^\circ$. Ze względu na występujące w układzie jezdny luzy na kołach zidentyfikowano błąd odczytu położenia kąowego względem osi Z_R o wartości około $\pm 0.5^\circ$. W celu ograniczenia wpływu błędu położenia kąowego względem osi Y_R , robota pozbawiono górnej platformy montażowej oraz sensora Kinect, a komputer pokładowy robota zamontowano na dolnej platformie montażowej. Dodatkowo robota obciążono ciężarkiem o masie 2.5 kg umieszczonym w przedniej części i minimalnie wysuniętym poza obrys robota, a przyspieszenie liniowe robota ograniczono programowo do wartości $0.4 \frac{m}{s^2}$. Wprowadzone modyfikacje (Rys. 2.3) wyeliminowały efekt kołysania się robota podczas ruchu, a co za tym idzie błąd położenia kąowego w osi Y_R . W celu ograniczenia wpływu błędu położenia kąowego względem osi Z_R zastosowano filtrację programową za pomocą filtra uśredniającego i ograniczenie przyspieszenia kątowego do wartości $0.8 \frac{rad}{s^2}$.



Rys. 2.2. Błędy położenia kąowego wynikające z wad konstrukcyjnych robota TURTLEBOT 2



Rys. 2.3. Zmodyfikowana konstrukcja robota TURTLEBOT 2

Wprowadzone modyfikacje mechaniczne spowodowały zmianę momentów bezwładności robota względem wskazanego na Rys. 2.1 układu współrzędnych. Po wprowadzeniu opisanych modyfikacji do modelu CAD robota wyznaczono nowe momenty bezwładności, które umieszczono w tabeli poniżej:

Tabela 4. Momenty bezwładności zmodyfikowanego robota TURTLEBOT 2

Moment robota względem osi X_R	0,1447 kg * m ²
Moment robota względem osi Y_R	0,1275 kg * m ²
Moment robota względem osi Z_R	0,1277 kg * m ²

Do budowy grupy o strukturze rozproszonej zastosowano trzy roboty TURTLEBOT 2 o zmodyfikowanej konstrukcji jak przedstawiono na Rys. 2.4 (umieszczone na robotach laptopy są otwarte jedynie w celu prezentacji).



Rys. 2.4. Grupa robotów TURTLEBOT 2

2.2. Robot Operating System

Ze względu na wybrany przedmiot badań, opisany w rozdziale 2, konieczne było opracowanie systemu sterowania w ROS (Robot Operating System), w który to system wyposażone są roboty TURTLEBOT 2. ROS to platforma programistyczna do tworzenia oprogramowania dla robotów, którą można traktować jako system o strukturze klient/serwer. Platforma wyposażona jest we własny mechanizm budowania kodu do postaci programów wykonywalnych. Dzięki temu prawidłowa kolejność dołączania bibliotek, zależność z innymi modułami i uruchamiania narzędzi jest w każdym module jednakowa. Pozwala to zaoszczędzić czas podczas szukania błędów zarówno w czasie kompilacji jak i wykonywania programu. Opisany charakter platformy ROS daje możliwość poprawnego działania programu niezależnie od poziomu skomplikowania. Ze względu na rozproszony charakter platformy nie jest istotne czy poszczególne moduły znajdują się na jednym komputerze, pracują w sieci lokalnej, czy poprzez łącze internetowe na wielu komputerach. Własność ta pozwala na przeniesienie ograniczeń wydajności i poziom skomplikowania systemu, głównie na przepustowość połączeń sieciowych.

Twórcy platformy ROS zaproponowali strukturę projektu, która wprowadza porządek pomagający w jego budowie i konfiguracji. Struktura projektu ROS składa się z pakietów (ang. Package), które obejmują węzły (ang. Nodes), tematy (ang. Topics) i wiadomości (ang. Messages), usługi i parametry.

Pakiety są podstawowymi jednostkami organizacyjnymi projektowanego systemu w ROS. Pakietem nazywany jest każdy moduł programowy realizujący składową funkcjonalność całego systemu. Nazwa pakietu jest jednocześnie nazwą katalogu, w którym znajdują się wszystkie składowe danego modułu programowego. Dodatkowo kompilator systemu w ROS podczas budowy projektu tworzy plik XML, zawierający specyfikację konkretnego modułu programowego zrozumiałą dla platformy ROS. Plik XML składa się z dwóch części: pierwsza część zawiera dane takie jak nazwę i opis pakietu, autora, licencję użytkownika pakietu, druga część określa zależność danego pakietu od innych oraz dodatkowe informacje takie jak język programowania czy wersję ROS. Najważniejszym elementem pliku XML jest informacja o zależności z innymi pakietami wymagana dla wewnętrznego mechanizmu budowania projektu, dzięki której kompilator wie, które pakiety i w jakiej kolejności należy zbudować przed kompilacją opisywanego pakietu. Podczas kompilacji pakietu automatycznie tworzony jest także plik konfiguracyjny CMakeList.txt, który programista musi wypełnić wybierając potrzebne dla danego projektu makra i flagi.

Węzły są podstawowymi elementami wykonywalnymi w pakietach systemu ROS. Realizują komunikację ze sprzętem zewnętrznym, użytkownikiem jak i komunikację wewnątrzsistemową. Węzły tworzą sieć *peer-to-peer* (wszystkie urządzenia są równe w hierarchii) wymieniając między sobą informacje w postaci żądań i wiadomości. Węzły w systemie ROS funkcjonują jak ścieżki dostępu (np. do publikowanych wiadomości).

Wiadomości są ramkami danych, przesyłanymi między węzłami przez wirtualne połączenie komunikacyjne nazywane tematem. Jedne węzły mogą tworzyć, udostępniać i publikować wiadomości do tematów, inne natomiast nasłuchują tematy i pobierają pojawiające się wiadomości. Wiadomości mogą mieć strukturę zdefiniowaną w standardowym pakiecie platformy ROS (np. prędkość, położenie, godzina, poziom baterii) lub mogą być zdefiniowane przez programistę wewnątrz poszczególnych pakietów.

Ze względu na sieciowy i standaryzowany charakter platformy ROS, możliwe jest stworzenie modułów programowych współpracujących z systemem w dowolnym języku programowania. Cecha ta stanowi niezależność od języka programowania i pozwala na współpracę w obrębie jednej aplikacji elementów napisanych w różnych językach programowania. Przedmiotowy system sterowania opracowano przy użyciu języka C++.

2.3. Opracowanie modelu matematycznego robota TURTLEBOT 2

Model matematyczny robota TURTLEBOT 2 podzielono na dwie części: model kinematyczny i model dynamiczny. Wyprowadzenie poszczególnych modeli przedstawiono w rozdziałach poniżej. Model dynamiczny robota zawiera nieznane parametry, które należało zidentyfikować. W końcowym kroku przeprowadzono walidację modelu teoretycznego.

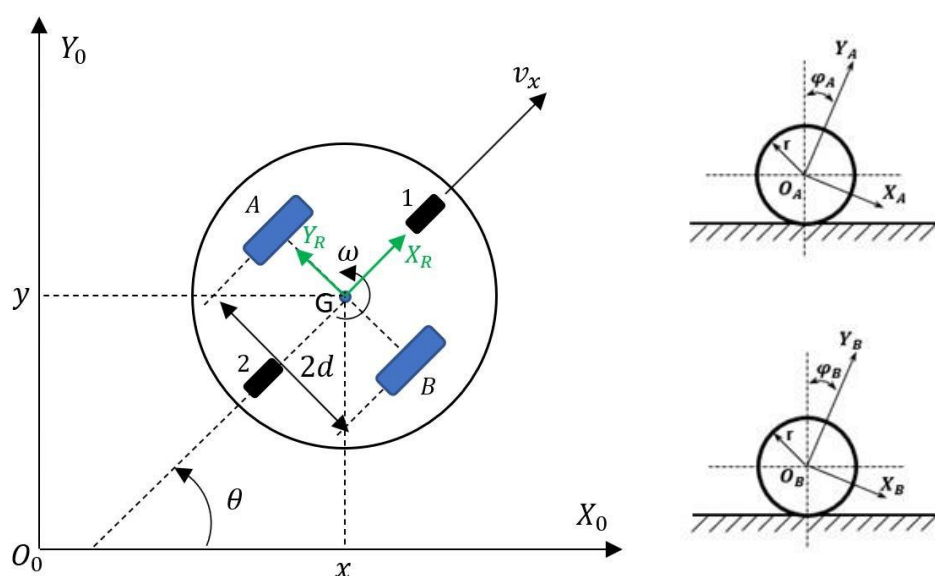
2.3.1. Model kinematyczny

Do wyznaczenia modelu kinematycznego robota TURTLEBOT 2 opracowano jego strukturę kinematyczną, na której uwzględniono podstawowe cechy robota. Schemat przedstawiono na Rys. 2.5.

Modelując kinematykę robota mobilnego należy rozważyć każdą ruchomą część wchodzącą w jego budowę oraz przypisany do niej prawoskrętny układ współrzędnych.

Do opisu rozważanego zagadnienia przyjęto następujące założenia upraszczające:

- model robota zastąpiono modelem z jednym kołem zastępczym umieszczonym w środku geometrycznym robota i promieniu zgodnym z promieniem kół napędowych;
- ruch odbywa się na płaszczyźnie X_0Y_0 , nie będą rozpatrywane zmiany wysokości robota;
- ruch odbywa się bez poślizgu kół;
- rama robota oraz zastępcze koło napędowe są bryłami sztywnymi o znanych bezwładnościach;
- tarcie występujące w łożysku koła zastępczego zostanie pominięte;
- prędkości kątowe kół są znane.



Rys. 2.5. Struktura kinematyczna robota TURTLEBOT 2 do opracowania modelu kinematycznego

W analizowanym przypadku rozpatrywane były wyłącznie prędkościowe ograniczenia ruchu wprowadzane przez koła napędowe A i B. Pozostałe koła podporowe wyłącznie zwiększają stabilność i nośność konstrukcji. Przyjęto, że robot zbudowany jest ze sztywnej ramy nośnej o kształcie koła oraz zespołu napędowego. Zespół napędowy stanowią dwa koła A i B napędzane niezależnie dwoma silnikami prądu stałego. Elementem sprzęgającym koła z silnikami są przekładnie. Sprzężenie zwrotne, dające informacje o położeniu kątowym poszczególnych kół napędowych, stanowią enkodery. Koła napędowe obracają się wokół własnych osi i nie zmieniają położenia względem ramy. Sterowanie kierunkiem ruchu realizowane jest poprzez sterowanie prędkością poszczególnych kół napędowych. Jeżeli koła będą obracały się w tym samym kierunku, lecz z różną prędkością robot będzie poruszał się po łuku. Jeżeli koła będą obracały się w tym samym kierunku i z taką samą prędkością robot będzie przemieszczał się ruchem prostoliniowym. Obrót kół z tą samą prędkością, lecz w przeciwnym kierunku spowoduje obrót robota wokół punktu G - środka geometrycznego robota, który jest jednocześnie punktem toru ruchu robota.

Orientację robota w przestrzeni opisano wprowadzając układy współrzędnych dla punktów charakterystycznych robota. Przedstawiony na Rys. 2.5 układ współrzędnych $O_0X_0Y_0$ stanowi globalny, nieruchomy układ odniesienia. Układ współrzędnych $O_RX_RY_R$ jest układem o początku w środku geometrycznym robota i jest związany z ramą robota. Układy współrzędnych $O_AX_AY_A$ oraz $O_BX_BY_B$ związane są z kołami napędowymi, kolejno kołem A i kołem B, o początku w osiach kół. Kąty φ_A oraz φ_B są kątami obrotu własnego kół A i B.

Korzystając ze współrzędnych jednorodnych na podstawie prac [97], [33], [34], [72] określono jednorodne macierze transformacji wybranych układów współrzędnych do nieruchomego układu odniesienia $O_0X_0Y_0$. Macierz T_{RO} w postaci:

$$T_{RO}(t) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) & 0 & x(t) \\ \sin\theta(t) & \cos\theta(t) & 0 & y(t) \\ 0 & 0 & 1 & z(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

jest macierzą transformacji układu $O_RX_RY_R$ związanego ze środkiem geometrycznym robota do układu odniesienia $O_0X_0Y_0$. Transformację układów współrzędnych $O_AX_AY_A$ i $O_BX_BY_B$ związanych z kołami A i B do układu odniesienia $O_0X_0Y_0$ opisują macierze T_{AO} i T_{BO} :

$$\begin{aligned}
T_{Ao}(t) &= \begin{bmatrix} \cos\theta(t)\sin\varphi_A(t) & \cos\theta(t)\cos\varphi_A(t) & -\sin\theta(t) & x(t) + d\sin\theta(t) \\ \sin\theta(t)\sin\varphi_A(t) & \sin\theta(t)\cos\varphi_A(t) & \cos\theta(t) & y(t) - d\cos\theta(t) \\ \cos\varphi_A(t) & -\sin\varphi_A(t) & 0 & z(t)^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
T_{Bo}(t) &= \begin{bmatrix} \cos\theta(t)\sin\varphi_B(t) & \cos\theta(t)\cos\varphi_B(t) & -\sin\theta(t) & x(t) + d\sin\theta(t) \\ \sin\theta(t)\sin\varphi_B(t) & \sin\theta(t)\cos\varphi_B(t) & \cos\theta(t) & y(t) - d\cos\theta(t) \\ \cos\varphi_B(t) & -\sin\varphi_B(t) & 0 & z^2(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2.2}$$

Dla rozważanego robota laboratoryjnego określono wektor współrzędnych konfiguracyjnych, w następującej postaci:

$$q(t) = [x(t), y(t), \theta(t), \varphi_A(t), \varphi_B(t)] \tag{2.3}$$

Dokonując definicji ograniczeń nieholonomicznych, nałożonych na prędkości kół w kierunku wzdłużnym i poprzecznym jako:

$$\dot{x}(t)\sin\theta(t) - \dot{y}(t)\cos\theta(t) = 0 \tag{2.4}$$

sformułowano równania opisujące kinematykę rozważanego robota w postaci:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{\varphi}_A(t) \\ \dot{\varphi}_B(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) \\ \sin\theta(t) \\ 0 \\ -\frac{d}{r} \\ \frac{d}{r} \end{bmatrix} u_{kv}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ \frac{1}{r} \\ \frac{1}{r} \end{bmatrix} u_{k\omega}(t) \tag{2.5}$$

gdzie:

$u_{kv}(t)$ – prędkość liniowa robota, a $u_{k\omega}(t)$ – prędkość kątowa robota.

Przekształcając równanie (2.5) do postaci, w której sygnałami wejściowymi są prędkości kątowe kół napędowych A i B otrzymano następujący model kinematyczny robota:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \frac{r}{2d}\cos\theta(t) & -\frac{r}{2}\cos\theta(t) \\ \frac{r}{2d}\sin\theta(t) & -\frac{r}{2}\sin\theta(t) \\ \frac{r}{2d} & -\frac{r}{2} \end{bmatrix} \begin{bmatrix} u'_A(t) \\ u'_B(t) \end{bmatrix} \tag{2.6}$$

$$\begin{bmatrix} \dot{\varphi}_A(t) \\ \dot{\varphi}_B(t) \end{bmatrix} = \begin{bmatrix} u'_A(t) \\ u'_B(t) \end{bmatrix}$$

Łącząc powyższe równania w jeden układ równań otrzymano model kinematyczny robota w postaci:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) & 0 \\ \sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} r \begin{bmatrix} \dot{\phi}_A(t) + \dot{\phi}_B(t) \\ 0 \\ \dot{\phi}_A(t) - \dot{\phi}_B(t) \end{bmatrix} \frac{1}{2d} \quad (2.7)$$

Przekształcając powyższe równanie do postaci, gdzie sygnałami sterującymi są prędkość liniowa i kątowa robota, opisane poniższymi zależnościami:

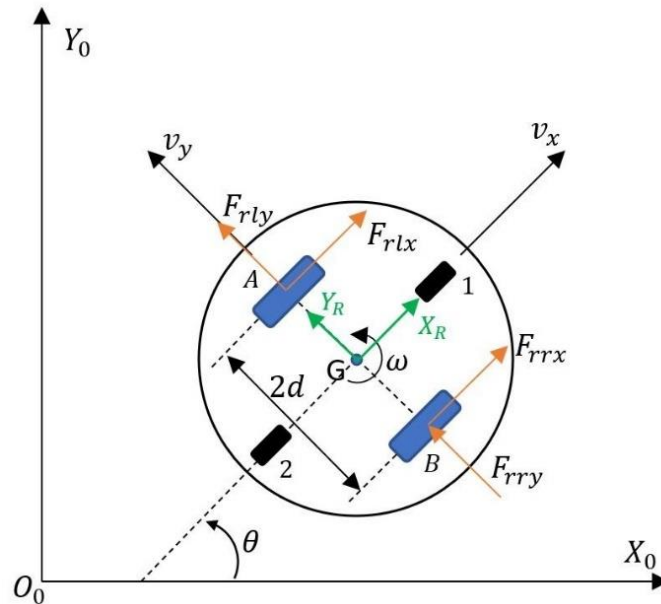
$$u_{kv}(t) = v(t) = \frac{r(\dot{\phi}_A(t) + \dot{\phi}_B(t))}{2}, \quad u_{k\omega}(t) = \omega = \frac{r(\dot{\phi}_A(t) - \dot{\phi}_B(t))}{2d} \quad (2.8)$$

otrzymano końcową postać modelu kinematycznego robota:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (2.9)$$

2.3.2. Model dynamiczny

Do opracowania modelu dynamicznego robota posłużono się schematem przedstawionym na Rys. 2.6 oraz pracami [24] i [76].



Rys. 2.6. Schemat robota TURTLEBOT 2 do opracowania modelu dynamicznego

Punkt G przedstawiony na schemacie robota TURTLEBOT 2 (Rys. 2.6) jest środkiem masy i środkiem obrotu robota oraz jednocześnie punktem realizującym (kreślącym) trajektorię zadaną.

Pozycja i orientacja robota są opisane następującymi równaniami:

$$\begin{aligned}\dot{x}(t) &= v_x(t)\cos\theta(t) \\ \dot{y}(t) &= v_y(t)\sin\theta(t) \\ \dot{\theta}(t) &= \omega(t)\end{aligned}\tag{2.10}$$

Prędkości poruszanie się robota oznaczono jako:

$v_x(t), v_y(t)$ – prędkość wzdłużna i boczna,

$\omega(t)$ – prędkość kątowna.

Na Rys. 2.6 oznaczono siły napędowe robota jako F_{rrx}, F_{rry} - siła wzdłużna i poprzeczna koła prawego oraz F_{rlx}, F_{rly} - siła wzdłużna i poprzeczna koła lewego. W celu opracowania modelu dynamicznego omawianego robota, na podstawie twierdzenia o zmianie pędu i momentu pędu sformułowano wyjściowy układ równań:

$$\begin{aligned}\sum F_x(t) &= F_{rlx}(t) + F_{rrx}(t) = m(\dot{v}_x(t) - v_y(t)\omega(t)) \\ \sum F_y(t) &= F_{rly}(t) + F_{rry}(t) = m(\dot{v}_y(t) - v_x(t)\omega(t)) \\ \sum M_z(t) &= I_z\omega(t) = \frac{d}{2}(F_{rrx}(t) - F_{rlx}(t))\end{aligned}\tag{2.11}$$

gdzie:

m – masa robota,

I_z – moment bezwładności względem osi przechodzącej przez punkt G.

Bazując na [129] i [75] dla prędkości liniowej v_x (zgodnej z kierunkiem osi X_R układu współrzędnych robota) oraz prędkości kątowej ω , sformułowano następujące zależności:

$$\begin{aligned}v_x(t) &= \frac{1}{2}\left(r(\omega_p(t) + \omega_l(t)) + (v_p^s(t) + v_l^s(t))\right) \\ \omega(t) &= \frac{1}{2}\left(r(\omega_p(t) - \omega_l(t)) + (v_p^s(t) - v_l^s(t))\right)\end{aligned}\tag{2.12}$$

gdzie:

r – promień kół napędowych,

$\omega_p(t), \omega_l(t)$ – prędkości kątowe odpowiednio koła prawego i lewego,

$v_p^s(t), v_l^s(t)$ – prędkości poślizgu wzdłużnego odpowiednio koła prawego i lewego.

Elementami napędowymi robota są dwa silniki prądu stałego opisane modelem [24]:

$$\begin{aligned}\tau_r(t) &= \frac{k_a(V_r(t) - k_b\omega_p(t))}{R_a} \\ \tau_l(t) &= \frac{k_a(V_l(t) - k_b\omega_l(t))}{R_a}\end{aligned}\tag{2.13}$$

gdzie:

$V_p(t)$, $V_l(t)$ – napięcia zasilania odpowiednio silnika prawego i lewego,

k_a – stała elektryczna silnika pomnożona przez przełożenie przekładni,

k_b – stała mechaniczna pomnożona przez przełożenie przekładni,

R_a – rezystancja elektryczna uzwojeń silników.

Dostępne na rynku roboty laboratoryjne zwykle mają wbudowane, w ich wewnętrzny układ sterowania regulatory prędkości, a użytkownik nie ma bezpośredniego dostępu do modyfikacji tego układu sterowania. W związku z tym, w modelu dynamicznym robota należy uwzględnić regulator prędkości, który jako sygnały sterujące traktuje prędkość liniową i kątową. W formułowanym modelu przyjęto regulator PD opisany równaniami [24]:

$$\begin{bmatrix} u_v(t) \\ u_w(t) \end{bmatrix} = \begin{bmatrix} k_{PT}(u_v(t) - u_{me}(t)) - k_{DT}\dot{u}_{me}(t) \\ k_{PR}(u_w(t) - \omega_{me}(t)) - k_{DR}\dot{\omega}_{me}(t) \end{bmatrix} \quad (2.14)$$

gdzie:

$$\begin{aligned} u_v(t) &= \frac{V_l(t) + V_p(t)}{2}, \quad u_w(t) = \frac{V_p(t) - V_l(t)}{2} \\ u_{me}(t) &= \frac{1}{2} \left[r(\omega_p(t) + \omega_l(t)) \right], \quad \omega_{me}(t) = \frac{1}{d} \left[r(\omega_p(t) - \omega_l(t)) \right] \end{aligned} \quad (2.15)$$

którego wzmocnienia proporcjonalne przyjmują wartości dodatnie, a wzmocnienia różniczkowe przyjmują wartości nieujemne:

$$k_{PT} > 0, \quad k_{PR} > 0, \quad k_{DT} \geq 0, \quad k_{DR} \geq 0 \quad (2.16)$$

Łącząc równania od (2.10) do (2.15) oraz dokonując przekształceń analitycznych (Załącznik A) otrzymano model dynamiczny robota TURTLEBOT 2 w następującej postaci:

$$\begin{bmatrix} \dot{v}_x(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} -\frac{\sigma_3}{\sigma_1} v_x(t) \\ -\frac{\sigma_4}{\sigma_2} \omega(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma_1} & 0 \\ 0 & \frac{1}{\sigma_2} \end{bmatrix} \begin{bmatrix} v_{xd}(t) \\ \omega_d(t) \end{bmatrix} \quad (2.17)$$

Parametry $\sigma_1 \dots \sigma_4$ modelu (2.17) są funkcjami parametrów fizycznych robota takich jak: masa m , moment bezwładności robota I_z , rezystancja elektryczna uzwojeń silników R_a , stałe silnika k_a i k_b , współczynnik tarcia B_e , moment bezwładności przekładni I_e , promień kół r , rozstaw kół d . Można je opisać następującymi zależnościami:

$$\begin{aligned}
\sigma_1 &= -\frac{R_a R_t}{k_{PT} k_a} \left(\frac{m}{2} + \frac{I_e}{d} \right) - \frac{k_{DT}}{k_{PT}} [s] \\
\sigma_2 &= \frac{R_a R_t}{d k_{PR} k_a} (I_z - I_e) - \frac{k_{DR}}{k_{PR}} [s] \\
\sigma_3 &= \frac{k_b}{k_{PT}} - \frac{B_e R_a R_t}{r k_{PT} k_a} \\
\sigma_4 &= 1 - \frac{d k_{DR} k_b}{k_{PR}} - \frac{B_e R_a R_t}{d k_{PR} k_a}
\end{aligned} \tag{2.18}$$

Parametry modelu $\sigma_1 \dots \sigma_4$ zawierają parametry fizyczne robota bardzo trudne do uzyskania przez użytkownika robotów laboratoryjnych. Zwykle nie są one zawarte nawet w instrukcjach obsługi czy kartach technicznych urządzeń, dlatego konieczne było przeprowadzenie identyfikacji parametrów $\sigma_1 \dots \sigma_4$, której proces opisano w rozdziale 2.3.3.

2.3.3. Identyfikacja parametrów modelu dynamicznego

Przed przystąpieniem do identyfikacji parametrów modelu należało sprawdzić czy któryś z parametrów modelu może być zapisany jako liniowa kombinacja dwóch innych. W takim przypadku możliwy byłby zapis modelu dynamicznego robota z mniejszą liczbą parametrów. Analizując opracowany model matematyczny robota (2.17) i jego parametry (2.18) liniowa niezależność między parametrami $\sigma_1 \dots \sigma_4$ nie jest wyraźnie widoczna, ponieważ niektóre parametry fizyczne robota mają wpływ na więcej niż jeden parametr σ_n , dlatego konieczna jest szczegółowa analiza, którą przeprowadzono w pracy [76] wykazując, że parametry modelu od σ_1 do σ_4 są niezależne, czyli nie mogą być zapisane jako ich kombinacja liniowa. Ostatecznie konieczne jest przeprowadzenie identyfikacji każdego z parametrów $\sigma_1 \dots \sigma_4$.

Identyfikację offline parametrów modelu dynamicznego robota przeprowadzono wykorzystując metody offline i pomiary rzeczywistej trajektorii testowej zrealizowanej przez robota przy zadanych znanych, stałych wartościach prędkości liniowej i kątowej [105], [102]. Identyfikacja jest procesem iteracyjnym, w związku z tym w celu wykonania pierwszej iteracji na wstępie musiały zostać wprowadzone wartości startowe identyfikowanych parametrów. Aby ocenić wpływ początkowego błędu parametrów modelu na czas i dokładność identyfikacji, proces przeprowadzono dla losowych wartości startowych identyfikowanego parametru opisanych rozkładem normalnym o wartości oczekiwanej $\mu = 1$ wariancji $\sigma^2 = 1$ w przedziale $\langle 0, 1 \rangle$.

Identyfikację parametrów modelu robota przeprowadzono wykorzystując i porównując dwie metody identyfikacji parametrów funkcji: metodę Levenberga-Marguardta (w Matlabie funkcja lsqnonline) dla kryterium minimalizacji w postaci [35], [81]:

$$\min_d \|J(x)d + F(x)\|_2^2 \quad (2.19)$$

gdzie $J(x)$ jest jacobianem wektora $F(x)$, a d jest kierunkiem poszukiwania minimum funkcji wyznaczany zgodnie z regułą:

$$[J(x)^T J(x) + \alpha I]d = -J(x)^T F(x), \quad \text{dla } \alpha \geq 0 \quad (2.20)$$

oraz metodę quasi-Newtona (w Matlabie funkcja `fminunc`), dla której kryterium minimalizacji przyjmuje postać [120], [38], [37]:

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} x^T H x \right\} \quad (2.21)$$

gdzie H jest symetryczną i dodatnio określoną macierzą hesjanu aproksymowanej funkcji.

W procesie identyfikacji przeprowadzono 50 prób. Następnie wyznaczono wartość średnią ze wszystkich prób identyfikowanych parametrów oraz wariancję i odchylenie standardowe. Przykładowe wartości parametrów modelu dynamicznego robota zidentyfikowanych metodami quasi-Newtona i Levenberga-Marguardta przedstawiono w Tabeli 5. Na podstawie odchylenia standardowego stwierdzono poprawność procesu identyfikacji dla metody Levenberga-Marguardta oraz błędną identyfikację metodą quasi-Newtona.

Tabela 5. Przykładowe wartości parametrów modelu zidentyfikowane metodami quasi-Newtona i Levenberga-Marguardta

Nr próby		σ_1	σ_2	σ_3	σ_4	Czas obliczeń [s]
1	Wartości startowe	0,16	0,52	0,96	0,37	
	lsqnonline	4,09	5,46	0,97	0,29	6,65
	fminunc	253,07	186,13	-4,54	-4,47	52,34
2	Wartości startowe	0,61	0,26	0,76	0,29	
	lsqnonline	3,14	4,06	0,81	0,24	6,01
	fminunc	0,59	0,85	0,39	0,15	45,96
3	Wartości startowe	0,59	0,55	0,92	0,29	
	lsqnonline	3,14	4,17	0,83	0,24	6,40
	fminunc	107,60	145,67	0,48	-2,84	64,27
4	Wartości startowe	0,65	0,45	0,55	0,30	
	lsqnonline	3,49	4,63	0,85	0,24	5,92
	fminunc	0,55	0,63	0,68	0,30	38,10
5	Wartości startowe	0,38	0,56	0,08	0,05	
	lsqnonline	3,37	4,56	0,84	0,26	5,65
	fminunc	52,34	16,42	0,49	7,64	47,98

lsqnonline					
Wartość średnia	3,45	4,58	0,86	0,26	6,13
Wariancja	0,12	0,24	0,00	0,00	
Odchylenie standardowe	0,35	0,49	0,06	0,02	
fminunc					
Wartość średnia	82,83	69,94	-0,50	0,16	52,64
Wariancja	8811,65	6335,37	4,09	17,28	
Odchylenie standardowe	93,87	79,60	2,02	4,16	

Błędny proces identyfikacji metodą quasi-Newtona związany jest z tym, że metoda wykazuje trudności z identyfikacją, jeżeli punkt początkowy położony jest daleko od minimum lokalnego. W związku z tym postanowiono dla metody quasi-Newtona zawęzić przedziały losowych wartości startowych parametrów, opierając się wynikami otrzymanymi metodą Levenberga-Marguardta: $\sigma_{1_0} \in \langle 3, 4 \rangle$, $\sigma_{2_0} \in \langle 4, 5 \rangle$, $\sigma_{3_0} \in \langle 0.8, 0.9 \rangle$, $\sigma_{4_0} \in \langle 0.2, 0.3 \rangle$. Wyniki identyfikacji parametrów dla zawężonych przedziałów wartości startowych przedstawiono w Tabeli 6. Na podstawie obliczonych ich odchyłeń standardowych można stwierdzić, że dla zawężonych przedziałów parametrów startowych, identyfikacja metodą quasi-Newtona przebiegła poprawnie. Zidentyfikowane wartości parametrów są zbieżne z metodą Levenberga-Marguardta. Małe wartości odchylenia standardowego zidentyfikowanych parametrów sugerują poprawność procesu identyfikacji dla wybranych metod, co zostało zweryfikowane na podstawie badań symulacyjnych oraz laboratoryjnych i przedstawione w rozdziałach 5 i 6.

Tabela 6. Wartości parametrów modelu identyfikowanych metodą quasi-Newtona dla zawężonych przedziałów wartości startowych

Nr próby		σ_1	σ_2	σ_3	σ_4	Czas obliczeń [s]
1	Wartości startowe	3,62	4,59	0,82	0,23	
	fminunc	3,59	4,66	0,61	0,07	13,52
2	Wartości startowe	3,91	4,98	0,84	0,21	
	fminunc	3,90	4,98	0,84	0,22	3,76
3	Wartości startowe	3,26	4,41	0,86	0,22	
	fminunc	3,28	4,42	0,36	0,11	25,11
4	Wartości startowe	3,93	4,73	0,85	0,26	
	fminunc	3,91	4,75	0,84	0,24	173,61
5	Wartości startowe	3,52	4,23	0,84	0,26	
	fminunc	3,37	4,56	0,85	0,26	5,65

Wartość średnia	3,61	4,67	0,70	0,18	44,33
Wariancja	0,07	0,04	0,04	0,01	
Odchylenie standardowe	0,26	0,19	0,19	0,08	

W rzeczywistości w systemie dynamicznym mogą wystąpić niepewności co do wyznaczonych metodami offline parametrów robota, w związku z tym na podstawie prac [35], [39], [98] zaproponowano adaptację parametrów modelu dynamicznego robota w czasie ruchu. W pracy [39] przedstawiono dwa prawa adaptacji parametrów modelu dynamicznego robota. Natomiast w autorskim systemie sterowania, przedstawionym w niniejszej rozprawie doktorskiej zastosowano adaptację parametrów metodą RLS (ang. Recursive Least Squares) na podstawie wyrażenia:

$$\hat{\sigma}(t) = \gamma^{-1} A^T(t) \Delta(t) - \gamma^{-1} \Gamma \hat{\sigma}(t) \quad (2.22)$$

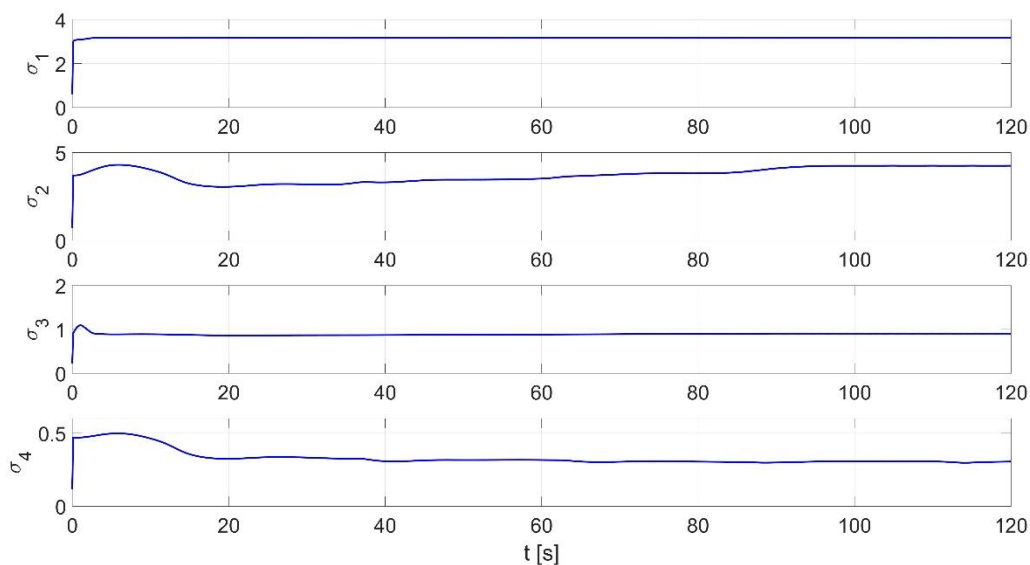
gdzie $\gamma \in \mathbb{R}^{4 \times 4}$ jest macierzą wzmocnień dla adaptacji parametrów, a $\hat{\sigma}(t)$ jest macierzą estymowanych parametrów modelu dynamicznego, $A(t)$ jest macierzą prędkości i przyspieszeń (patrz Załącznik B), a $\Delta(t)$ jest macierzą uchybów w postaci:

$$\Delta(t) = \begin{bmatrix} v_{xd}(t) \\ \omega_d(t) \end{bmatrix} - \begin{bmatrix} v_x(t) \\ \omega(t) \end{bmatrix} \quad (2.23)$$

Prawo adaptacji (2.22) poprzez wprowadzenie macierzy wzmocnień $\Gamma \in \mathbb{R}^{4 \times 4}$ daje możliwość wyłączenia aktualizacji parametrów modelu, gdy błąd jest mniejszy od zadanej wartości granicznej.

W celu sprawdzenia wpływu nieznanych parametrów modelu na początkowy błąd śledzenia trajektorii zadanej, proces identyfikacji online przeprowadzono w dwóch krokach, w których w procesie identyfikacji jako parametry startowe wykorzystano wartości losowe, a następnie zastąpiono je wartościami wyznaczonymi metodą offline.

W pierwszym kroku przeprowadzono identyfikację online dla losowych parametrów startowych opisanych rozkładem normalnym o wartości oczekiwanej $\mu = 1$, wariancji $\sigma^2 = 1$ w przedziale $\langle 0,1 \rangle$. Czas symulacji ustalono na 120s. Przykładowy przebieg procesu identyfikacji parametrów przedstawiono na Rys. 2.7. Wartości parametrów startowych i zidentyfikowanych metodą RLS przedstawiono w tabeli 7.

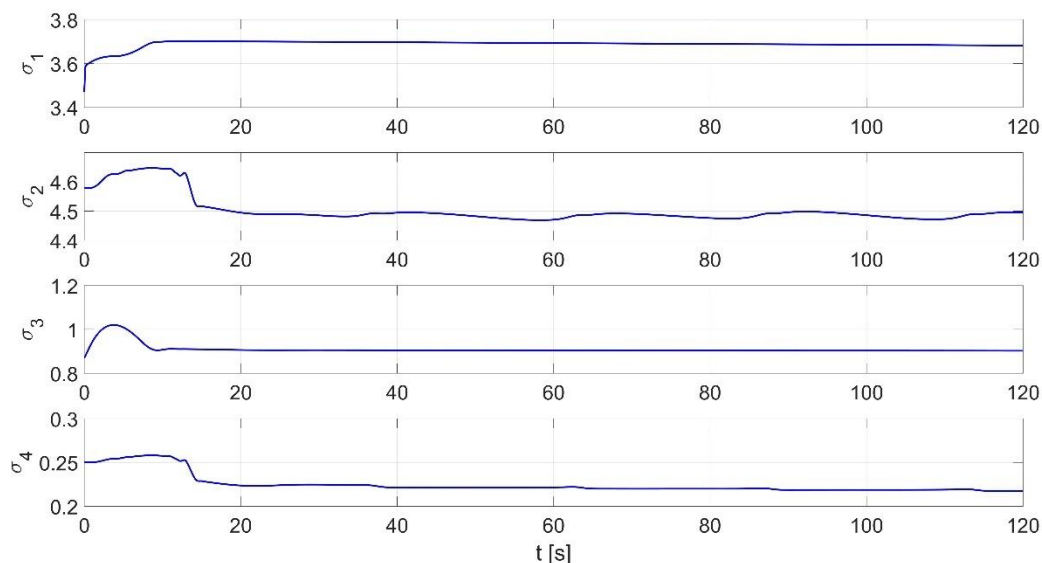


Rys. 2.7. Przykładowy przebieg procesu identyfikacji metodą RLS (próba nr 1 z Tabela 7)

Tabela 7. Przykładowe wartości parametrów modelu identyfikowanych metodą RLS z losowymi wartościami parametrów startowych

Nr próby		σ_1	σ_2	σ_3	σ_4
1	Wartości startowe	0,6	0,71	0,22	0,12
	RLS	3,16	4,15	0,92	0,3
2	Wartości startowe	0,93	0,73	0,49	0,58
	RLS	3,26	4,51	0,89	0,22
3	Wartości startowe	0,24	0,46	0,96	0,55
	RLS	3,84	4,34	0,86	0,25
4	Wartości startowe	0,52	0,23	0,49	0,62
	RLS	3,37	3,87	0,78	0,28
5	Wartości startowe	0,68	0,39	0,37	0,99
	RLS	3,32	4,16	0,76	0,24
Wartość średnia		3,39	4,21	0,84	0,26
Wariancja		0,06	0,05	0,00	0,00
Odchylenie standardowe		0,24	0,21	0,06	0,03

W drugim kroku procesu identyfikacji metodą RLS jako parametry startowe wykorzystano parametry zidentyfikowane metodą Levenberga-Marguardta. Na Rys. 2.8 przedstawiono przebieg procesu identyfikacji parametrów $\sigma_1, \dots, \sigma_4$. Wartości parametrów startowych i zidentyfikowanych metodą RLS wspomaganą metodą Levenberga-Marguardta przedstawiono w tabeli 8.



Rys. 2.8. Przykładowy przebieg procesu identyfikacji metodą RLS wspomaganą metodą Levenberga-Marguardta

Tabela 8. Wartości parametrów identyfikowanych metodą RLS wspomaganą metodami offline

		σ_1	σ_2	σ_3	σ_4
Nr próby	Wartości startowe	3,47	4,58	0,87	0,25
1	RLS + offline	3,68	4,49	0,90	0,22
2	RLS + offline	3,66	4,50	0,80	0,22
3	RLS + offline	3,62	4,51	0,85	0,23
4	RLS + offline	3,67	4,52	0,89	0,20
5	RLS + offline	3,62	4,54	0,83	0,24
Wartość średnia		3,65	4,51	0,85	0,22
Wariancja		0,00	0,00	0,00	0,00
Odchylenie standardowe		0,03	0,02	0,04	0,01

2.4. Walidacja modelu matematycznego robota TURTLEBOT 2

Opracowany model matematyczny robota TURTLEBOT 2 poddano walidacji. W tym celu w środowisku MATLAB/Simulink przeprowadzono symulację, na podstawie której wyznaczono trajektorię ruchu robota wywołaną sterowaniem prędkością liniową i kątową. Następnie trajektorię otrzymaną w wynik symulacji porównano z trajektorią wzorcową – wyznaczoną na podstawie przejazdu rzeczywistego robota. Do symulacji ruchu robota wykorzystano parametry fizyczne robota TURTLEBOT 2 zgodne z dokumentacją

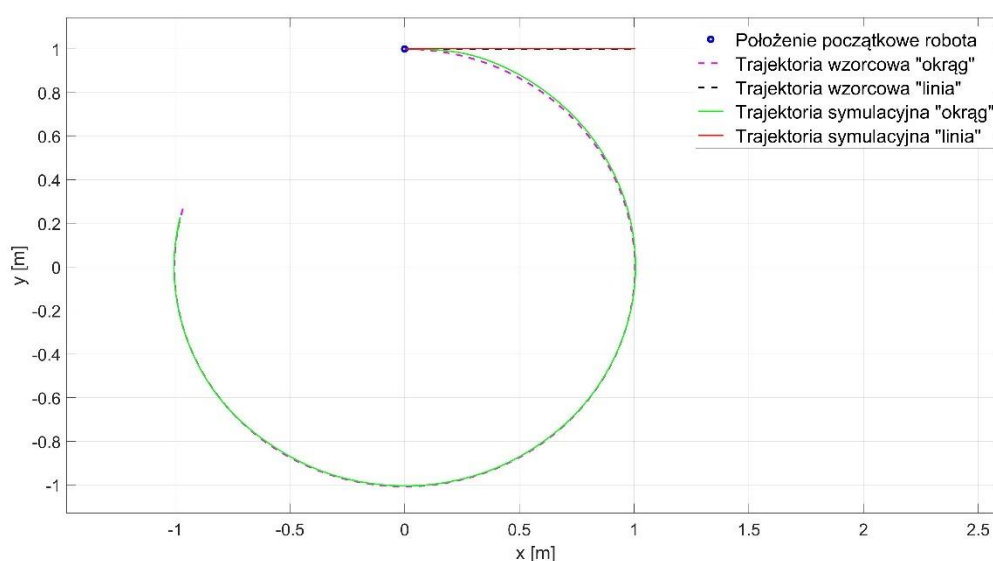
(przedstawione w rozdziale 2) oraz zidentyfikowane parametry modelu dynamicznego, których wartości wyznaczono w rozdziale 2.3.3.

Symulację ruchu przeprowadzono dla stałych wartości prędkości liniowej i kątowej w dwóch przypadkach:

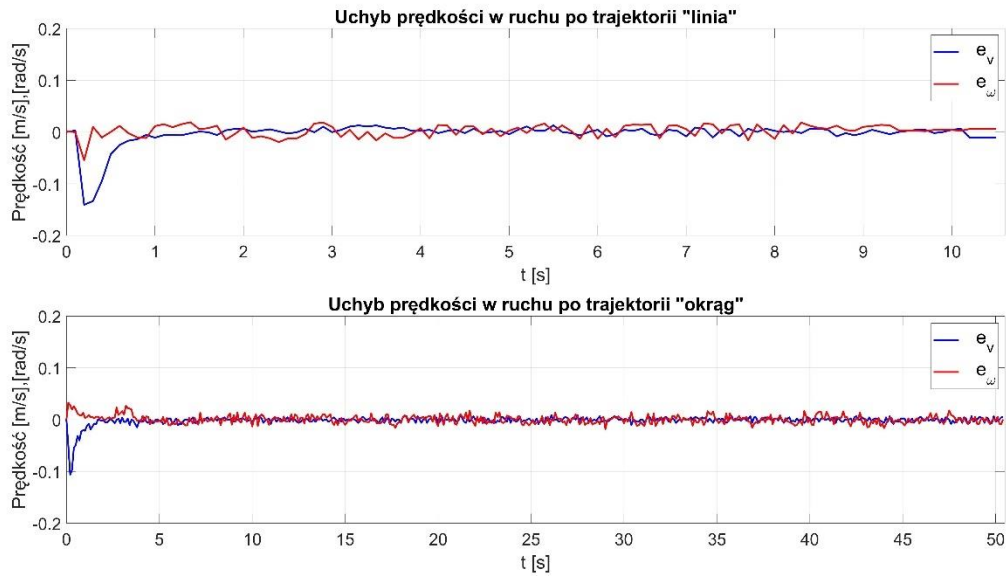
- ruchu po linii prostej dla $v_d = 0.1$ i $\omega_d = 0$,
- ruchu po okręgu o promieniu 1 m dla $v_d = 0.1$ i $\omega_d = 0.1$.

W symulacji pozycję i orientację początkową robota opisano wektorem współrzędnych konfiguracyjnych $q_0 = [0, 1, 0]$.

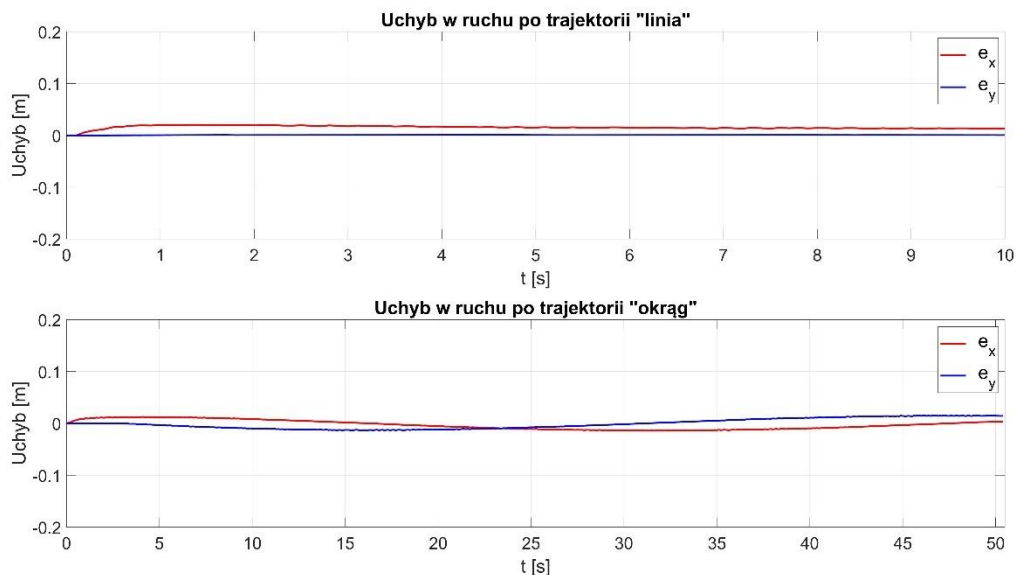
Wyniki walidacji przedstawiono w postaci trzech wykresów: porównania trajektorii symulacyjnej z trajektorią rzeczywistą (Rys. 2.9), uchybu prędkości robota – różnicy pomiędzy prędkościami otrzymanymi w wyniku symulacji a wzorcowymi (zarejestrowanymi podczas przejazdów) (Rys. 2.10), uchybów położenia – różnicy pomiędzy położeniem wzorcowym, a wyznaczonym symulacyjnie w danej chwili (Rys. 2.11).



Rys. 2.9. Walidacja modelu matematycznego robota – porównanie trajektorii symulacyjnej i wzorcowej



Rys. 2.10. Walidacja modelu matematycznego robota – uchyb prędkości



Rys. 2.11. Walidacja modelu matematycznego robota – uchyby położenia

W procesie walidacji wykazano, że wartości uchybów nie były większe niż przedstawione w tabeli poniżej.

Tabela 9. Wartości uchybów wyznaczonych podczas walidacji modelu matematycznego

Uchyby w ruchu po trajektorii „linia”	Uchyby w ruchu po trajektorii „okrąg”
- w kierunku osi x: $e_x = 0.012 \text{ m}$; - w kierunku osi y: $e_y = 0.001 \text{ m}$; - prędkości liniowej $e_v = 0.006 \frac{\text{m}}{\text{s}}$; - prędkości kątowej $e_\omega = 0.009 \frac{\text{rad}}{\text{s}}$	- w kierunku osi x: $e_x = 0.014 \text{ m}$; - w kierunku osi y: $e_y = 0.015 \text{ m}$; - prędkości liniowej $e_v = 0.004 \frac{\text{m}}{\text{s}}$; - prędkości kątowej $e_\omega = 0.012 \frac{\text{rad}}{\text{s}}$

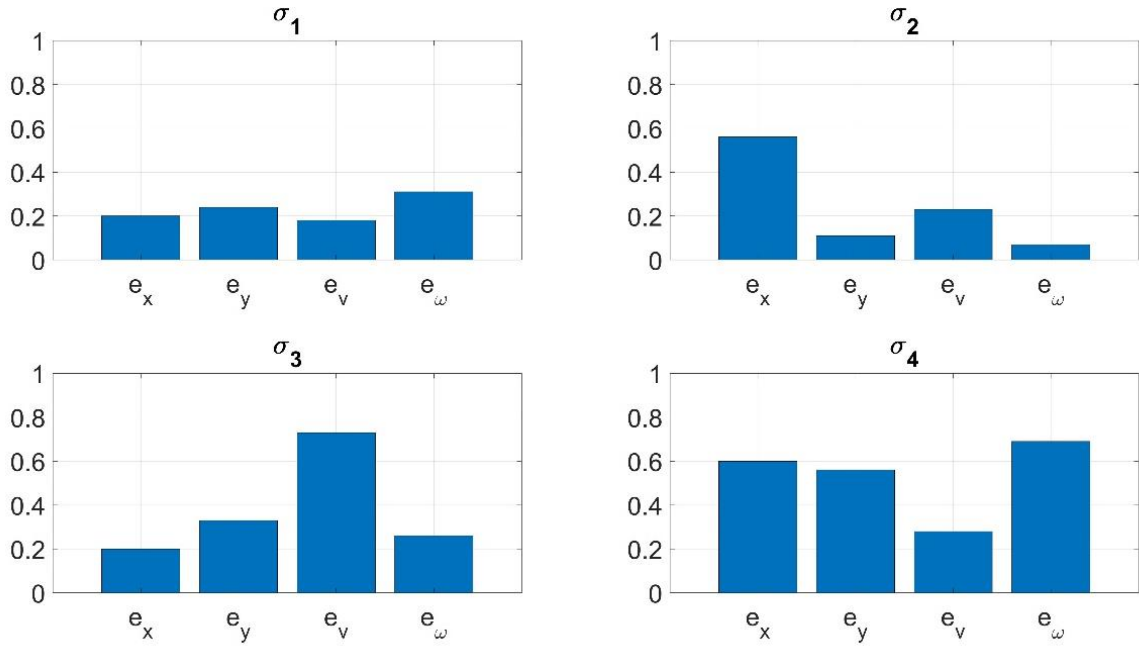
Na podstawie przeprowadzonej analizy porównawczej wyników można stwierdzić, że matematyczny model robota TURTLEBOT 2 został opracowany prawidłowo, a wyznaczone w drodze identyfikacji wartości jego parametrów są poprawne. Świadczą o tym oczekiwane odpowiedzi układu sterowania na zadawane wymuszenia.

W ostatnim etapie walidacji modelu, przeprowadzono analizę wrażliwości [13], [59], [93], która miała na celu sprawdzenie w jakim stopniu niedokładności wyznaczenia parametrów $\sigma_1, \dots, \sigma_4$ modelu dynamicznego robota wpływają na uchyby: pozycji e_x, e_y oraz prędkości liniowej e_v i kątowej e_ω . Matematycznie, wrażliwość funkcji kosztu względem wybranych parametrów jest równa pochodnej cząstkowej funkcji kosztu względem tych parametrów [95]. Analizę wrażliwości przeprowadzono wykorzystując aplikację Sensitivity Analyzer (narzędzie pakietu MATLAB/Simulink) oraz model matematyczny robota złożony z modelu kinematycznego (2.9) i dynamicznego (2.17). Jako zestaw danych wejściowych zastosowano wszystkie zbiory zidentyfikowanych parametrów $\sigma_1, \dots, \sigma_4$. W badaniu sprawdzono wrażliwość modelu w oparciu o cztery funkcje kosztu, które określono jako uchyby e_x, e_y, e_v, e_ω . Efektem przeprowadzonej analizy wrażliwości za pomocą narzędzia Sensitivity Analyzer było wyznaczenie współczynników korelacji, które określały siłę i kierunek związku pomiędzy dwiema zmiennymi. Wartości współczynników korelacji wyznaczono metodą Kendalla na podstawie wyrażenia [118]:

$$T = \frac{2}{n(n-1)} \sum_{i < j} \text{sgn}(x_i - x_j) \text{sgn}(y_i - y_j) \quad (2.24)$$

gdzie x_i, y_i są powiązаныmi danymi wejściowymi i wyjściowym, n jest liczebnością zbioru wejściowego. Wartość współczynnika korelacji T mieści się w przedziale $(-1, 1)$. Im większa jest jego wartość bezwzględna, tym silniejsza jest zależność liniowa między zmiennymi. 0 - oznacza brak liniowej zależności, 1 – oznacza zależność dodatnią, a -1 – oznacza zależność ujemną między cechami [127], [25], [13].

Wartości bezwzględne wyznaczonych współczynników korelacji w odniesieniu do uchybów jako funkcji kosztu przedstawiono na Rys. 2.12.



Rys. 2.12. Wartości bezwzględne współczynników korelacji parametrów modelu dynamicznego z uchybami

Analizując wykresy współczynników korelacji przedstawione na Rys. 2.12 można stwierdzić, że parametr modelu σ_1 nie ma znaczącego wpływu na uchyby. Parametr σ_2 jest silnie powiązany z uchybem śledzenia e_x . Parametr σ_3 ma znaczący wpływ na uchyb śledzenia prędkości liniowej e_v . Natomiast parametr σ_4 jest silnie skorelowany ze wszystkimi błędami. Otrzymane w wyniku analizy wrażliwości dane wskazują jednak na fakt, że na dokładne śledzenie trajektorii ma wpływ prawidłowe wyznaczenie wartości wszystkich parametrów $\sigma_1, \dots, \sigma_4$.

2.5. Opracowanie modelu teoretycznego grupy robotów o strukturze rozproszonej

Grupę współpracujących robotów można opisać poprzez wektor stanu grupy H , który jest złożeniem stanów poszczególnych robotów [24], [18]. Rozważając system sterowania grupą o strukturze rozproszonej, wektor stanu ciągłego $\dot{h}_i(t)$, każdego robota wchodzącego w skład grupy jest funkcją jego $\dot{h}_i(t)$ wektora wejściowego - sterowania $u_i(t)$, oraz wektora stanu $h_i(t)$.

Określając wektor stanu grupy rozproszonej jako:

$$H(t) = [h_1(t), h_2(t), \dots, h_n(t)]^T \quad (2.25)$$

oraz przyjmując równanie stanu każdego robota jako:

$$\dot{h}_i(t) = f_i(h_i(t), u_i(t)) \quad (2.26)$$

gdzie wektor sterowania u_i przyjmuje postać:

$$u_i(t) = k_i(h_i(t), e_i(t)) \quad (2.27)$$

a wektor uchybu e_i :

$$e_i(t) = h_{i0}(t) - h_i(t) \quad (2.28)$$

można sformułować równanie stanu rozproszonej grupy robotów jako:

$$\dot{H}(t) = F(H(t), U(t)), \quad \text{gdzie } U(t) = [u_1(t), u_2(t), \dots, u_n(t)]^T \quad (2.29)$$

W celu minimalizacji błędów sterowania ruchem grupy robotów o strukturze rozproszonej koniecznym jest wprowadzenie synchronizacji czasowej robotów, na co zwrócono uwagę w rozdziale 1.4. Rozpatrując grupę robotów TURTLEBOT 2 możemy określić wektor współrzędnych konfiguracyjnych każdego robota w postaci:

$$h_i(t_i) = [x_{is}(t_i), y_{is}(t_i), \theta_i(t_i), v_i(t_i), \omega_i(t_i)] \quad (2.30)$$

Przyjmując wektor sterowań z uwzględnieniem synchronizacji (opóźnień czasowych) w postaci:

$$u_i(t_i + \Delta t_i) = [v_{id}(t_i + \Delta t_i), \omega_{id}(t_i + \Delta t_i)] \quad (2.31)$$

gdzie poszczególne wejścia odpowiadają za prędkość liniową v_i i prędkość kątową ω_i można sformułować równanie stanu uwzględniające synchronizację czasową dla każdego robota w postaci:

$$\begin{aligned} \dot{h}_i = f_i(h_i(t_i), u_i(t_i + \Delta t_i)) \rightarrow \begin{bmatrix} \dot{x}_i(t_i) \\ \dot{y}_i(t_i) \\ \dot{\psi}_i(t_i) \\ \dot{v}_i(t_i) \\ \dot{\omega}_i(t_i) \end{bmatrix} &= \begin{bmatrix} v_i(t_i + \Delta t_i) \cos \psi_i(t_i + \Delta t_i) \\ \omega_i(t_i + \Delta t_i) \sin \psi_i(t_i + \Delta t_i) \\ \omega_i(t_i + \Delta t_i) \\ -\frac{\theta_{3i}}{\theta_{1i}} v_i(t_i + \Delta t_i) \\ -\frac{\theta_{4i}}{\theta_{2i}} \omega_i(t_i + \Delta t_i) \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_{1i}} & 0 \\ 0 & \frac{1}{\theta_{2i}} \end{bmatrix} \begin{bmatrix} v_{id}(t_i + \Delta t_i) \\ \omega_{id}(t_i + \Delta t_i) \end{bmatrix} \end{aligned} \quad (2.32)$$

oraz równanie stanu grupy rozproszonej w postaci:

$$\dot{H} = F(H(t_i), U(t_i + \Delta t_i)) \quad (2.33)$$

$$\text{gdzie } U(t) = [u_1(t_1 + \Delta t_1), u_2(t_2 + \Delta t_2), \dots, u_n(t_n + \Delta t_n)]^T$$

Przedstawione równania modelują ciągłe zachowanie każdego z robotów, a co za tym idzie ciągłe zachowanie grupy podczas realizacji wspólnego zadania [24], [18], [114], [88], [113].

Rozdział 3

BADANIA ROZPOZNAWCZE DOKŁADNOŚCI POZYCJONOWANIA ROBOTÓW W GRUPIE O STRUKTURZE ROZPROSZONEJ

Obecnie obserwuje się bardzo duży wzrost zakupów i użytkowania urządzeń mobilnych takich jak laptopy, tablety, smartfony, można stwierdzić, że właśnie komunikacja bezprzewodowa jest najczęściej wykorzystywana przez zwykłego użytkownika. Ze względu na wiele zalet transmisji bezprzewodowej ten rodzaj komunikacji jest również najczęściej stosowany w robotyce mobilnej. Pozwala na osiągnięcie pełnej mobilności przez robota w budynkach, jak i w przestrzeni otwartej. Transmisję bezprzewodową wykorzystuje się do przesyłania danych sterujących urządzeniem jak i danych zwrotnych np. obrazu z kamer, dźwięku, danych o środowisku, w którym pracuje robot. Transmisja bezprzewodowa nie pozostaje jednak bez wad. Szczególną uwagę należy zwrócić na środowiska, w których występuje wiele sieci WiFi np. biurowce, fabryki. Są to duże skupiska urządzeń generujących wiele zakłóceń oraz powodujące tłumienie. W przypadku komunikacji bezprzewodowej w takich miejscach należy spodziewać się błędów transmisji danych jak i opóźnień sygnałów. Opóźnienie transmisji danych jest istotnym problemem w sterowaniu grupą robotów. Duże opóźnienie danych sterujących może powodować niedokładności trajektorii ruchu grupy robotów, której zachowanie jest konieczne do prawidłowego wykonania postawionego zadania.

W celu rozpoznania charakteru i wielkości błędów pozycjonowania robotów w grupie o strukturze rozproszonej, wywołanych opóźnieniem transmisji danych sterujących, przeprowadzono dwa badania laboratoryjne z wykorzystaniem robotów Turtlebot 2 oraz dwa badania symulacyjne w środowisku MATLAB/Simulink i MSC Adams. Badania laboratoryjne dotyczyły zbadania wpływu ilości robotów w grupie o strukturze rozproszonej oraz wpływu rozmieszczenia przeszkód w środowisku pracy grupy na opóźnienia sygnałów sterujących. Badania symulacyjne dotyczyły sprawdzenia wpływu opóźnień sygnałów sterujących o określonych w badaniach laboratoryjnych charakterze i wartościach na ruch grupy i wzajemne oddziaływanie robotów w grupie o strukturze rozproszonej.

3.1. Badania laboratoryjne wpływu ilości robotów w grupie i rozmieszczenia przeszkód środowiskowych na opóźnienia sygnałów sterujących

Transmisja bezprzewodowa Wi-Fi jest obecnie najczęściej wykorzystywanym sposobem transmisji danych w robotyce mobilnej. Komunikacja bezprzewodowa Wi-Fi jednak jest bardzo podatna na zakłócenia zewnętrzne. Częstym przykładem zakłóceń może być interferencja z sieciami pracującymi na tym samym paśmie częstotliwości. Rezultatem tego może być obniżenie przepustowości sieci, a w skrajnych przypadkach całkowite zaprzestanie pracy. W przypadku robotyki mobilnej, gdzie wymagana jest szybka transmisja danych, brak zabezpieczenia systemu przed tego typu błędami może powodować niepoprawną realizację zadania oraz wzajemne kolizje robotów w grupie lub kolizje z przeszkodami.

Celem badań laboratoryjnych, było określenie charakteru i wartości opóźnień czasowych sygnałów sterujących rzeczywistymi robotami Turtlebot 2 tworzącymi grupę o strukturze rozproszonej. Przeprowadzono dwa badania. W badaniu pierwszym sprawdzono wpływ ilości robotów Turtlebot 2 zintegrowanych w grupę na opóźnienia czasowe sygnałów sterujących wynikające ze zmiennego obciążenia sieci. W badaniu drugim sprawdzono wpływ rozmieszczenia przeszkód środowiskowych na opóźnienia sygnałów sterujących.

Badania laboratoryjne opóźnień transmisji sygnałów sterujących robotami mobilnymi przeprowadzono w laboratorium robotyki Zakładu Mechatroniki ITR WML WAT. Wyniki badań, które częściowo opublikowano w pracy [101] są wymagane do opracowania metody sterowania grupą robotów realizujących zadania transportowe, w związku z tym rozpatrywano transmisję sygnałów sterujących wyłącznie w środowisku indor (środowisku wewnętrznym).

3.1.1. Zakres, metodyka i wyniki badań laboratoryjnych wpływu ilości robotów w grupie o strukturze rozproszonej na opóźnienia sygnałów sterujących

Stanowisko badawcze składało się z trzech robotów Turtlebot 2 oraz komputera zarządzającego PC jako jednostki wysyłającej komendy startowe i wizualizującej dane. Dwa roboty zintegrowane były z laptopami PC Lenovo Yoga 300 natomiast trzeci z laptopem Asus F200M. Laptopy Lenovo i Asus umiejscowione były na robotach i służyły jako jednostki sterujące i zarządzające podzespołami robota. Na wszystkich komputerach zainstalowano system Linuks Ubuntu i platformę ROS. W Tabeli 10 przedstawiono podstawowe parametry techniczne zintegrowanych z robotami laptopów. Widok stanowiska badawczego złożonego z trzech robotów TURTLEBOT 2 i komputera zarządzającego przedstawiono na Rys. 3.1.



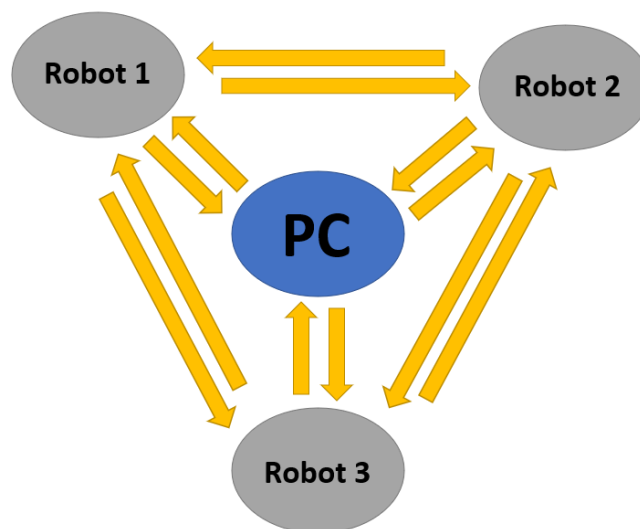
Rys. 3.1. Widok stanowiska badawczego grupy robotów TURTLEBOT 2

Tabela 10. Dane techniczne komputerów pokładowych robotów TURTLEBOT 2

Parametr	Lenovo Yoga 300	Asus F200M
Procesor	Intel Celeron N3060 1.6-2.16 GHz	Intel Celeron N2830 2.16 GHz
RAM	2 GB DDR3L	2 GB DDR3
Karta graficzna	Intel HD Graphics	Intel HD Graphics
Karta Wi-Fi	Intel	Qualcomm Atheros

W celu pomiaru opóźnień czasowych sygnałów sterujących robotami opracowano program komputerowy, który realizuje transmisję sygnałów pomiędzy robotami. Na komputerze zarządzającym został uruchomiony serwer - program zarządzający transmisją danych sterujących, na komputerach pokładowych robotów uruchomiono programy pracujące jako klient. Roboty mogą połączyć się z serwerem i nawiązać komunikację. Po uruchomieniu transmisji serwer wysyłał taką samą ramkę danych do wszystkich podłączonych robotów. Kiedy roboty otrzymały dane natychmiast w odpowiedzi dla komputera sterującego wysyłały ramkę tych samych danych. Serwer rejestrował czas wysłania danych do robotów oraz czas otrzymania odpowiedzi od każdego z robotów. Następnie program obliczał opóźnienie pomiędzy czasem wysłania danych i czasem uzyskania odpowiedzi. Połączenie sieciowe pomiędzy komputerami zrealizowano w trybie Ad-Hoc, który jest znany również jako *peer-to-peer*. Tryb Ad-Hoc umożliwia bezpośrednią komunikację urządzeń bez konieczności wykorzystywania punktu dostępowego. W tym trybie nie ma sztywno określonej infrastruktury sieci. Aby komputery mogły nawiązać komunikację, powinny znajdować się w zasięgu swojej

sieci. Do propagacji transmisji sygnałów zastosowano ruter NETGEAR N150 Wireless zgodny ze standardem IEEE 802.11n pozwalający na transmisje z prędkością 100 Mbps. Strukturę zbudowanej sieci przedstawiono na Rys. 3.2.



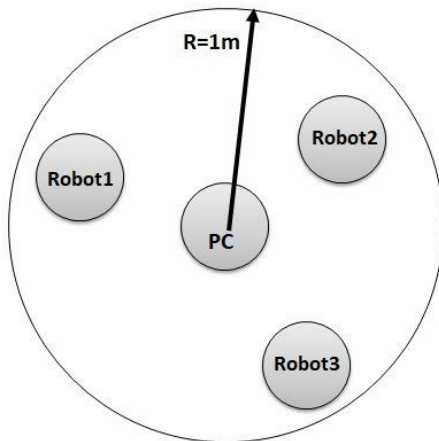
Rys. 3.2. Struktura sieci bezprzewodowej zastosowanej w badaniach

Badania przeprowadzono w warunkach laboratoryjnych dla dwóch przypadków, uwzględniając zmianę ilości robotów w grupie oraz przeszkody otoczenia (w postaci ścian i szyb) na drodze transmisji sygnałów sterujących. Podczas badań ramkę danych wysyłało z częstotliwością 1000 Hz celem wyznaczenia wartości opóźnień z dokładnością do 1 ms. Podczas testów rejestrowano czas od wysłania danych z komputera sterującego do otrzymania odpowiedzi zwrotnej od wszystkich robotów.

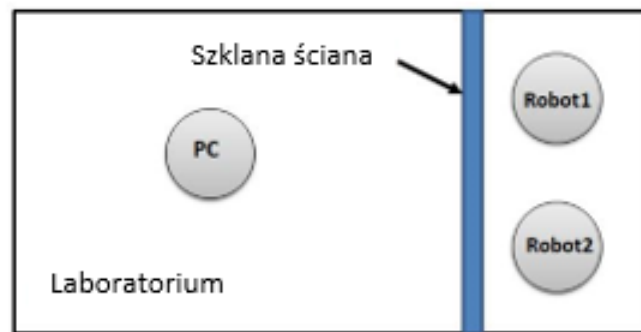
Pierwszy przypadek badań dotyczył sprawdzenia wpływu ilości zintegrowanych w grupie robotów na przepustowość sieci oraz wielkość opóźnień czasowych ich sygnałów sterujących. Roboty ustawiono w odległości nie przekraczającej 1 m od komputera sterującego, jak przedstawiono na Rys. 3.3a, a następnie mierzono opóźnienie sygnałów sterujących dla jednego, dwóch oraz trzech podłączonych do sieci robotów.

W drugim przypadku roboty umieszczono za przeszkodą w postaci szklanej ściany, jak przedstawiono na Rys. 3.3b. Najpierw sprawdzono opóźnienia czasowe sygnałów sterujących dla jednego podłączonego robota, a następnie dla dwóch i trzech podłączonych robotów.

a)

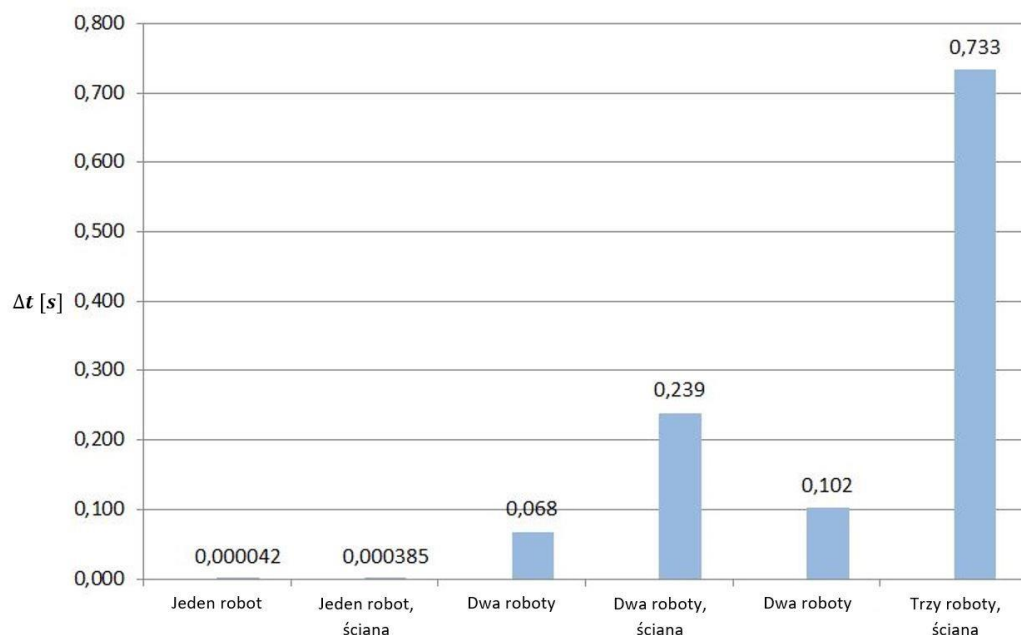


b)



Rys. 3.3. Rozmieszczenie robotów w badaniach: a) roboty umieszczone w niewielkiej odległości, b) roboty oddzielone szklaną przeszkodą

Zaimplementowane rozwiązanie programowe pozwoliło na zidentyfikowanie charakteru oraz poziomu występujących w systemie zdecentralizowanym opóźnień sygnałów sterujących. Wyniki pomiarów opóźnień czasowych sygnałów sterujących przedstawiono na Rys. 3.4. Umieszczono na nim wartości średnie opóźnień czasowych sygnałów sterowania.



Rys. 3.4. Opóźnienie czasowe sygnałów sterujących w zależności od ilości robotów w grupie i przeszkód pomiędzy robotami

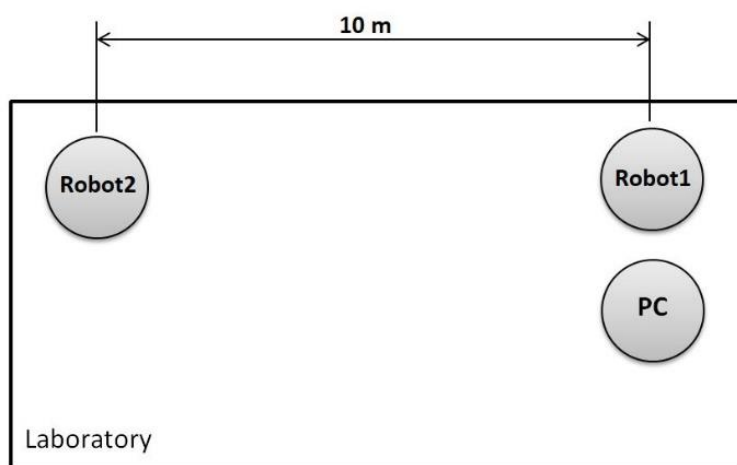
W przypadku podłączenia jednego robota do sieci można zaobserwować bardzo małe opóźnienie na poziomie kilkudziesięciu mikrosekund oraz wartość średnią równą 42 μs . Pojawienie się przeszkody na drodze transmisji podniosło wartość opóźnień o rząd wielkości,

a wartość średnia z pomiarów była równa 385 μ s. W przypadku podłączenia dwóch robotów otrzymano znaczny wzrost wartości opóźnień z poziomu mikrosekund do poziomu milisekund. W przypadku robotów znajdujących się blisko siebie otrzymano wartość średnią opóźnień na poziomie 68 ms. W przypadku wystąpienia przeszkody na drodze transmisji otrzymano wartość średnią o rząd wielkości większą, równą 239 ms. Podłączenie do sieci trzeciego robota wywołało wzrost wartości średnie do poziomu 102 ms dla robotów w bliskiej odległości oraz 733 ms w przypadku uwzględnienia przeszkody na drodze transmisji.

3.1.2. Zakres, metodyka i wyniki badań wpływu rozmieszczenia przeszkód w środowisku pracy grupy robotów na opóźnienia sygnałów sterujących

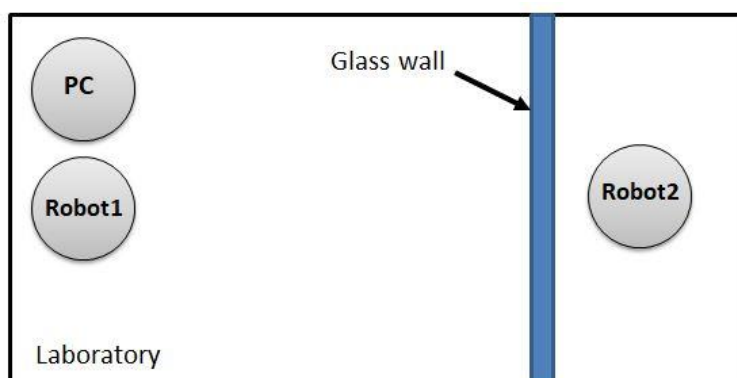
Badania doświadczalne wpływu rozmieszczenia przeszkód w środowisku pracy grupy robotów o strukturze rozproszonej na opóźnienia sygnałów sterujących przeprowadzono z wykorzystaniem stanowiska badawczego i konfiguracji sieci przedstawionych w rozdziale 3.1 na Rys. 3.1 i Rys. 3.2. Ze względu na wykazany we wcześniejszych testach wpływ sprzętu sieciowego do badań wpływu rozmieszczenia przeszkód w środowisku pracy grupy, wykorzystano wyłącznie dwa roboty, które wyposażone były w komputery Lenovo. Pozwoliło to uzyskać wyniki bez dodatkowych opóźnień wynikających z różnic sprzętowych układu pomiarowego.

Podczas badań zrealizowano trzy warianty konfiguracji środowiska pracy, w którym pracowały roboty. Pierwszy wariant dotyczył sprawdzenia wpływu odległości pomiędzy robotami na wartość opóźnień czasowych. W tym celu dwa roboty umieszczono w tym samym pomieszczeniu co ruter i komputer sterujący, lecz Robot 1 umieszczony był w sąsiedztwie rutera, Robot 2 umieszczony był w odległości 10 m od rutera jak przedstawiono na Rys. 3.5.



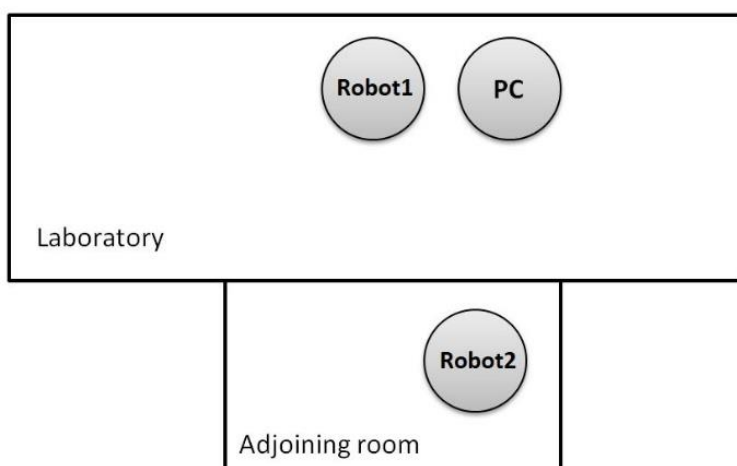
Rys. 3.5. Rozmieszczenie grupy robotów z jednym robotem w pobliżu rutera

Wariant drugi badań dotyczył sprawdzenia wpływu rozmieszczenia robotów w różnych pomieszczeniach rozdzielonych szklaną ścianą, w zasięgu tej samej sieci. Pierwszego robota pozostawiono w pomieszczeniu, w którym znajdował się ruter, drugiego przeniesiono do sąsiedniego zamkniętego pomieszczenia. Na drodze transmisji sygnału pomiędzy Robotem 1, a ruterem pojawiła się przeszkoda w postaci szklanej ściany. Widok rozmieszczenia robotów w laboratorium podczas tego testu przedstawiono na Rys. 3.6.



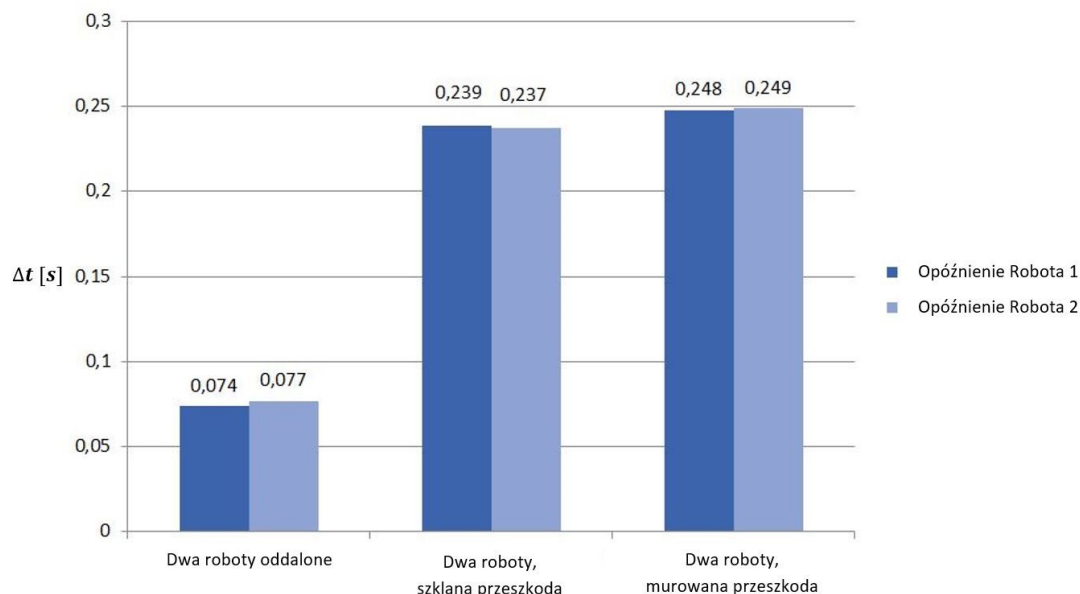
Rys. 3.6. Rozmieszczenie grupy robotów z uwzględnieniem szklanej przeszkody

Wariant trzeci badań dotyczył sprawdzenia wpływu murowanej ściany ustawionej na drodze transmisji sygnałów sterujących pomiędzy Robotem 2 i komputerem sterującym. W tym celu Robota 2 umieszczono w zamkniętym pomieszczeniu, odgrodzonym murowaną ścianą od laboratorium, znajdującym się w zasięgu wspólnej sieci Wi-Fi. Drugiego robota pozostawiono w bliskiej odległości od komputera sterującego oraz rutera. Rozmieszczenie robotów w laboratorium podczas tego testu przedstawiono na Rys. 3.7.



Rys. 3.7. Rozmieszczenie grupy robotów z jednym robotem w innym pomieszczeniu

Wyniki badań w postaci porównania opóźnień czasowych transmisji sygnałów sterujących przedstawiono na Rys. 3.8



Rys. 3.8. Porównanie opóźnień czasowych sygnałów sterujących między dwoma robotami

Na podstawie pomiarów przedstawionych jako wariant pierwszy, wyznaczono średnie wartości opóźnień czasowych: dla Robot 1 opóźnienie 74 ms i dla Robot 2 opóźnienie 77 ms. Wyniki tego testu pozwalają stwierdzić, że duże odległości pomiędzy robotami (w zakresie jednego dużego pomieszczenia) nie wnoszą różnicy w czasie otrzymania odpowiedzi robotów przez komputer sterujący.

Analizując wyniki badań zrealizowanych jako wariant drugi i trzeci można stwierdzić, że przeszkody w postaci szklanych i murowanych ścian na drodze transmisji nie wywołują różnicy opóźnienia czasowego transmitowanych sygnałów pomiędzy poszczególnymi robotami.

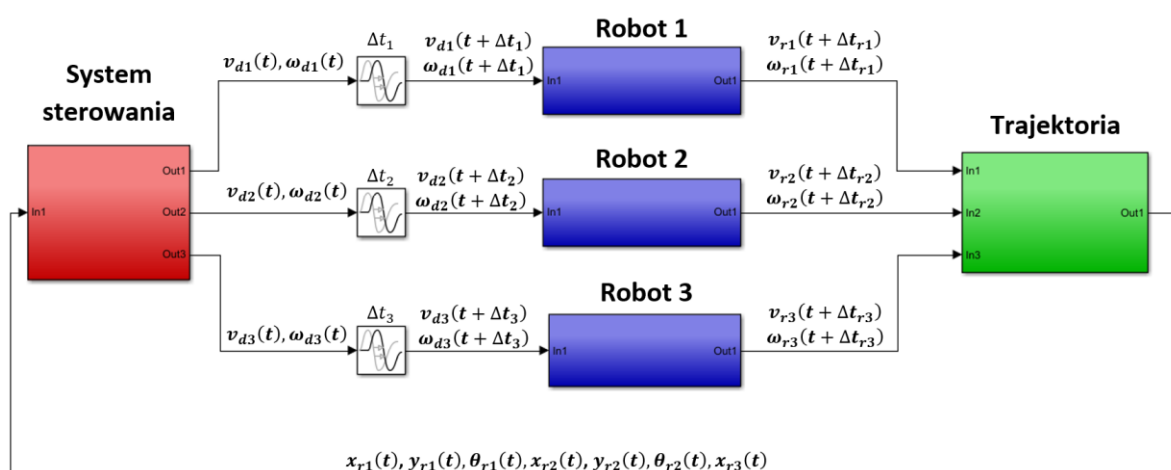
3.2. Badania symulacyjne wpływu opóźnień sygnałów sterujących na ruch grupy robotów o strukturze rozproszonej

W przypadku grupy robotów o strukturze rozproszonej, w której roboty pozostają w ciągłej komunikacji między sobą, opóźnienie czasowe sygnałów sterujących ma znaczący wpływ na podejmowanie przez algorytmy sterujące decyzji oraz przydzielanie zadań członkom grupy. Podczas badań symulacyjnych sprawdzono wpływ braku synchronizacji czasowej robotów, występującej na różnych etapach wykonywania zadania, na dokładność pozycjonowania robotów w grupie (częściowe wyniki badań zostały już opublikowane [104], [100]).

Do badań symulacyjnych przyjęto model matematyczny robota Turtlebot 2 opisany równaniami (2.9) i (2.17). W badaniach zastosowano sterowanie kinematyczne polegające na zadawaniu stałych i znanych wartości prędkości liniowej i kątowej.

3.2.1. Zakres i metodyka badań wpływu opóźnień sygnałów sterujących na ruch grupy robotów

Badania wpływu opóźnień sygnałów sterujących na ruch grupy robotów o strukturze rozproszonej przeprowadzono w środowisku MATLAB/Simulink. gdzie zrealizowano symulację wpływu opóźnień sygnałów sterujących (prędkości liniowej $v_{di}(t)$ i kątowej $\omega_{di}(t)$ o stałych wartościach) na ruch grupy o strukturze rozproszonej. Roboty (opisane modelem matematycznym przedstawionym w rozdziałach 2.3.1 i 2.3.2) ustawiono w grupę tworzącą trójkąt równoboczny o długości boku 1m. Środki geometryczne robotów umieszczono w wierzchołkach trójkąta. Na sygnały sterujące nakładano opóźnienie czasowe z przedziału od 300 ms do 3000 ms. Zakres wartości opóźnień dobrano na podstawie badań opóźnienia czasowego dla różnych konfiguracji bezprzewodowego połączenia sieciowego, opisanych w pracach [131] i [132], a następnie przeprowadzono analizę wrażliwości układu na opóźnienia. Badania symulacyjne przeprowadzono dla minimalnych i maksymalnych wartości opóźnień z wybranego przedziału. W symulacji obserwowano odległości pomiędzy robotami, a ich zmiana w stosunku do odległości zadeklarowanej wskazywała na błąd w szyku grupy – nie utrzymanie założonej formacji. Schemat blokowy symulacji przedstawiono na Rys. 3.9.



Rys. 3.9. Schemat blokowy symulacyjnego badania wpływu opóźnień czasowych sygnałów sterujących na ruch grupy o strukturze rozproszonej

Schemat symulacji złożony jest z trzech głównych bloków:

- systemu sterowania, w którym zadawane jest wymuszenie i kształtowane są sygnały sterujące ruchami poszczególnych robotów (stałe wartości prędkości liniowej $v_{di}(t)$ i kątowej $\omega_{di}(t)$),
- bloku zakłóceń, w którym na sygnały sterujące nakładane jest opóźnienie czasowe,
- bloku robotów, w których zaimplementowane są modele matematyczne robotów (przedstawione w rozdziałach 2.3.1 i 2.3.2),
- bloku trajektorii służącego do wizualizacji wyników symulacji.

W badaniach symulacyjnych wszystkie roboty podczas ruchu po linii prostej poruszały się z prędkością liniową $v_d(t) = 0.4 \frac{m}{s}$ i kątową $\omega_d(t) = 0 \frac{rad}{s}$, a po łuku z prędkością liniową $v_d(t) = 0.4 \frac{m}{s}$ i kątową $\omega_d(t) = 0.4 \frac{rad}{s}$. W modelach matematycznych robotów uwzględniono ograniczenie przyspieszenia do wartości $a_v(t) = 0.8 \frac{m}{s^2}$ dla przyspieszenia liniowego oraz $a_\omega(t) = 0.8 \frac{rad}{s^2}$ dla przyspieszenia kąowego, co odpowiada ograniczeniom rzeczywistych robotów TURTLEBOT 2. Opóźnienie czasowe nakładane było na komendy sterujące Robotu 3 - w chwili rozpoczęcia ruchu lub w chwili zmiany jego parametrów ruchu. Badania przeprowadzono dla dwóch scenariuszy sterowania:

- a) przejazd zadanej odległości ruchem prostoliniowym następnie zmiana orientacji o zadany kąt podczas ruchu,
- b) przejazd ruchem prostoliniowym i wymuszenie niespodziewanej zmiany orientacji o zadany kąt w trakcie ruchu prostoliniowego.

W scenariuszu a) cała trajektoria była znana przed rozpoczęciem ruchu. Zadaniem końcowym był przejazd zadanej trajektorii więc opóźnienie na sygnał sterujący nakładano przed rozpoczęciem ruchu. W scenariuszu b), w trakcie realizacji zadania wprowadzono nagłą zmianę wymagań, więc opóźnienie na sygnał sterujący zostało nałożone przed rozpoczęciem ruchu prostoliniowego oraz przed wymuszeniem zmiany orientacji. Scenariusz b), odpowiada sytuacji związanej z nagłym wykryciem przeszkody i koniecznością jej ominięcia przez grupę robotów.

Podczas badań mierzono uchyb położenia robota $e_{xy}(t)$, który był różnicą pomiędzy położeniem zadany, a wynikającym z obliczeń symulacyjnych oraz uchyb odległości pomiędzy robotami $e_{ij}(t)$.

Uchyb położenia wyznaczono z zależności:

$$e_{xy}(t) = \sqrt{(x_d(t) - x_r(t))^2 + (y_d(t) - y_r(t))^2} \quad (3.1)$$

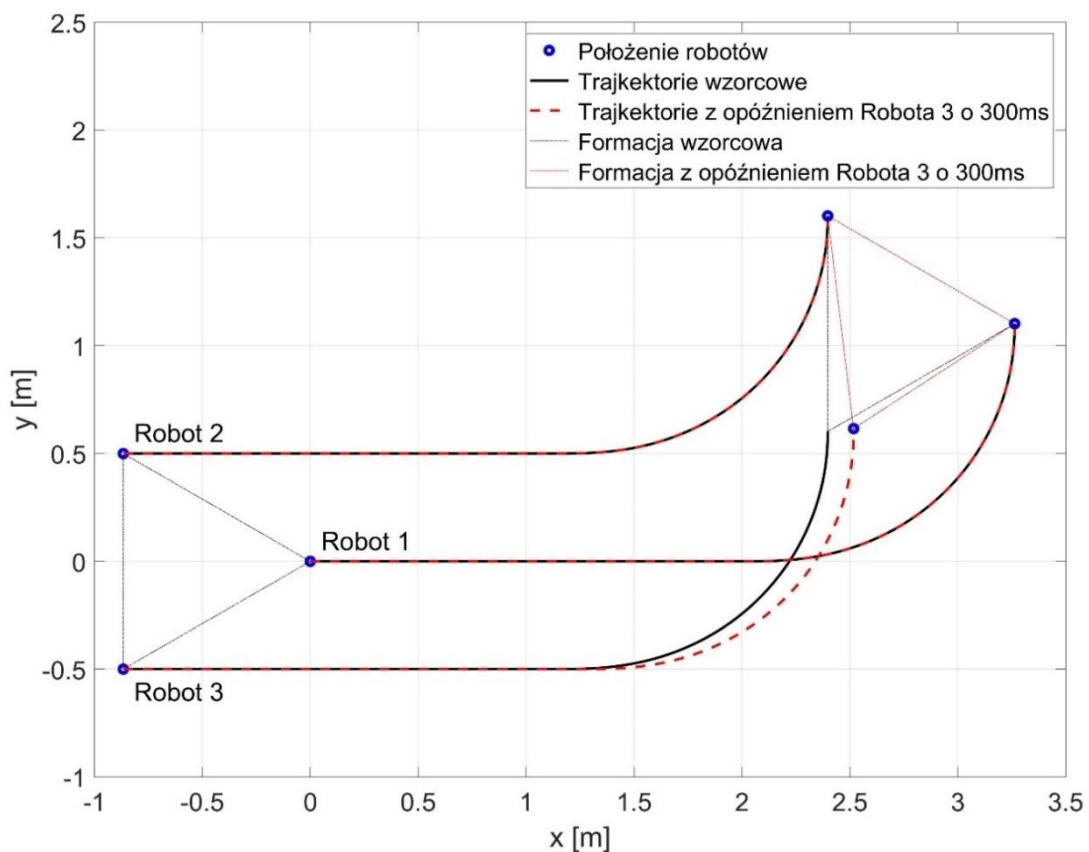
a uchyb odległości pomiędzy robotami z zależności:

$$e_{ij}(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \quad (3.2)$$

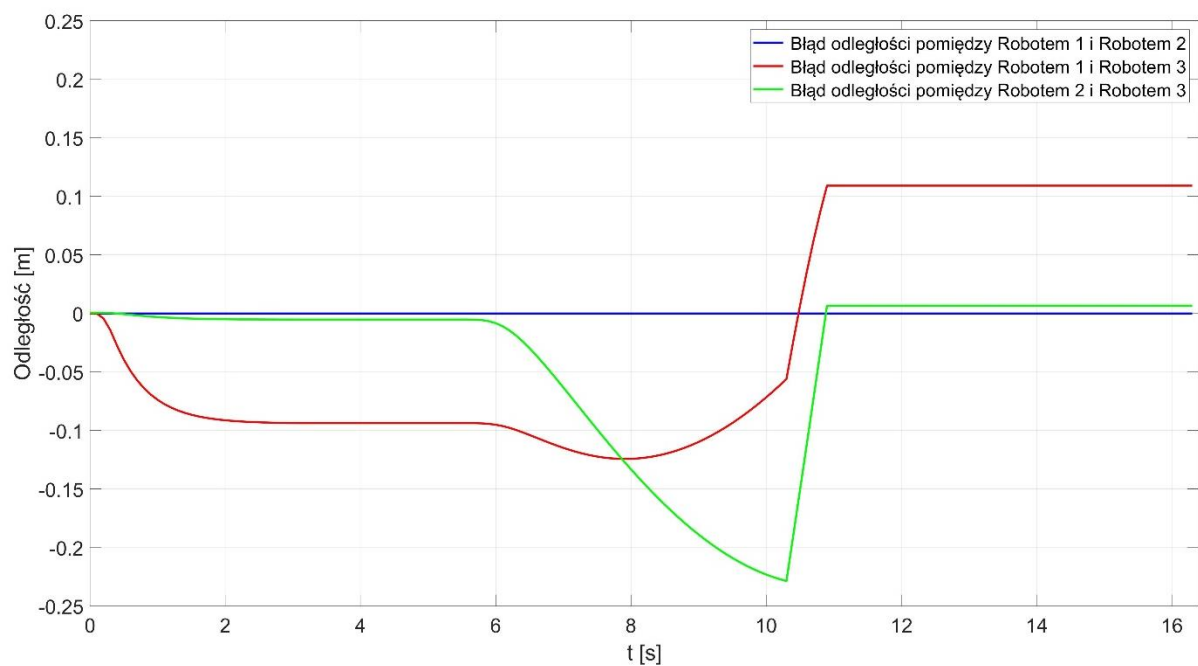
3.2.2. Wyniki badań wpływu opóźnień sygnałów sterujących na ruch grupy robotów

Wyniki symulacji ruchu grupy robotów o strukturze rozproszonej, zrealizowanych zgodnie ze scenariuszami a) i b), przedstawiono w postaci porównania trajektorii wzorcowych (bez opóźnień sygnałów sterujących) z trajektoriami robotów uwzględniającymi opóźnienia sygnałów sterujących: dla 300 ms (Rys. 3.10) oraz dla 3000 ms (Rys. 3.12). Dokonano także pomiarów błędu odległości pomiędzy poszczególnymi robotami wchodzącymi w skład grupy, co przedstawiono na wykresach: dla opóźnienia 300 ms (Rys. 3.11) oraz dla opóźnienia 3000 ms (Rys. 3.13).

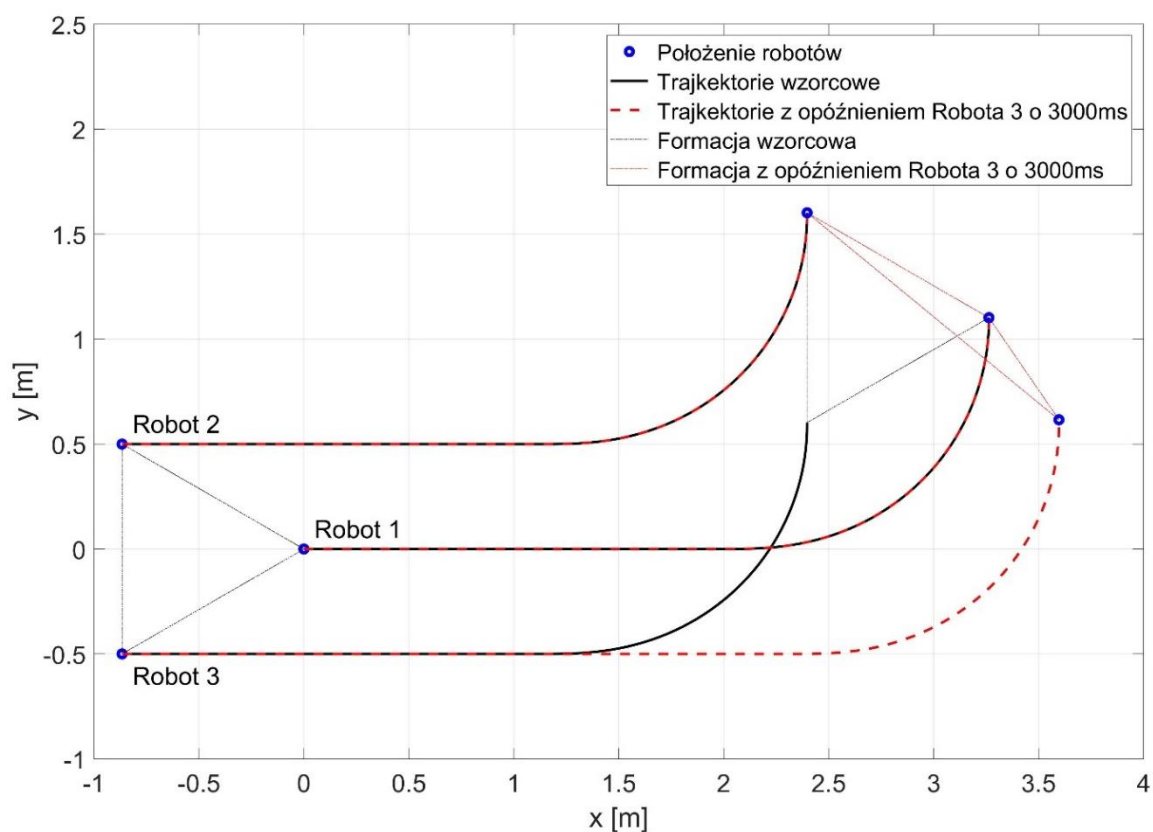
Scenariusz a) - przejazd odległości 2m ruchem prostoliniowym następnie zmiana orientacji o kąt $\frac{\pi}{2}$ podczas ruchu



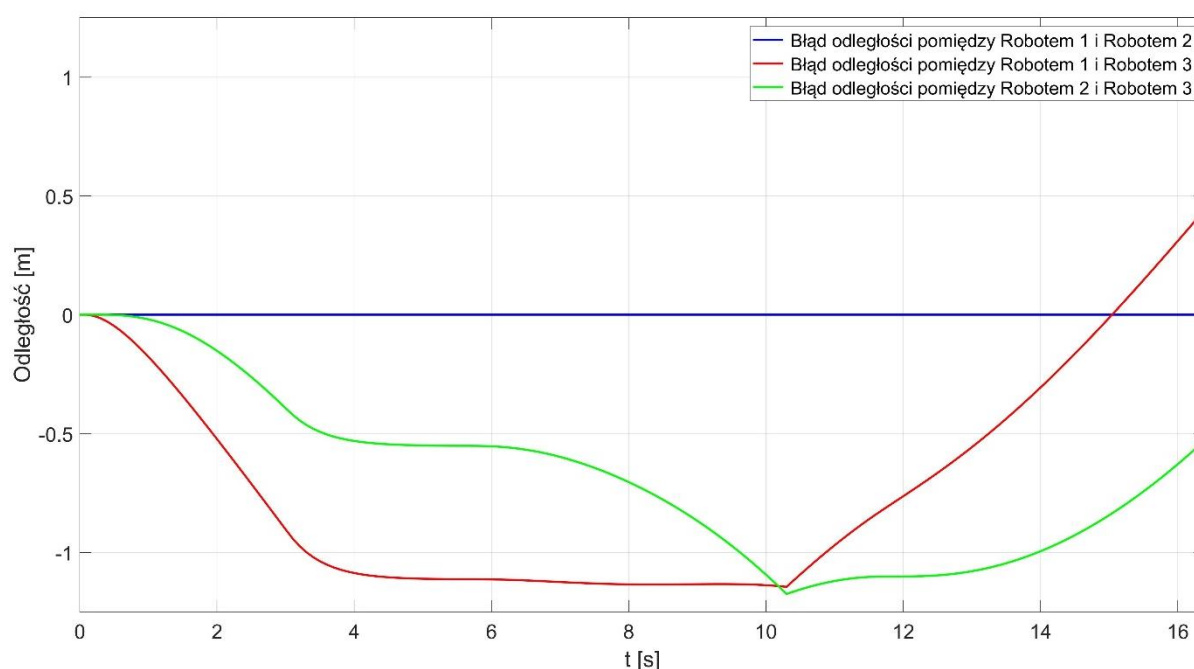
Rys. 3.10. Wizualizacja trajektorii wyznaczonej na podstawie scenariusza a) badań dla opóźnienia 300 ms



Rys. 3.11. Błąd odległości pomiędzy robotami wyznaczony na podstawie scenariusza a) badań dla opóźnienia 300 ms



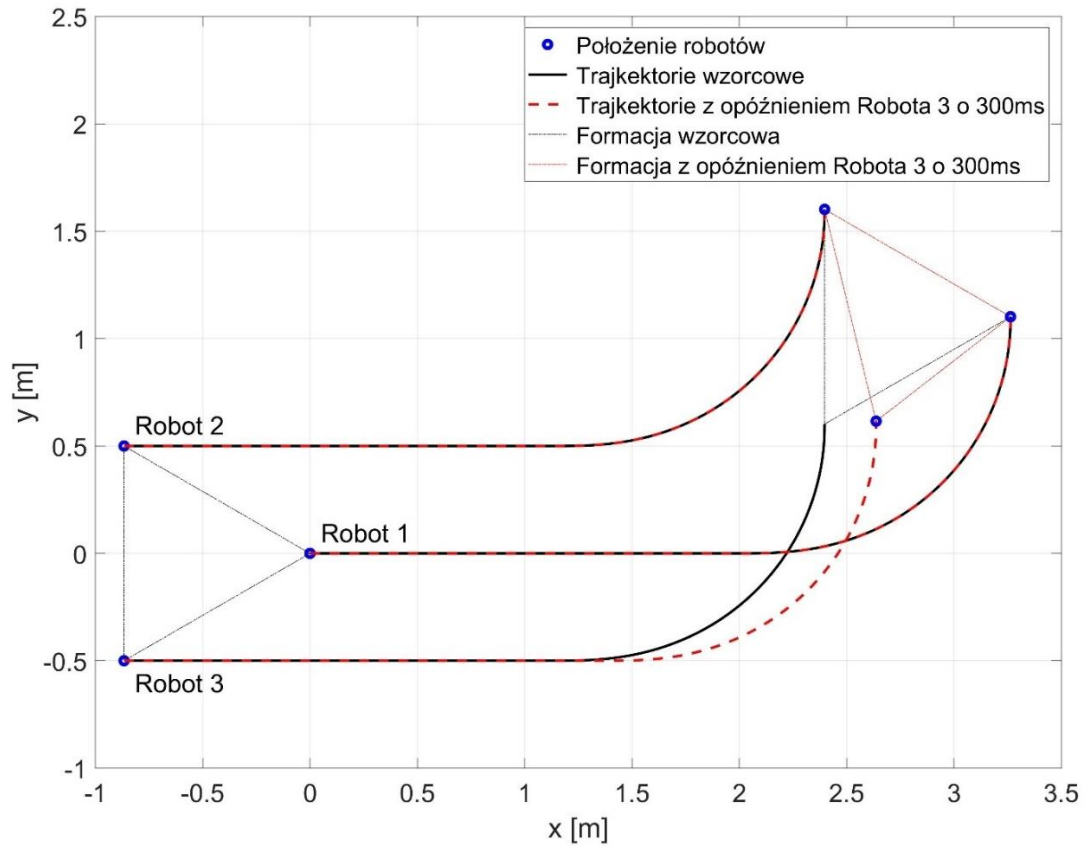
Rys. 3.12. Wizualizacja trajektorii wyznaczonej na podstawie scenariusza a) badań dla opóźnienia 3000 ms



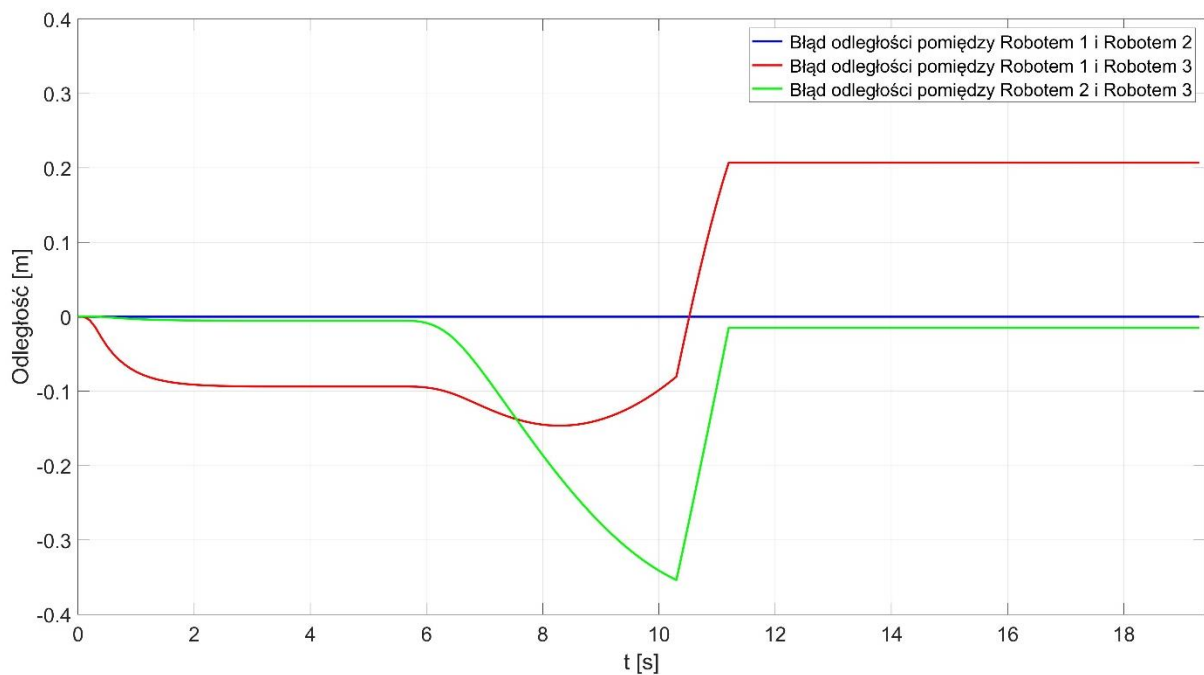
Rys. 3.13. Błąd odległości pomiędzy robotami wyznaczony na podstawie scenariusza a) badań dla opóźnienia 3000 ms

Podczas badań symulacyjnych zgodnie ze scenariuszem a) (Rys. 3.10 - Rys. 3.13) zaobserwowano brak zachowania wzajemnych odległości między robotami na początku ruchu. Błąd położenia końcowego Robota 3, którego sygnały sterujące zostały opóźnione, wynosił $e_{xy}(t_k) = 0,12 \text{ m}$ dla opóźnienia 300 ms i $e_{xy}(t_k) = 1,2 \text{ m}$ dla opóźnienia 3000 ms. Maksymalny błąd odległości pomiędzy robotami w czasie ruchu dla zadanej prędkości $v_d(t) = 0,4 \frac{\text{m}}{\text{s}}$ wynosił $e_{ij} = 0,225 \text{ m}$ dla opóźnienia 300 ms i $e_{ij} = 1,1 \text{ m}$ dla opóźnienia 3000 ms. Istotne błędy stwierdzono również w przypadku konfiguracji grupy po zakończeniu ruchu. Porządek grupy nie jest zachowany, nie odzwierciedla zadanej konfiguracji początkowej trójkąta równobocznego.

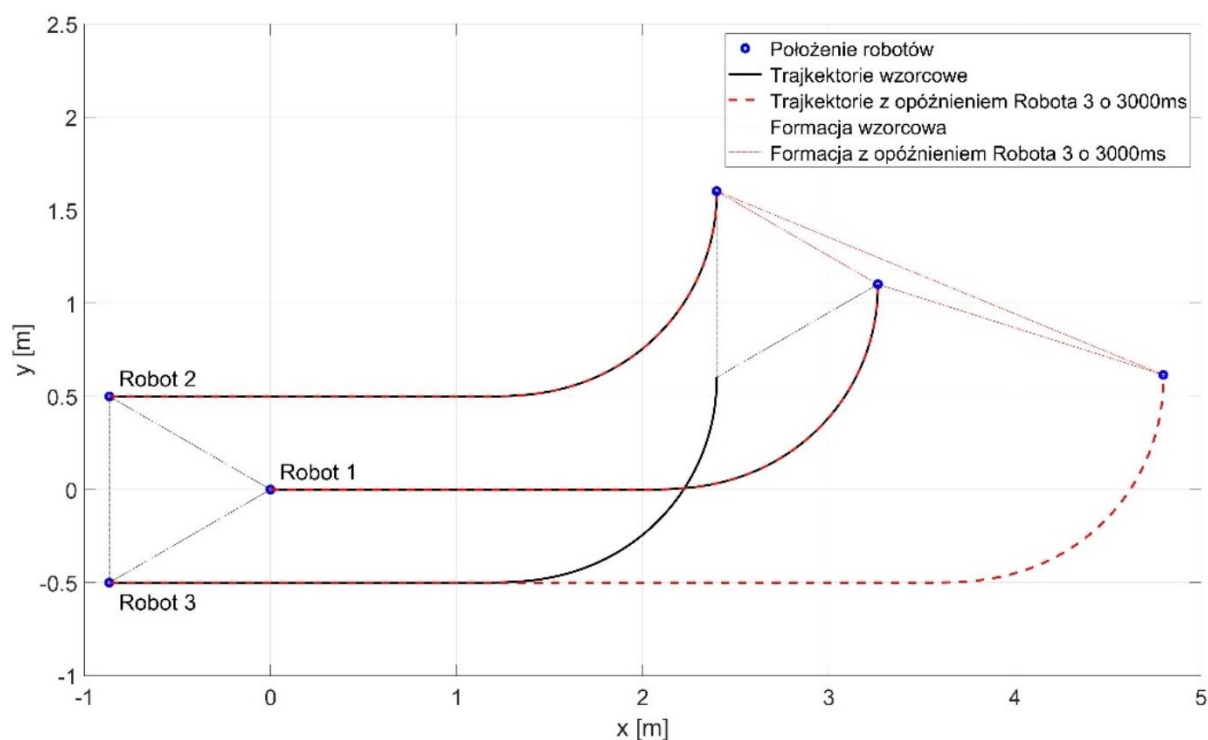
Scenariusz b) - przejazd odległości 2m ruchem prostoliniowym następnie wymuszenie niespodziewanej zmiany orientacji o kąt $\frac{\pi}{2}$ podczas ruchu



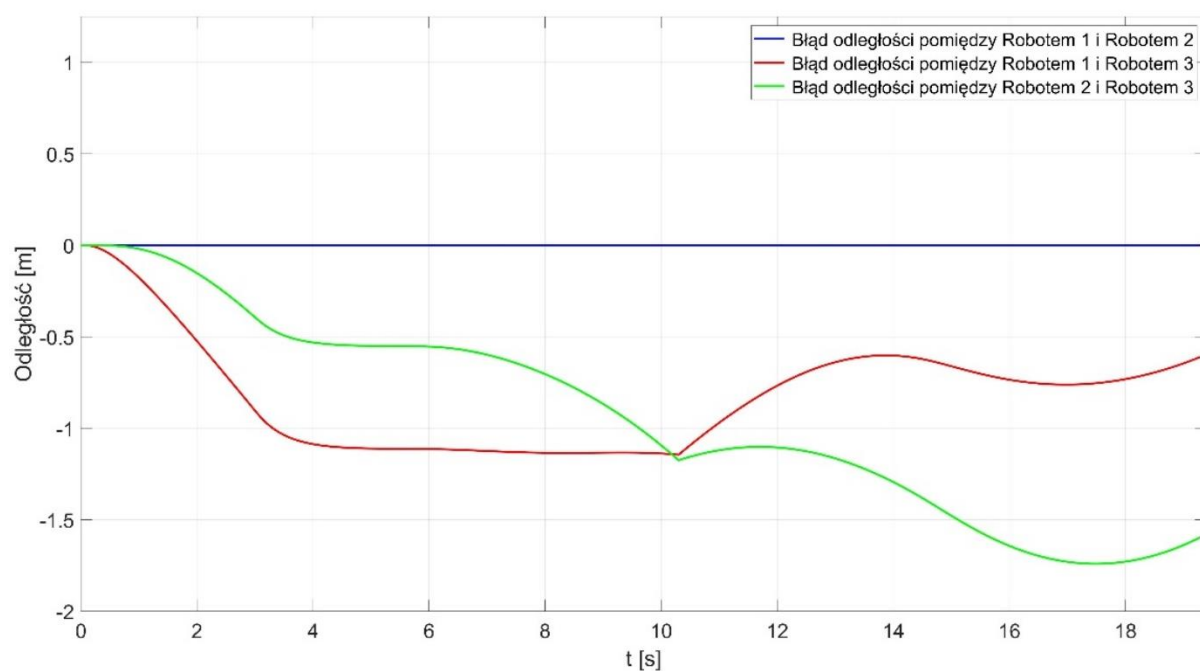
Rys. 3.14. Wizualizacja trajektorii wyznaczonej na podstawie scenariusza b) badań dla opóźnienia 300 ms



Rys. 3.15. Błąd odległości pomiędzy robotami wyznaczony na podstawie scenariusza b) badań dla opóźnienia 300 ms



Rys. 3.16. Wizualizacja trajektorii wyznaczonej na podstawie scenariusza b) badań dla opóźnienia 3000 ms



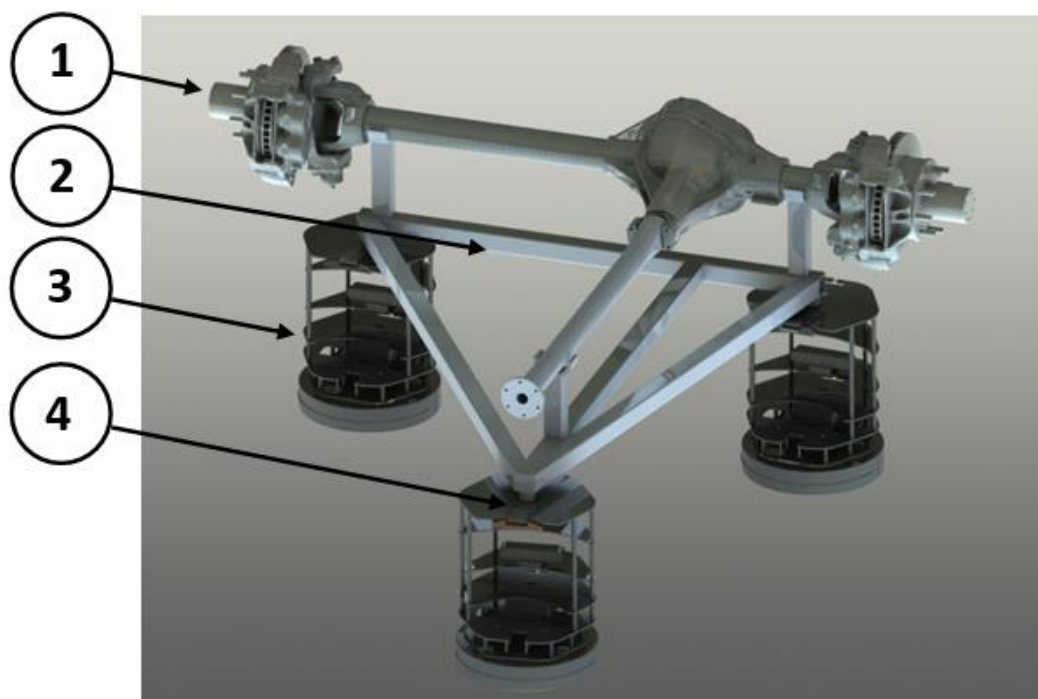
Rys. 3.17. Błąd odległości pomiędzy robotami wyznaczony na podstawie scenariusza b) badań dla opóźnienia 3000 ms

W przypadku badania symulacyjnego przeprowadzonego według scenariusza b) (Rys. 3.14 - Rys. 3.17) występuje powielenie błędów omówionych dla badania według scenariusza a). Błędy pojawiały się w wyniku dodatkowych opóźnień sygnałów sterujących w trakcie ruchu

grupy. Uzyskano końcowe błędy położenia Robota 3 o wartościach $e_{xy}(t_k) = 0.25$ m dla opóźnienia 300 ms i $e_{xy}(t_k) = 2.4$ m dla opóźnienia 3000 ms. Maksymalny błąd odległości pomiędzy robotami w czasie ruchu dla zadanej prędkości $v_d(t) = 0.4 \frac{m}{s}$ wynosił $e_{ij} = 0.35$ m dla opóźnienia 300 ms i $e_{ij} = 1.7$ m dla opóźnienia 3000 ms. Odległości pomiędzy poszczególnymi robotami osiągnęły wartość znacznie większą niż w przypadku badania pierwszego. Wystąpiła również znaczna deformacja szyku grupy.

3.3. Badania symulacyjne wzajemnego oddziaływania robotów w grupie rozproszonej transportującej ładunek

W rozdziale przedstawiono badania wzajemnego oddziaływania pomiędzy robotami w grupie rozproszonej transportującej ładunek, które częściowo opublikowano w pracy [103]. Na potrzeby badań symulacyjnych wzajemnego oddziaływania robotów w grupie rozproszonej rozważono ruch grupy trzech robotów Turtlebot 2 transportujących ładunek umieszczony na sztywnej paletie transportowej. Roboty połączone były z paletą za pomocą obrotowych przegubów o jednym stopniu swobody. Rozpatrzono sytuacje, w których pomiędzy robotami występowały opóźnienia w sygnałach sterujących. Widok systemu transportowego przedstawiono na Rys. 3.18.

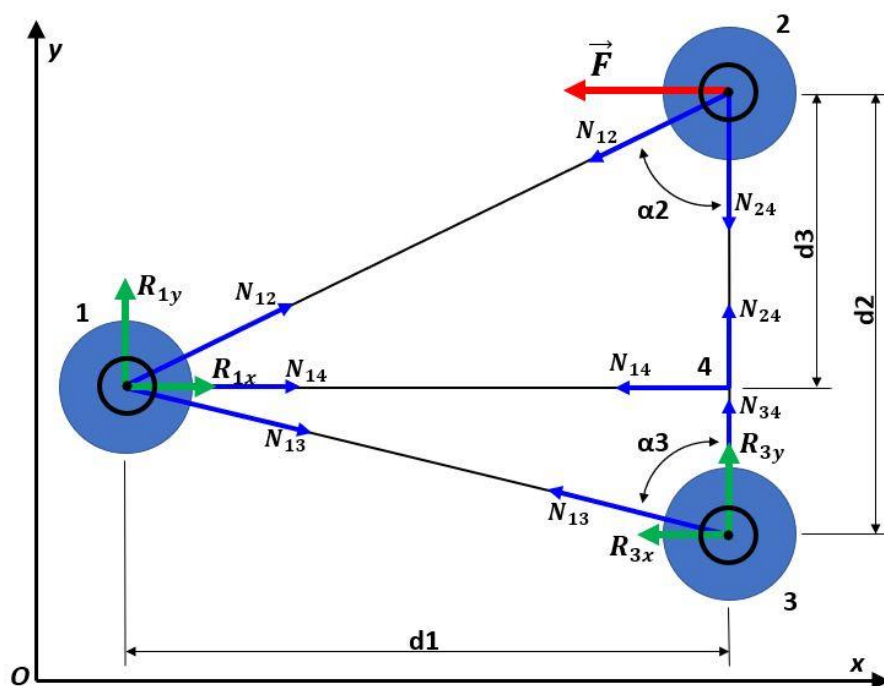


Rys. 3.18. Widok systemu transportowego: 1 - element transportowany, 2 - paleta, 3 - robot mobilny, 4 - przegub obrotowy

3.3.1. Zakres i metodyka badań wzajemnego oddziaływania robotów w grupie rozproszonej transportującej ładunek

Badania symulacyjne wzajemnego oddziaływania robotów w grupie rozproszonej podzielono na dwie części: analizę statyczną i analizę dynamiczną. Analizę statyczną przeprowadzono w celu sprawdzenia jak siła napędowa pochodząca od robota, który rozpoczął ruch, oddziałuje na roboty nieruchome (opóźnione). Zjawisko to w praktyce może skutkować boksowaniem kół, spalaniem silnika lub zerwaniem złącza. Analizę dynamiczną przeprowadzono w celu pomiaru odchyłek położenia robotów wywołanych siłą oddziałującą na nieruchome roboty.

Uproszczony schemat systemu transportowego przedstawiono na Rys. 3.19, gdzie 1, 2, 3 – roboty mobilne, d1, d2, d3 – wymiary palety transportowej, α_1, α_2 – kąty nachylenia belek zewnętrznych palety, Oxy – globalny układ współrzędnych, F – siła z jaką robot 2 pcha paletę, $R_{1x}, R_{1y}, R_{3x}, R_{3y}$ – reakcje w przegubach 1 i 3 zgodne z osiami x oraz y globalnego układu współrzędnych Oxy, N_{ij} gdzie $i, j = 1, \dots, 4$ – siły wewnętrzne w elementach konstrukcyjnych palety. W przeprowadzonych badaniach przyjęto, że obciążenia są przyłożone tylko do przegubów, a przeguby nie poruszają się względem siebie. W związku z tym, układ można było uznać za statyczny. Roboty 1 i 3 traktowane były jako podpory stałe, a przeguby, za pomocą których paleta była połączona z robotami, mogły obracać się tylko w osiach prostopadłych do płaszczyzny podłoża xy.



Rys. 3.19. Uproszczony schemat systemu transportowego

W badaniach założono, że dwa roboty będą opóźnione (ze względu na błędy transmisji danych) w stosunku do trzeciego, co powoduje oddziaływanie (poprzez paletę) robota nr 2 siłą napędową na opóźnione roboty nr 1 i 3. Siła narasta do wartości maksymalnej, wynikającej z maksymalnego momentu napędowego robota. Efektem tego zjawiska będą reakcje w przegubach robota 1 i 3 oraz siły wewnętrzne w elementach konstrukcyjnych palety transportowej. Przedstawiony na rysunku Rys. 3.19 schemat palety transportowej można traktować jako kratownicę. Zakładając, że obciążenia przyłożone są wyłącznie do węzłów kratownicy, układ można uznać za statyczny. Roboty 1 i 3 traktowane są jako nieruchome podpory. Przeguby, za pomocą których paleta jest połączona z robotami, mają możliwość obrotu wyłącznie w osiach prostopadłych do płaszczyzny podłoża xy .

3.3.2. Analiza statyczna obciążeń wynikających z fizycznego połączenia robotów

Do wyznaczenia reakcji w podporach i sił wewnętrznych w elementach konstrukcyjnych ramy, zastosowano metodę równowagi sił i momentów w węzłach kratownicy. W tym celu sformułowano układ równań, w postaci przedstawionej poniżej:

$$\begin{aligned}\sum M_1 &= -Fd_3 - R_{3y}d_1 + R_{3x}(d_2 - d_3) = 0 \\ \sum M_2 &= R_{3x}d_2 - R_{1x}d_3 + R_{1y}d_1 = 0 \\ \sum M_3 &= -Fd_2 + R_{1x}(d_2 - d_3) + R_{1y}d_1 = 0 \\ \sum P_x &= -F + R_{1x} - R_{3x} = 0\end{aligned}\tag{3.3}$$

gdzie M_1, M_2, M_3 są sumarycznymi momentami obrotowymi w węzłach kratownicy, a P_x i P_y sumarycznymi siłami wzdłuż osi globalnego układu współrzędnych Oxy .

Dodatkowo sformułowano równanie pomocnicze:

$$\sum P_y = R_{1y} + R_{3y} = 0\tag{3.4}$$

za pomocą którego sprawdzono brak błędów w układzie równań (4.3).

Układ równań (4.3) przekształcono do postaci macierzowej (4.5), a następnie rozwiązano w oprogramowaniu MATLAB.

$$\begin{bmatrix} Fd_3 \\ 0 \\ Fd_2 \\ F \end{bmatrix} = \begin{bmatrix} 0 & 0 & d_2 - d_3 & -d_1 \\ -d_1 & d_3 & d_2 & 0 \\ d_2 - d_3 & d_1 & 0 & 0 \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} R_{1x} \\ R_{1y} \\ R_{3x} \\ R_{3y} \end{bmatrix}\tag{3.5}$$

Do wyznaczenia reakcji podpór przyjęto wartość siły F i wymiary kratownicy zamieszczone w Tabeli 11.

Tabela 11. Wartości parametrów systemu transportowego

Parametr	d_1 [m]	d_2 [m]	d_3 [m]	α_1 [°]	α_2 [°]
Wartość	1	1.2	0.7	60	75

Wyznaczenie wartości reakcji podpór pozwoliło na wyznaczenie sił wewnętrznych w elementach konstrukcyjnych ramy. W tym celu sformułowano układy równań równowagi:

Węzeł 1

$$\begin{aligned}\sum P_{1x} &= R_{1x} + N_{13} \sin(90^\circ - \alpha_3) + N_{12} \cos(90^\circ - \alpha_2) + N_{14} = 0 \\ \sum P_{1y} &= R_{1y} + N_{12} \sin \alpha_2 - N_{13} \sin \alpha_3 = 0\end{aligned}\quad (3.6)$$

Węzeł 2

$$\begin{aligned}\sum P_{2x} &= -F - N_{12} \cos(90^\circ - \alpha_2) = 0 \\ \sum P_{2y} &= -N_{24} - N_{12} \sin \alpha_2 = 0\end{aligned}\quad (3.7)$$

Węzeł 3

$$\begin{aligned}\sum P_{3x} &= -R_{3x} - N_{13} \cos(90^\circ - \alpha_3) = 0 \\ \sum P_{3y} &= R_{3y} + N_{34} - N_{13} \sin \alpha_3 = 0\end{aligned}\quad (3.8)$$

3.3.3. Wyniki analizy statycznej obciążeń wynikających z fizycznego połączenia robotów

Do wyznaczenia reakcji w podporach i sił wewnętrznych w elementach konstrukcyjnych ramy przyjęto wartość siły $F = 250$ N, która odpowiada wartości siły generowanej przez napęd robota z w pełni naładowanym akumulatorem. Wartości wyznaczonych reakcji w podporach zostały przedstawione w tabeli 12.

Tabela 12. Wartości wyznaczonych reakcji w podporach

Parametr	R_{1x} [N]	R_{1y} [N]	R_{3x} [N]	R_{3y} [N]
Wartość	197.5	127.5	75	-75

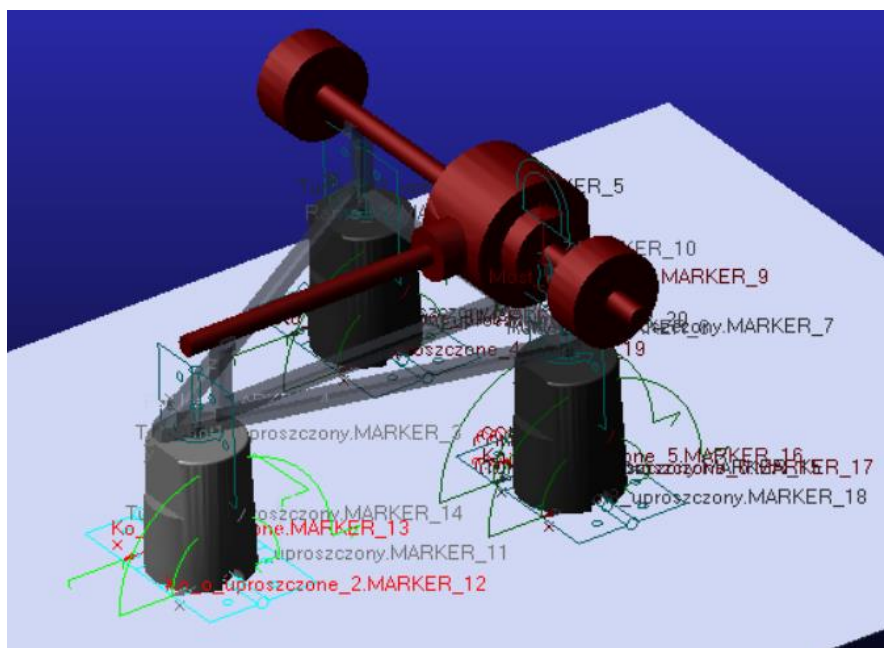
Znając wartości sił reakcji w podporach wyznaczono siły wewnętrzne w elementach konstrukcyjnych ramy. Wyznaczone wartości sił wewnętrznych przedstawiono w tabeli 13.

Tabela 13. Wartości sił wewnętrznych w elementach konstrukcyjnych ramy

Parametr	N_{12} [N]	N_{13} [N]	N_{14} [N]	N_{24} [N]	N_{34} [N]
Wartość	-287.35	288.46	-22,5	250	-351.92

3.3.4. Analiza dynamiczna obciążeń wynikających z opóźnień sygnałów sterujących

Analiza dynamiczna została przeprowadzona z wykorzystaniem oprogramowania MSC Adams. W tym celu w oprogramowaniu zaimplementowano uproszczony model systemu transportowego (Rys. 3.20), umożliwiający pomiar odchyłek położenia spowodowanych siłą $F = 250 \text{ N}$ działającą na opóźnione roboty, przez czas równy opóźnieniu sygnałów sterujących. Na wstępie sformułowano warunki początkowe symulacji. Określono początkowe położenie i początkową orientację robotów w globalnym układzie współrzędnych. Przyjęto uproszczony model robota złożony z korpusu i dwóch kół napędowych. Na kołach zadeklarowano przeguby obrotowe sterowane momentem napędowym. W miejscach połączeń palety transportowej zadeklarowano przeguby obrotowe o swobodnym obrocie. Transportowany ładunek został sztywno połączony z paletą transportową. Wszystkim elementom modelu symulacyjnego nadano masę zgodną z ich masą rzeczywistą oraz momenty bezwładności wyznaczone w programie CAD względem środka ciężkości. Analizowano jedynie ruch na płaszczyźnie związanej z podłożem, dlatego na model nałożono ograniczenia umożliwiające jedynie ruch na płaszczyźnie.



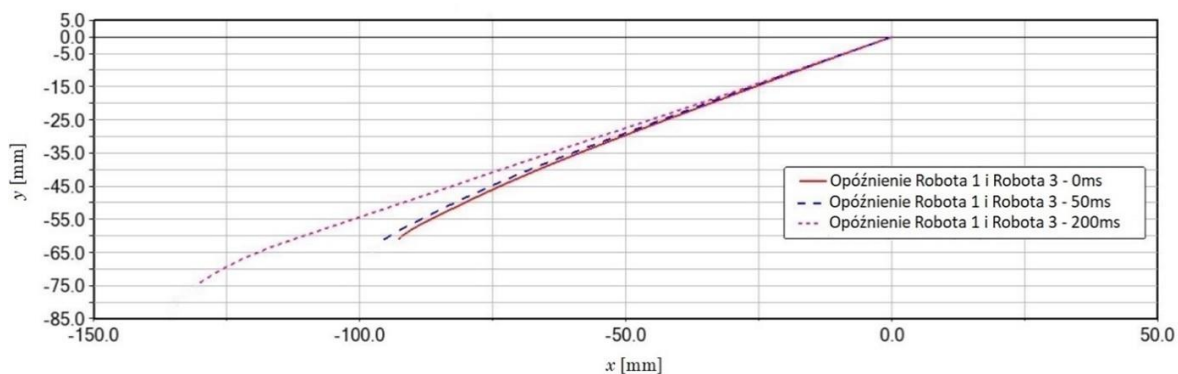
Rys. 3.20. Uproszczony model symulacyjny systemu transportowego w oprogramowaniu MSC Adams

3.3.5. Wyniki analizy dynamicznej obciążeń wynikających z opóźnień sygnałów sterujących

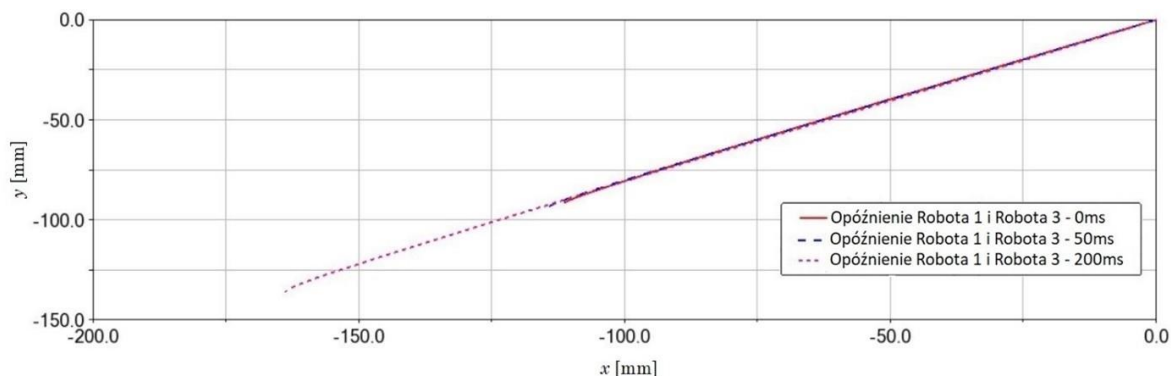
W czasie symulacji ruchu systemu transportowego wyznaczono przemieszczenia wzdłuż osi Ox i Oy punktów mocowania ramy do robotów. Prędkość robota została ustalona

na $0.5 \frac{m}{s}$. Opóźnienia czasowe sygnałów sterujących wynosiły od 0 ms do 200 ms. Na Rys. 3.21, Rys. 3.22 oraz Rys. 3.23 przedstawiono zmianę błędu położenia podczas ruchu grupowego dla poszczególnych robotów (1, 2 i 3) w wyniku zastosowania trzech wartości opóźnień Roboty 1 i Roboty 3 równych 0, 50 i 200 ms.

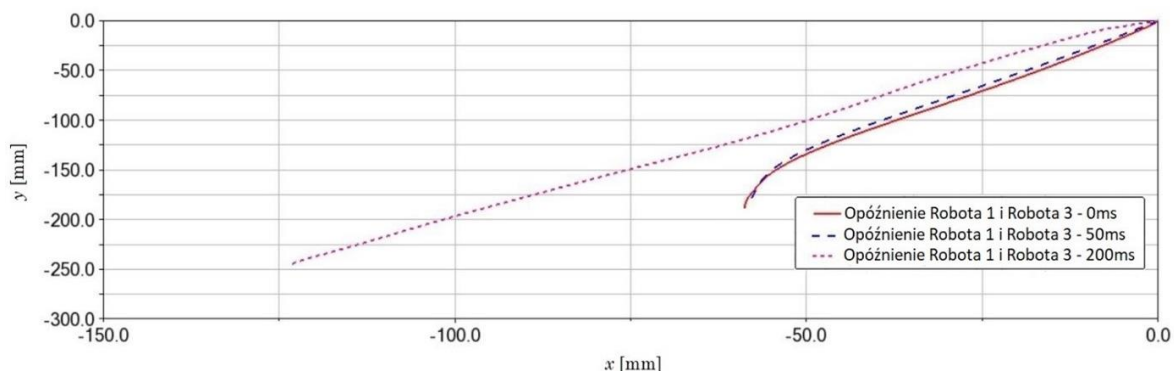
Analizując przedstawione na Rys. 3.21, Rys. 3.22, Rys. 3.23 trajektorie można zauważyć dwie składowe błędów: rozkład obciążenia oraz opóźnienia czasowe. Sposobem na eliminację zauważonych błędów może być sformułowanie prawa sterowania uwzględniającego opóźnienia powodowane zmienną w czasie siłą działającą na roboty. W związku z tym, w funkcji celu, należy uwzględnić różnicę sił występujących pomiędzy robotami, którą należałoby minimalizować.



Rys. 3.21. Trajektoria Roboty 1 przy opóźnieniu Roboty 1 i Roboty 3 od 0 ms do 200 ms



Rys. 3.22. Trajektoria Roboty 2 przy opóźnieniu Roboty 1 i Roboty 3 od 0 ms do 200 ms



Rys. 3.23. Trajektoria Roboty 3 przy opóźnieniu Roboty 1 i Roboty 3 od 0 ms do 200 ms

3.4. Podsumowanie badań rozpoznawczych

Przeprowadzona analiza wyników badań rozpoznawczych dokładności pozycjonowania robotów w grupie rozproszonej podczas ruchu wykazała, że opóźnienia czasowe mają bardzo znaczący wpływ na czas i dokładność osiągnięcia zadanej pozycji końcowej oraz na wzajemne odległości robotów w trakcie wykonywania zadania. Wielkość błędów uniemożliwia prawidłowe wykonywanie zadań przez grupę robotów.

W wyniku przeprowadzonych badań symulacyjnych wpływu opóźnień sygnałów sterujących na ruch grupy robotów o strukturze rozproszonej zaobserwowano brak zachowania zadeklarowanych odległości pomiędzy robotami na początku ruchu. Różnica pomiędzy odległością początkową a końcową wynosiła od ok. 0.2 m do ok. 1.2 m. Istotne błędy stwierdzono również w przypadku konfiguracji grupy po zakończeniu ruchu. Ustalony szyk grupy nie został zachowany.

W wyniku przeprowadzonych badań symulacyjnych wzajemnego oddziaływania robotów w grupie rozproszonej transportującej ładunek zauważono, że istotnym problemem w tego typu aplikacjach jest dokładność osiągnięcia zadanej pozycji przez system transportowy wynikająca z opóźnień czasowych sygnałów sterujących systemem. Otrzymane wyniki analizy pozwoliły stwierdzić, że pojawienie się opóźnień czasowych sygnałów sterujących wywoływało niesynchroniczny start robotów tworzących grupę rozproszoną. Podczas nierównoczesnego startu pojawiały się reakcje na siłę, z jaką robot ruchomy poprzez przewożony ładunek działa na pozostałe, jeszcze nieruchome roboty, co w efekcie przenosi się na żywotność i dokładność ich układów napędowych. Pojawiające się siły skutkowały przemieszczeniem się i zmianą orientacji systemu transportowego, co wywołuje błędy pozycji i orientacji w stosunku do konfiguracji początkowej.

W wyniku przeprowadzonych badań laboratoryjnych wpływu ilości robotów w grupie o strukturze rozproszonej na opóźnienia sygnałów sterujących wykazano, że multiplikowanie robotów w grupie powoduje wzrost opóźnień transmisji z każdym dodanym robotem. Pojawienie się przeszkody na drodze transmisji sygnałów również jest dodatkowym czynnikiem wpływającym na wartość opóźnień. Dodatkowo zaobserwowano, że na opóźnienia transmisji mają wpływ zastosowane narzędzia do transmisji bezprzewodowej. W przypadku pomiarów obejmujących dwa roboty zastosowano takie same laptopy Lenovo Yoga 300, umieszczone na robotach. Trzeci robot wyposażony był w laptop Asus o innych parametrach karty sieciowej niż Lenovo. Na wszystkich komputerach jednak zainstalowany był ten sam system operacyjny i taka sama wersja ROS. Badanie z uwzględnieniem trzech robotów

pokazało znaczący wpływ urządzeń sieciowych, w które wyposażone były komputery na wartość opóźnień transmisji sygnałów sterujących [101].

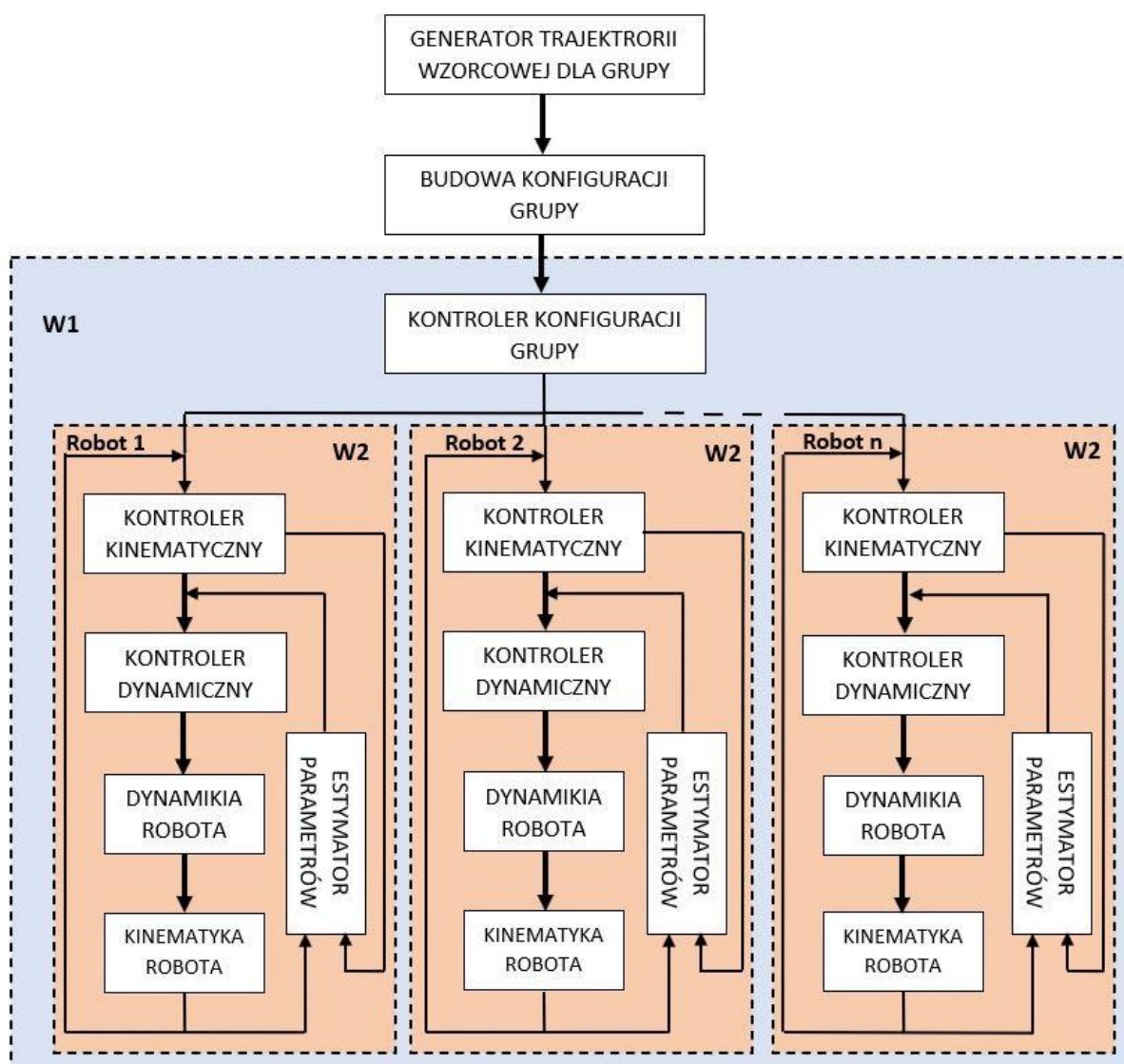
W wyniku przeprowadzonych badań laboratoryjnych wpływu rozmieszczenia przeszkód w środowisku pracy grupy robotów o strukturze rozproszonej na opóźnienia sygnałów sterujących wykazano, że przeszkody środowiskowe w postaci dużych odległości transmisji danych oraz w postaci szklanych i murowanych ścian występujących na drodze transmisji danych pomiędzy elementami składowymi grupy robotów nie wpływają na wartość opóźnienia czasowego sygnałów sterujących grupą robotów.

Uzyskane wyniki badań dają podstawową wiedzę na temat charakteru, źródła pochodzenia i wartości opóźnień czasowych sygnałów sterujących. Na podstawie przeprowadzonych badań można stwierdzić, że w przypadku sterowania grupą robotów o strukturze rozproszonej, konieczne jest wprowadzenie synchronizacji czasowej robotów w algorytmie sterującym. Nieprawidłowa lub nieefektywna synchronizacja czasu skutkowałą błędnym działaniem algorytmu sterowania grupą rozproszoną. W praktyce oznacza to, że działania takie jak rozpoczęcie ruchu, wykonanie manewru, zatrzymanie się będą realizowane niesynchronicznie przez poszczególne roboty, czego efektem będzie nieprawidłowa realizacja zadania (ruch po złej trajektorii, nieutrzymanie formacji, zderzenie z przeszkodą).

Rozdział 4

OPRACOWANIE AUTORSKIEGO ALGORYTMU SYNCHRONICZNEGO STEROWANIA GRUPĄ ROBOTÓW O STRUKTURZE ROZPROSZONEJ

Dla rozpatrywanej w pracy grupy robotów mobilnych o strukturze rozproszonej zaproponowano algorytm sterowania dwuwarstwowego, który przedstawiono na Rys. 4.1.



Rys. 4.1. Struktura dwuwarstwowego systemu sterowania rozproszoną grupą robotów mobilnych

Warstwa W1 – sterowania nadrzędnego grupą odpowiada za utrzymanie wymaganej formacji grupy. Warstwa W2 - sterowania podrzędnego odpowiada za sterowanie poszczególnymi grupami realizując zadanie śledzenia trajektorii. Algorytm składa się z kilku modułów programowych realizujących określone funkcje, które opisano w kolejnych podrozdziałach.

4.1. Generator trajektorii wzorcowej dla grupy

W bloku GENERATOR TRAJEKTORII WZORCOWEJ DLA GRUPY (Rys. 4.1) definiowana jest wzorcowa trajektoria $(x_d(t), y_d(t))$ i orientacja $\theta_d(t)$ wirtualnej struktury (ruchomego układu współrzędnych względem, którego budowana jest formacja) .względem globalnego układu współrzędnych. Następnie, przy założeniu, że robot porusza się po zadanej trajektorii bez zakłóceń i bez błędów początkowych wyznaczane są wzorcowe sygnały sterujące $u_1(t)$ i $u_2(t)$, które są odpowiednio wzorcową prędkością liniową i kątową w zadaniu śledzenia trajektorii.

Kąt orientacji wzorcowej $\theta_d(t)$ w punkcie $(x_d(t), y_d(t))$ wyznaczany jest na podstawie zależności:

$$\theta_d(t) = \arctan2(\dot{x}_d(t), \dot{y}_d(t)) + k\pi \quad (4.1)$$

gdzie $k = 0, 1$ odpowiada za kierunek obrotu (dla $k = 0$ obrót w lewo, dla $k = 1$ obrót w prawo).

Prędkość liniowa grupy wyznaczana jest z zależności:

$$u_1(t) = v_d(t) = \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (4.2)$$

gdzie znak zależy od kierunku ruchu (+ deklaruje ruch do przodu, - deklaruje ruch do tyłu).

Różniczkując równanie (4.1) w dziedzinie czasu otrzymano zależność na wyznaczenie referencyjnej prędkości kątowej grupy w postaci:

$$u_2(t) = \omega_d(t) = \frac{\dot{x}_d(t)\ddot{y}_d(t) - \dot{y}_d(t)\ddot{x}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (4.3)$$

Łącząc powyższe równania otrzymano wektor współrzędnych konfiguracyjnych wzorcowych $q_d(t)$:

$$q_d(t) = [x_d(t), y_d(t), \theta_d(t)]^T \quad (4.4)$$

i wektor sterowań $u_d(t)$ w postaci:

$$u_d(t) = [v_d(t), \omega_d(t)]^T \quad (4.5)$$

UWAGA

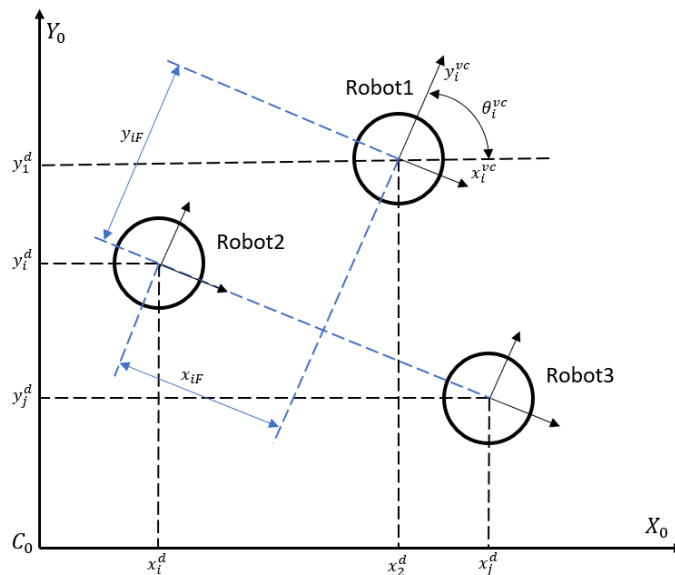
Aby otrzymać wektor współrzędnych konfiguracyjnych (4.4) i wektor sterowań (4.5) na podstawie równań (4.1) - (4.3) należy spełnić warunki nałożone na trajektorię i prędkość:

- trajektoria wzorcowa musi być opisana funkcją umożliwiającą obliczenie jej drugiej pochodnej;
- prędkość liniowa robota musi być różna od zera $v_r(t) \neq 0$.

W przypadku, kiedy prędkość liniowa $v_r(t)$ jest równa 0 robot obraca się z prędkością $\omega_r(t)$ w punkcie, który stanowi jego środek obrotu. Kąt orientacji $\theta_r(t)$ i prędkość kątowna $\omega_r(t)$ nie mogą być wtedy wyznaczone z zależności (4.1) i (4.3) i muszą być sztywno podane, co stanowi warunki początkowe zadania śledzenia trajektorii.

4.2. Budowa konfiguracji grupy

Do modelowania formacji wykorzystano założenia algorytmu Wirtualnej Struktury [50], [63], [78], [61], który szerzej opisano w rozdziale 1.1.2. Schemat wzajemnych relacji robotów w grupie rozproszonej przedstawiono na Rys. 4.2. Punkt środkowy wirtualnej struktury umieszczony jest w środku geometrycznym Robotu 1, którego współrzędne i orientację określa wektor $q_1(t) = [x_1^d(t), y_1^d(t), \theta_1^d(t)]$. Kąt $\theta_1^d(t)$ jest taki sam jak kąt $\theta_i^{vc}(t)$, który określa orientację grupy w globalnym układzie współrzędnych $C_0X_0Y_0$. Położenie pozostałych robotów w grupie określane jest poprzez sztywne odległości x_{iF} oraz y_{iF} od środka układu współrzędnych wirtualnej struktury.



Rys. 4.2. Schemat wzajemnych relacji robotów w grupie rozproszonej z uwzględnieniem założeń algorytmu Wirtualnej Struktury

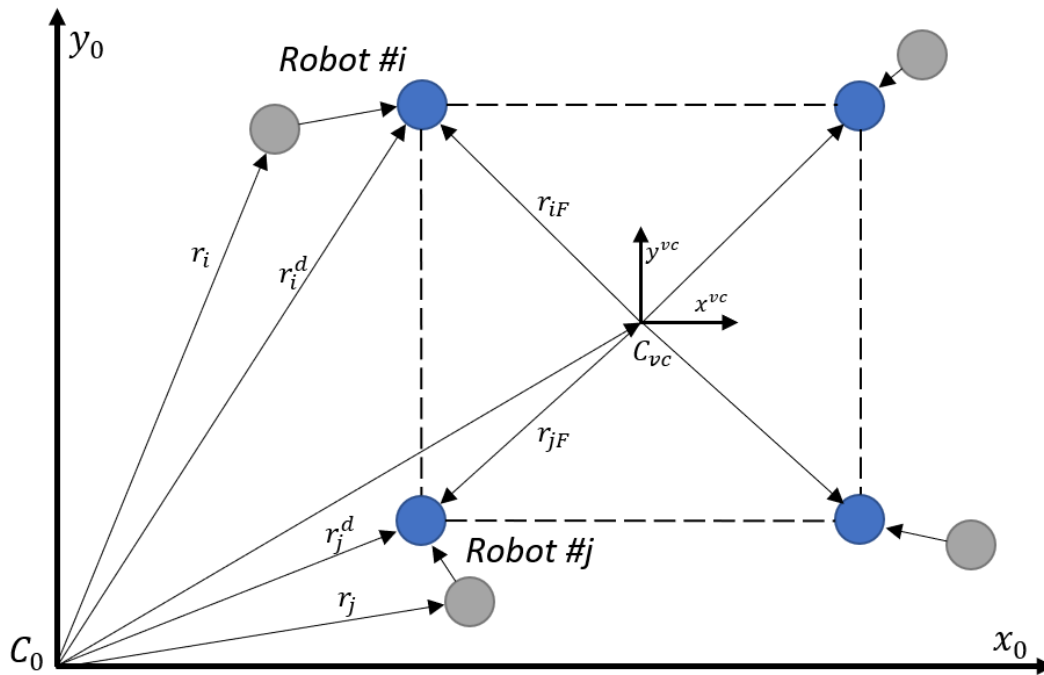
Współrzędne robotów dla dowolnej konfiguracji i ilości robotów względem układu współrzędnych wirtualnej struktury wyznaczane są z równań:

$$\begin{bmatrix} x_i^d(t) \\ y_i^d(t) \end{bmatrix} = \begin{bmatrix} x_i^{vc}(t) \\ y_i^{vc}(t) \end{bmatrix} + \begin{bmatrix} \cos(\theta_i^{vc}(t)) & -\sin(\theta_i^{vc}(t)) \\ \sin(\theta_i^{vc}(t)) & \cos(\theta_i^{vc}(t)) \end{bmatrix} \begin{bmatrix} x_{iF} \\ y_{iF} \end{bmatrix} \quad (4.6)$$

4.3. Kontroler konfiguracji grupy

Do realizacji postawionego w niniejszej pracy zadania wykorzystano i rozwinięto - opracowania [91], [6], [55], [17]. Aby zastosować algorytmy konsensusu w celu utrzymania formacji grupy zakłada się, że dla każdego robota w grupie znana jest wzorcowa trajektoria grupy - położenie i orientacja wirtualnego układu współrzędnych. Jeżeli istnieje ryzyko braku wymiany informacji pomiędzy robotami z powodu dynamicznie zmieniającego się środowiska, ograniczonej wymiany informacji lub zaniku komunikacji, pożądana formacja grupy może nie być utrzymana. W takim przypadku w algorytmie sterowania grupą należy wprowadzić estymator stanu grupy dla każdego robota [91].

Na Rys. 4.3 przedstawiono relacje geometryczne robotów w grupie rozproszonej dla opracowania algorytmu konsensusu.



Rys. 4.3. Relacje geometryczne grupy robotów dla algorytmu konsensusu [91]

Przyjmując rzeczywiste położenie w układzie odniesienia C_0 i-tego robota wchodzącego w skład grupy jako $r_i(t) = [x_i(t), y_i(t)]^T$, położenie środka geometrycznego wirtualnej struktury jako $C_{vc}(t) = [x^{vc}(t), y^{vc}(t)]$ i jego orientację $\theta^{vc}(t)$ względem układu odniesienia,

oraz dewiację i-tego robota w grupie względem środka wirtualnej struktury jako $r_{iF}(t) = [x_{iF}(t), y_{iF}(t)]$ można wyznaczyć położenie referencyjne i-tego robota wchodzącego w skład grupy jako $r_i^d(t) = [x_i^d(t), y_i^d(t)]^T$ z zależności:

$$\begin{bmatrix} x_i^d(t) \\ y_i^d(t) \end{bmatrix} = \begin{bmatrix} x^{vc}(t) \\ y^{vc}(t) \end{bmatrix} + \begin{bmatrix} \cos(\theta^{vc}(t)) & -\sin(\theta^{vc}(t)) \\ \sin(\theta^{vc}(t)) & \cos(\theta^{vc}(t)) \end{bmatrix} \times \begin{bmatrix} x_{iF} \\ y_{iF} \end{bmatrix} \quad (4.7)$$

Jeżeli każdy robot wchodzący w skład grupy podąża za swoim położeniem wzorcowym bezbłędnie wtedy formacja grupy jest zachowana. W rzeczywistości pojawiające się błędy w przesyłaniu informacji pomiędzy robotami o ich aktualnym stanie, doprowadzają do zaburzeń utrzymania formacji. W związku z tym dynamikę systemu wieloagentowego z uwzględnieniem więzów zależnych od czasu można zapisać na podstawie [17] jako:

$$\xi_i(t) = h(t, \xi_i) + \sum_{j=1}^m g_{ij} f(t, \xi_j(t + \Delta t_{ij}), \xi_i(t + \Delta t_{ij})), \quad i = 1, 2, \dots, m \quad (4.8)$$

gdzie $\xi_i(t) = [x_i(t), y_i(t), \theta_i(t)]$ - stan i-tego robota w chwili t , a g_{ij} - waga określająca połączenia komunikacyjne pomiędzy robotami i oraz j , Δt_{ij} - opóźnienie czasowe w komunikacji pomiędzy robotami i oraz j .

Określając położenie robota w chwili t jako $r_i(t) = [x_i(t), y_i(t)]$ oraz sygnał sterujący i- tym robotem w grupie jako $u_i(t)$ podstawowe prawo sterowania dla algorytmu konsensusu przyjmuje postać:

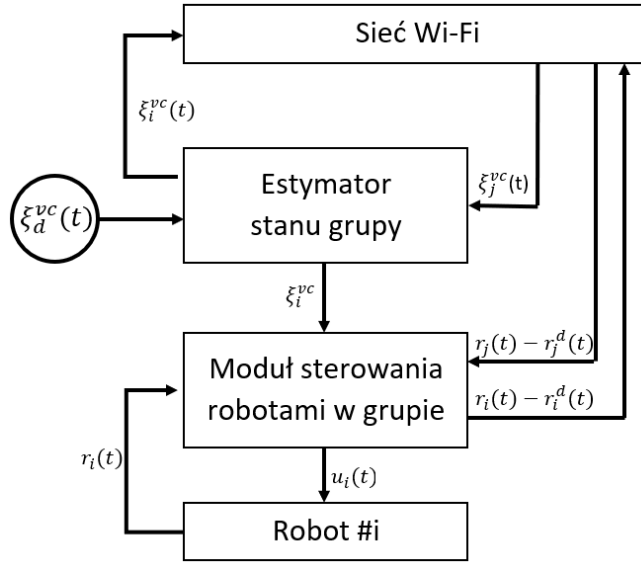
$$u_i(t) = - \sum_{j=1}^n g_{ij} k_{ij} f(r_i(t) - r_j(t)), \quad i = 1, 2, \dots, n \quad (4.9)$$

gdzie k_{ij} jest wzmocnieniem kontrolera dla pary robotów ij , $g_{ij} = 1$ jeżeli następuje transmisja danych z robota j do robota i , $g_{ij} = 0$ jeżeli transmisja nie następuje. Konsensus pomiędzy robotami i oraz j zostanie osiągnięty, jeżeli:

$$\lim_{t \rightarrow \infty} \|\xi_i(t) - \xi_j(t)\| = 0 \quad (4.10)$$

Jeżeli wszystkie pary robotów w grupie osiągną konsensus (spełnią równanie 4.10) wtedy cały system dynamiczny osiągnie konsensus grupowy – wzorcowa formacja zostanie zachowana [91], [55], [17].

Kontroler grupy zdecentralizowanej opracowano jako strukturę hierarchiczną zawierającą dwie warstwy: estymator stanu grupy oraz moduł sterowania robotami w grupie jak przedstawiono na Rys. 4.4.



Rys. 4.4. Schemat blokowy struktury kontrolera grupowego [91]

Oznaczając stan grupy (pozycję i orientację układu współrzędnych wirtualnej struktury) jako $\xi^{vc}(t) = [x^{vc}(t), y^{vc}(t), \theta^{vc}(t)]^T$, stan wzorcowy grupy jako $\xi_d^{vc}(t) = [x_d^{vc}(t), y_d^{vc}(t), \theta_d^{vc}(t)]^T$ oraz stan rzeczywisty i-tego robota jako $\xi_i^{vc}(t) = [x_i^{vc}(t), y_i^{vc}(t), \theta_i^{vc}(t)]^T$ sformułowano dla estymatora stanu grupy prawo sterowania, które uwzględnia opóźnienia czasowe w komunikacji pomiędzy robotami w postaci zależności (5.11), na podstawie którego estymowany jest stan układu współrzędnych wirtualnej struktury dla każdego robota:

$$\xi_i^{vc}(t) = \frac{\xi_d^{vc}(t) - \gamma(\xi_i^{vc}(t) - \xi_d^{vc}(t)) + \sum_{j=1}^n g_{ij}^{vc} [\xi_j^{vc}(t) - \gamma(\xi_i^{vc}(t + \Delta t_{ij}) - \xi_j^{vc}(t + \Delta t_{ij}))]}{1 + \sum_{j=1}^n g_{ij}^{vc}} \quad (4.11)$$

gdzie $g_{ij}^{vc} = 1$ jeżeli robot i otrzymuje informacje z estymatora stanu grupy o stanie robota j, $g_{ij}^{vc} = 0$ jeżeli robot i nie otrzymuje informacji o stanie robota j, oraz $\gamma > 0$ jest współczynnikiem wzmocnienia. Dodatkowo założono, że każdy robot otrzymuje informacje o stanie wirtualnego układu współrzędnych i wartości opóźnienia czasowego w komunikacji pomiędzy robotami, więc estymator stanu grupy powinien zapewnić $\xi_i^{vc}(t) \rightarrow \xi_d^{vc}(t)$.

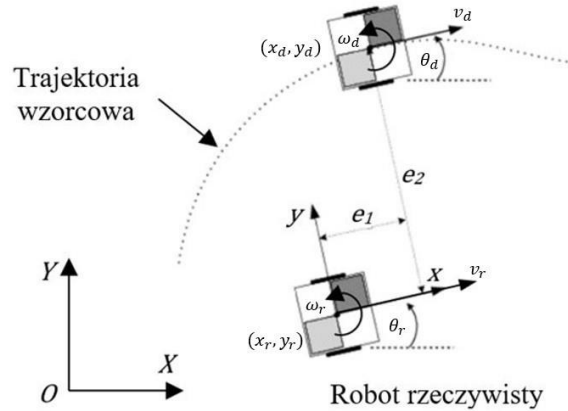
Jako warstwa sterowania robotami w grupie (bazując na [91]) zastosowano algorytm poszerzonego algorytmu konsensusu (4.12), w którym uwzględniono opóźnienia czasowe w komunikacji pomiędzy robotami. Równanie (4.12) pozwala na wyznaczenie sygnałów sterujących $u_i(t)$ tak aby wzorcowa formacja geometryczna została utrzymana podczas ruchu:

$$u_i(t) = \dot{r}_i^d(t) - \alpha_i (r_i(t) - r_i^d(t)) - \sum_{j=1}^n g_{ij} k_{ij} [(r_i(t + \Delta t_{ij}) - r_i^d(t)) - (r_j(t + \Delta t_{ij}) - r_j^d(t))] \quad (4.12)$$

gdzie $\alpha_i > 0, k_{ij} > 0$ są współczynnikami wzmocnienia oraz $g_{ij} = 1$ jeżeli robot i otrzymuje informacje z estymatora stanu grupy o stanie robota j , $g_{ij} = 0$ jeżeli robot i nie otrzymuje informacji o stanie robota j .

4.4. Kontroler kinematyczny

Realizacja zadania śledzenia trajektorii wzorcowej przez robota zawsze związana jest z pojawianiem się błędów - różnicy pomiędzy położeniem wzorcowym a rzeczywistym. W pracy zaproponowano kontroler proporcjonalny z pętlą sprzężenia zwrotnego, opracowany na podstawie [53], [117], [46], [125], [7], [1], którego zadaniem jest uzyskanie sygnałów sterujących sprowadzających robota na zadaną trajektorię. Zasadę działania kontrolera przedstawiono graficznie na rysunku Rys. 4.5.



Rys. 4.5. Zasada działania kontrolera kinematycznego [53]

Przedstawione na rysunku błędy e_1 i e_2 związane są ze śledzeniem trajektorii względem osi odpowiednio X oraz Y układu OXY związanego z robotem rzeczywistym. W celu określenia wszystkich błędów śledzenia trajektorii należy uwzględnić jeszcze błąd e_3 , który związany jest z orientacją robota. Uwzględniając transformację błędów w układzie współrzędnych robota (jak przedstawiono w pracach [117], [46]) otrzymano następujące równanie błędów śledzenia trajektorii w czasie:

$$\begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) & 0 \\ -\sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r(t) - x_d(t) \\ y_r(t) - y_d(t) \\ \theta_r(t) - \theta_d(t) \end{bmatrix} \quad (4.13)$$

Analizując Rys. 4.5 można zauważyć, że minimalizacja błędu e_1 wzdłuż osi X jest związana z prędkością liniową. Minimalizacja błędu e_2 wzdłuż osi Y związana jest z prędkością kątową. Minimalizacja błędu orientacji e_3 zależna jest od prędkości kątowej robota.

Kontroler kinematyczny powinien w taki sposób dobierać sygnały sterujące, aby operując prędkością kątową ω kierować robota na trajektorię wzorcową z zachowaniem profilu prędkości liniowej v [53], [46], [7]. Na tej podstawie, uwzględniając model kinematyczny robota (2.9) i równanie błędów (4.13), otrzymano model błędów śledzenia trajektorii w postaci:

$$\begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \end{bmatrix} = \begin{bmatrix} \cos e_3(t) & 0 \\ \sin e_3(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_d(t) \\ \omega_d(t) \end{bmatrix} + \begin{bmatrix} -1 & e_2(t) \\ 0 & -e_1(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad (4.14)$$

gdzie

$$u_1(t) = v_d(t)\cos e_3(t) - v_r(t), \quad u_2(t) = \omega_d(t) - \omega_r(t) \quad (4.15)$$

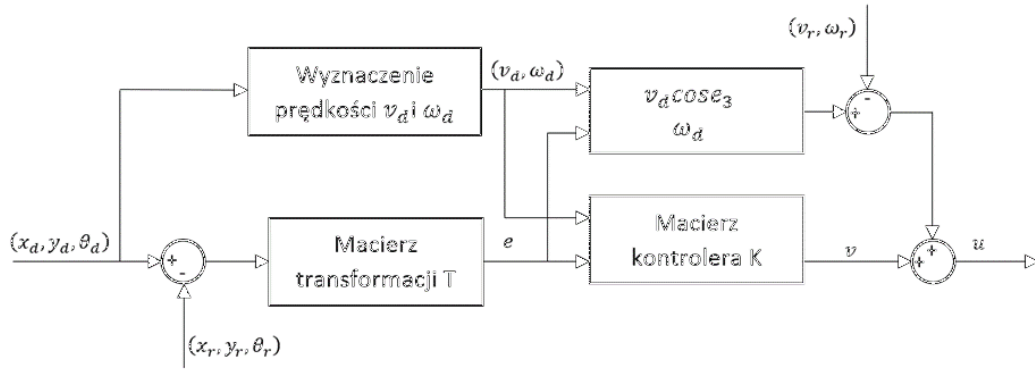
W celu pozbycia się nieliniowości modelu równanie (4.14) przekształcono do postaci [53],

$$\begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & u_2(t) & 0 \\ -u_2(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_3(t) \\ 0 \end{bmatrix} v_d(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r(t) \\ \omega_r(t) \end{bmatrix} \quad (4.16)$$

A następnie zlinearyzowano wokół punktu pracy, dla którego $e_1 = e_2 = e_3 = 0$ oraz $v_r = \omega_r = 0$. Otrzymano liniowy model błędów śledzenia trajektorii, który zapisano w przestrzeni stanu:

$$\Delta \dot{e}(t) = \begin{bmatrix} 0 & \omega_r(t) & 0 \\ -\omega_r(t) & 0 & v_r(t) \\ 0 & 0 & 0 \end{bmatrix} \Delta e(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \Delta v(t) \quad (4.17)$$

Po wyznaczeniu liniowego modelu błędów śledzenia trajektorii opracowano schemat blokowy kontrolera kinematycznego, który przedstawiono na rysunku poniżej [53].



Rys. 4.6. Schemat blokowy kontrolera kinematycznego

Przedstawiony system ma trzy zmienne stanu oraz dwa wejścia sterujące. Zaproponowane na podstawie [53], [7], [1] prawo sterowania pozwala wyznaczyć dwa sygnały sterujące - prędkość liniową i kątową na podstawie transformacji błędów położenia i orientacji robota oraz prędkości wzorcowych. Równanie kontrolera kinematycznego przyjmuje postać:

$$\begin{bmatrix} u_{kv}(t) \\ u_{k\omega}(t) \end{bmatrix} = \begin{bmatrix} v_d(t) \cos \theta_d(t) + k_1 e_1(t) \\ \omega_d(t) + v_d(t) (k_2 e_2(t) + k_3 \sin e_3(t)) \end{bmatrix} \quad (4.18)$$

Wyznaczenie wzmocnień kontrolera przeprowadzono porównując rzeczywiste i pożądane pierwiastki wielomianu charakterystycznego jak w pracy [53]. Rozwiązaniem układu równań porównującego pierwiastki charakterystyczne wielomianu są wzmocnienia opisane zależnościami:

$$\begin{aligned} k_1 &= k_3 = 2\varepsilon\omega_n(t) \\ k_2 &= g * |v_d(t)| \end{aligned} \quad (4.19)$$

gdzie

$$\omega_n(t) = \sqrt{\omega_d^2(t) + g v_d^2(t)} \quad (4.20)$$

oraz

$$\varepsilon \in (0,1), \quad g > 0 \quad (4.21)$$

Stabilność kontrolera sprawdzono na podstawie [53], [7], [1] wykorzystując kryterium Lapunova wokół punktu $p_e = [e_1, e_2, e_3] = 0$ na podstawie następującej postaci funkcji Lapunova:

$$V(e) = \frac{1}{2}(e_1^2 + e_2^2) + (1 - \cos(e_3))/k_2 \quad (4.22)$$

Zakładając, że trajektoria wzorcowa jest funkcją ciągłą to wyprowadzone prawo sterowania (4.18) zapewnia, że $e_1 \rightarrow 0$ oraz $e_3 \rightarrow 0$ jeżeli $t \rightarrow \infty$. Nie jest to jednak

zagwarantowane dla e_2 w przypadku, kiedy robot jest ustawiony równolegle do trajektorii, a prędkość referencyjna $v_r = 0$. Jednak, jeżeli trajektoria jest zaprojektowana w taki sposób, że $v_r \neq 0$, kontroler (4.18) steruje robotem tak aby $e_1 \rightarrow 0$, $e_2 \rightarrow 0$ oraz $e_3 \rightarrow 0$ dla $t \rightarrow \infty$.

4.5. Kontroler dynamiczny

Realizacja ruchu z dużą prędkością (dla robotów AGV od 0.4 m/s do 2 m/s) wymagającego wysokiej precyzji pozycjonowania konieczne jest uwzględnienie w sterowaniu dynamiki robota (co opisano w rozdziale 2.2). Założeniem do opracowania kontrolera dynamicznego było potraktowanie sygnałów sterujących z kontrolera kinematycznego (4.18) (prędkości liniowej i kątowej) jako sygnały wejściowe. Do opracowania kontrolera wykorzystano parametryczny model dynamiczny robota (2.17) oraz rozważania przedstawione w pracy [75]. Wymnażając i dodając odpowiednie elementy macierzy modelu dynamicznego (2.17) oraz traktując sygnały sterujące kontrolera kinematycznego $u_{kv}(t)$ i $u_{k\omega}(t)$ jako sygnały wejściowe kontrolera dynamicznego, otrzymano nową postać modelu dynamicznego robota:

$$\begin{bmatrix} \dot{v}_x(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} -\frac{\sigma_4}{\sigma_1} v_x(t) + \frac{u_{kv}(t)}{\sigma_1} \\ -\frac{\sigma_6}{\sigma_2} \omega(t) + \frac{u_{k\omega}(t)}{\sigma_2} \end{bmatrix} \quad (4.23)$$

Dokonując przekształceń powyższego wyrażenia (załącznik B), można je zapisać w postaci parametrycznej jako:

$$\begin{bmatrix} u_{kv}(t) \\ u_{k\omega}(t) \end{bmatrix} = \begin{bmatrix} \dot{v}_x(t) & 0 & v_x(t) & 0 \\ 0 & \dot{\omega}(t) & 0 & \omega(t) \end{bmatrix} [\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4]^T \quad (4.24)$$

lub wyłączając sygnały sterujące w postaci następującego wyrażenia:

$$\begin{bmatrix} u_{kv}(t) \\ u_{k\omega}(t) \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \dot{v}_x(t) \\ \dot{\omega}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & v_x(t) & 0 \\ 0 & 0 & 0 & \omega(t) \end{bmatrix} [\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4]^T \quad (4.25)$$

Równanie dynamiki (4.25) można zapisać w formie skróconej jako:

$$u_d(t) = D(t)\dot{u}(t) + \eta(t) \quad (4.26)$$

gdzie

$$\begin{aligned} u_d(t) &= [u_{kv}(t) \quad u_{k\omega}(t)]^T, \quad u = [v_x(t) \quad \omega(t)]^T, \\ D(t) &= \text{diag}(\vartheta_1(t), \vartheta_2(t)) \\ \eta(t) &= \begin{bmatrix} 0 & 0 & v_x(t) & 0 \\ 0 & 0 & 0 & \omega(t) \end{bmatrix} [\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4]^T \end{aligned} \quad (4.27)$$

Wykorzystując zadanie dynamiki odwrotnej, na podstawie [75] sformułowano prawo sterowania opisane równaniem:

$$\begin{bmatrix} u_{dv}(t) \\ u_{d\omega}(t) \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \vartheta_1(t) \\ \vartheta_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & v_x(t) & 0 \\ 0 & 0 & 0 & \omega(t) \end{bmatrix} [\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4]^T \quad (4.28)$$

gdzie:

$$\begin{aligned} \vartheta_1(t) &= \dot{u}_{kv}(t) + k_v(u_{kv}(t) - v_x(t)), \quad k_v > 0 \\ \vartheta_2(t) &= \dot{u}_{k\omega}(t) + k_\omega(u_{k\omega}(t) - \omega(t)), \quad k_\omega > 0 \end{aligned} \quad (4.29)$$

W celu poprawy jakości śledzenia trajektorii z wykorzystaniem kontrolera dynamicznego (4.28) na podstawie punktu 2.3.3 zaproponowano adaptację parametrów $\sigma_1, \dots, \sigma_4$ w trakcie ruchu, zgodnie ze wzorem (2.22), gdzie:

$$\Delta(t) = \begin{bmatrix} u_{kv}(t) \\ u_{k\omega}(t) \end{bmatrix} - \begin{bmatrix} v_x(t) \\ \omega(t) \end{bmatrix} \quad (4.30)$$

Rozdział 5

BADANIA SYMULACYJNE ALGORYTMU SYNCHRONICZNEGO STEROWANIA GRUPĄ ROBOTÓW O STRUKTURZE ROZPROSZONEJ

W celu zbadania dokładności śledzenia trajektorii przez roboty z wykorzystaniem opracowanego algorytmu sterowania przeprowadzono szereg symulacji w postaci śledzenia wybranych rodzajów trajektorii. Badania podzielono na dwa etapy. W pierwszej kolejności przeprowadzono symulacje śledzenia trajektorii wzorcowych przez jednego robota, co realizowane jest przez warstwę wewnętrzną W2 algorytmu sterowania (Rys. 4.1), a następnie zbadano symulacyjnie warstwę zewnętrzną W1 algorytmu sterowania, która odpowiada

za synchronizację i utrzymanie zadanej formacji grupy robotów. Program badań warstwy W2 przedstawiono w rozdziale 5.1, a ich wyniki w rozdziale 5.2. Program badań warstwy W1 przedstawiono w rozdziale 5.3, a ich wyniki w rozdziale 5.4.

5.1. Zakres i metodyka badań teoretycznych warstwy wewnętrznej W2 algorytmu sterowania

Analizując ruch robotów w grupie zauważono, że błędy pozycjonowania każdego z robotów oddziałują na ruch pozostałych. Badania rozpoczęto od sprawdzenia warstwy wewnętrznej W2 systemu sterowania przedstawionego na Rys. 4.1. W tym celu przeprowadzono badania symulacyjne sprawdzające działanie zaimplementowanych kontrolerów dla trzech konfiguracji układu sterowania:

- 1) śledzenia trajektorii za pomocą kontrolera kinematycznego,
- 2) śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego (połączenia kontrolera kinematycznego i dynamicznego),
- 3) śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego z adaptacją.

W każdej z konfiguracji, badania przeprowadzono według dwóch kryteriów:

- 1) kryterium ruchu przez określony czas,
- 2) kryterium ruchu po skończonej drodze.

We wszystkich przypadkach zastosowano model matematyczny robota złożony z modelu kinematycznego (2.9) oraz modelu dynamicznego (2.17).

Badania wg. kryterium ruchu przez określony czas polegały na sprawdzeniu stabilności, czasu regulacji i uchybu w czasie symulacji równym 120s. Badania wg. kryterium ruchu po skończonej drodze polegały na sprawdzeniu stabilności, czasu regulacji i uchybu po zrealizowaniu drogi odpowiadającej 0.75 okresu funkcji opisujących trajektorię (równania 5.1 – 5.3). Najczęściej spotykane w literaturze trajektorie testowe to: trajektoria sinusoidalna („sinus”), kołowa („okrąg”) oraz trajektoria w kształci cyfry osiem („ósemka”). Trajektorie te charakteryzują się różnym stopniem skomplikowania dla zadania śledzenia trajektorii, wynikającym z częstotliwości zmian prędkości liniowej i kątowej w czasie ruchu. W przypadku ruchu po trajektorii kołowej prędkość liniowa i kątowa są stałe. W przypadku ruchu po trajektorii sinusoidalnej i w kształcie cyfry osiem prędkości liniowa i kątowa zmieniają się z różną częstotliwością. Wspomniane trajektorie wzorcowe opisano następującymi równaniami parametrycznymi:

- trajektoria kołowa

$$\begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} x_0 + R \sin(\omega t) \\ y_0 + R \cos(\omega t) \\ \text{atan2}(\dot{y}_d, \dot{x}_d) + k\pi \end{bmatrix} \quad (5.1)$$

- trajektoria sinusoidalna

$$\begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} x_0 + R \sin(\omega t) \\ y_0 + 2\omega t \\ \text{atan2}(\dot{y}_d, \dot{x}_d) + k\pi \end{bmatrix} \quad (5.2)$$

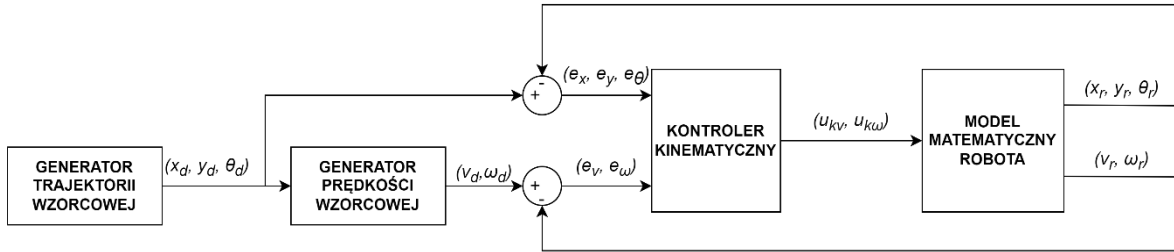
- trajektoria „ósemka”

$$\begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} x_0 + A_x \sin(\omega_x t) \\ y_0 + A_y \cos(\omega_y t) \\ \text{atan2}(\dot{y}_d, \dot{x}_d) + k\pi \end{bmatrix} \quad (5.3)$$

Do obliczeń trajektorii wzorcowych przyjęto następujące parametry: $x_0 = 0$, $y_0 = 0$, $R = 1$ m, $\omega = 0.1 \frac{\text{rad}}{\text{s}}$, $k = (0, 1)$ oraz prędkości liniowe i kątowe wyznaczone na podstawie wzorów (4.2) i (4.3).

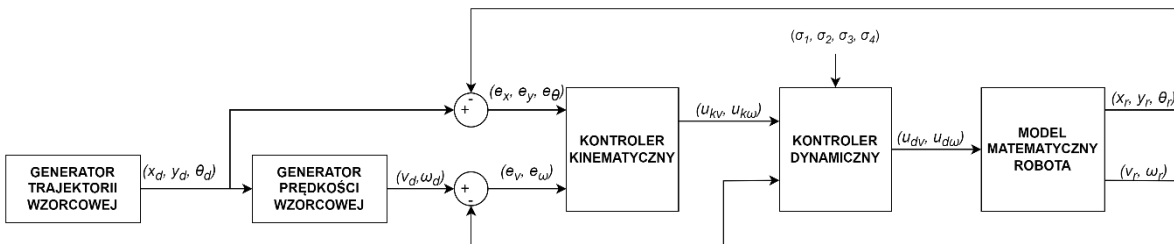
W badaniu śledzenia trajektorii za pomocą kontrolera kinematycznego przeprowadzono symulację śledzenia trajektorii wzorcowej zgodnie ze schematem na Rys. 5.1. Wartości wzmocnień k_1, k_2 kontrolera kinematycznego wyznaczono ze wzoru (4.19) dla $\varepsilon = 0.7$ oraz $g = 65$. Wartości parametrów ε i g wyznaczono symulacyjnie na podstawie odpowiedzi układu na zadane wymuszenie (w postaci prędkości liniowej i kątowej o wartościach

wyznaczonych z wzorów 4.2 i 4.3 dla wybranych trajektorii wzorcowych), postępując jak w pracach [10], [53], [117]. Dla wartości $\varepsilon = 0.7$ i $g = 65$ uzyskano najmniejszy błąd śledzenia i najkrótszy czas regulacji. Wartości te zastosowano w badaniach dla wszystkich konfiguracjach układu sterowania.



Rys. 5.1. Schemat symulacji do badania kontrolera kinematycznego

W badaniu śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego (połączenie kontrolera kinematycznego (4.18) i dynamicznego (4.28)) przeprowadzono symulację śledzenia trajektorii wzorcowej zgodnie ze schematem na Rys. 5.2. Zadanie śledzenia trajektorii wzorcowej realizowane jest przez kontroler kinematyczny (4.18) i dynamiczny (4.28). Wartości wzmocnień k_1, k_2 , kontrolera kinematycznego (4.18) wyznaczono ze wzoru (4.19) dla $\varepsilon = 0.7$ oraz $g = 65$. Wartości wzmocnień kontrolera dynamicznego (4.28) przyjęto jako $k_v = 1.2$, $k_\omega = 4$, $l_v = l_\omega = 0.5$. Wartości te wyznaczono symulacyjnie dla najmniejszych błędów śledzenia trajektorii i najkrótszego czasu regulacji przy zachowaniu stabilności układu [24], [75], [76], [129] i zastosowano dla wszystkich konfiguracji układu sterowania.

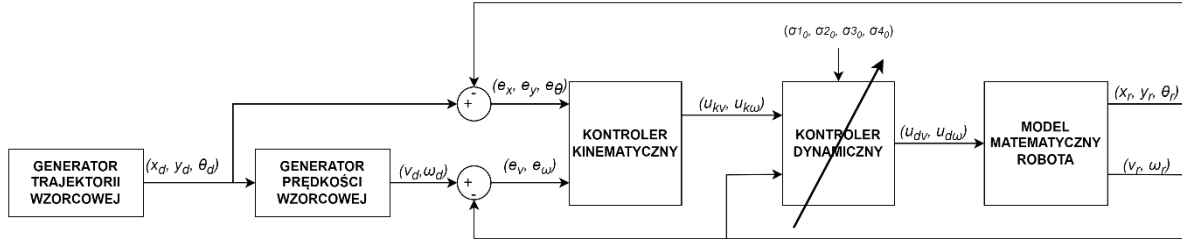


Rys. 5.2. Schemat symulacji do badania kontrolera kinematyczno-dynamicznego

W badaniu śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego z adaptacją przeprowadzono symulację śledzenia trajektorii wzorcowej zgodnie ze schematem przedstawionym na Rys. 5.3. Zadanie śledzenia trajektorii wzorcowej realizowane jest przez kontroler kinematyczny (4.18) i dynamiczny (4.28) z adaptacją (2.22). Wartości wzmocnień k_1, k_2 , kontrolera kinematycznego wyznaczono ze wzoru (4.19) dla $\varepsilon = 0.7$ oraz $g = 65$. Podobnie jak poprzednio przyjęto następujące wartości wzmocnień kontrolera dynamicznego

$k_v = 1.2$, $k_\omega = 4$, $l_v = 0.5$, $l_\omega = 0.5$. Z kolei wartości wzmocnień dla adaptacji parametrów modelu przyjęto jako:

$$\gamma = \begin{bmatrix} 4.5 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0.0005 & 0 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix} \quad (5.4)$$



Rys. 5.3. Schemat symulacji do badania kontrolera kinematyczno-dynamicznego z adaptacją

W badaniach założono pozycję początkową robota zgodną z początkowym punktem trajektorii:

- $x_{d0} = 0$, $y_{d0} = 0$ dla trajektorii sinus i „ósemka”,
- $x_{d0} = 0$, $y_{d0} = 1$ dla trajektorii „okrąg”.

Parametry ruchu robota ograniczono przyspieszeniami: liniowym $a_v = 0.4 \frac{m}{s^2}$ i kątowym $a_\omega = 0.8 \frac{rad}{s^2}$.

5.2. Badania symulacyjne warstwy wewnętrznej W2 algorytmu sterowania

Badania symulacyjne przeprowadzono w środowisku MATLAB/Simulink. Wyniki badań przedstawiono w formie wykresów:

- porównania trajektorii wzorcowej i otrzymanej w wyniku symulacji,
- uchybów pozycji e_x , e_y i orientacji e_θ ,
- uchybów prędkości liniowej e_v i kątowej e_ω
- oraz sygnałów sterujących, którymi są prędkości liniowa i kątowa.

Jakość zaproponowanych układów regulacji oceniono na podstawie wartości uchybów – różnicy pomiędzy położeniem, orientacją i prędkościami wzorcowymi, a wyznaczonymi w wyniku symulacji oraz czasu regulacji – czasu po jakim osiągnięto stały uchyb śledzenia trajektorii mieszczący się w przedziale $\pm 5\%$ największego osiągniętego uchybu e_x , e_y , e_θ .

W badaniach początkowa pozycja i orientacja robota była opisana wektorem współrzędnych konfiguracyjnych, w postaci:

- dla trajektorii „okrąg” - $q_0 = [0, 1, 0]$;

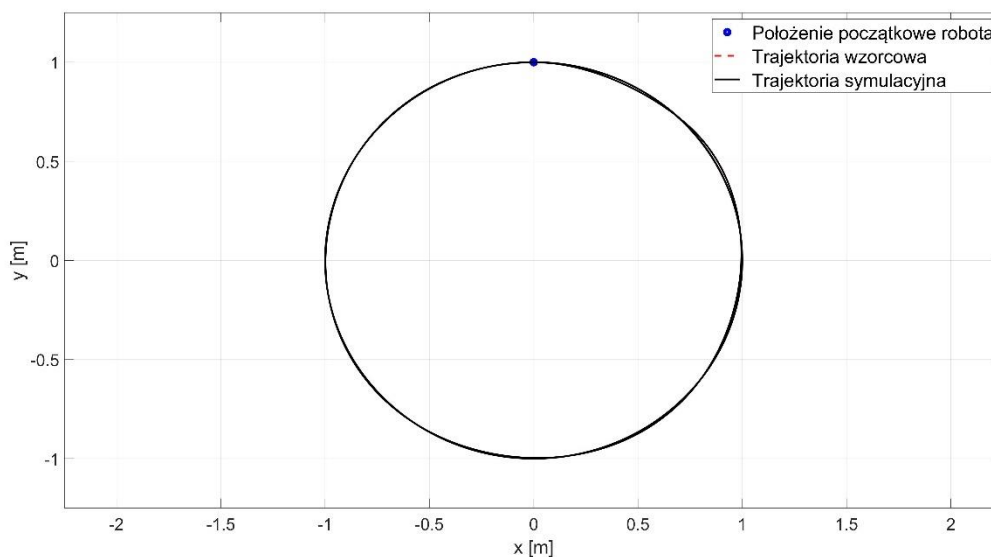
- dla trajektorii „sinus” - $q_0 = \left[0, 0, \frac{\pi}{6}\right]$;
- dla trajektorii „ósemka” - $q_0 = \left[0, 0, \frac{\pi}{4}\right]$.

5.2.1. Badania kontrolera kinematycznego

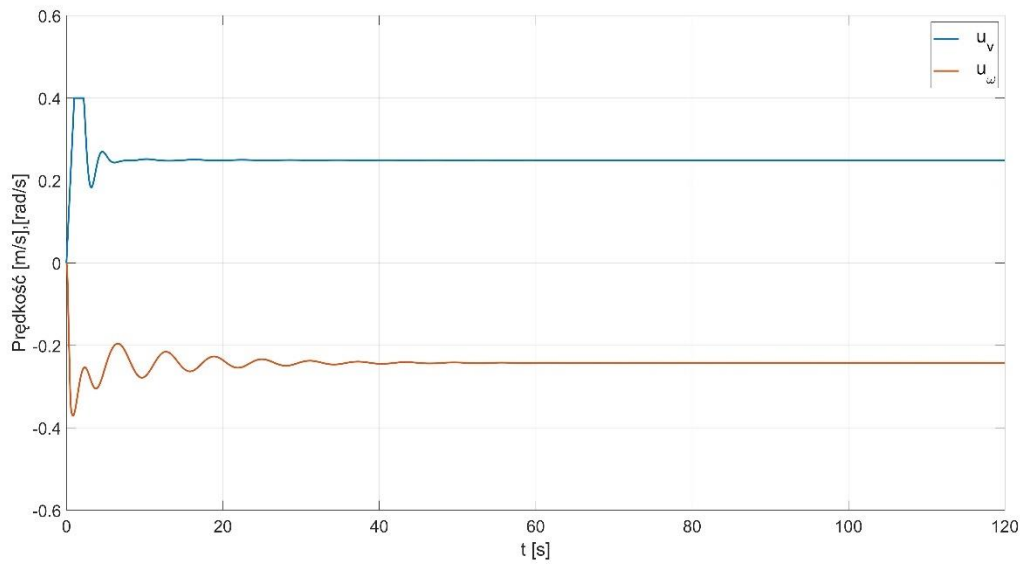
Wyniki badań symulacyjnych śledzenia trajektorii wzorcowej za pomocą kontrolera kinematycznego przedstawiono na Rys. 5.4 - Rys. 5.27. Badania przeprowadzono według schematu przedstawionego na Rys. 5.1.

5.2.1.1. Kryterium ruchu przez określony czas dla kontrolera kinematycznego

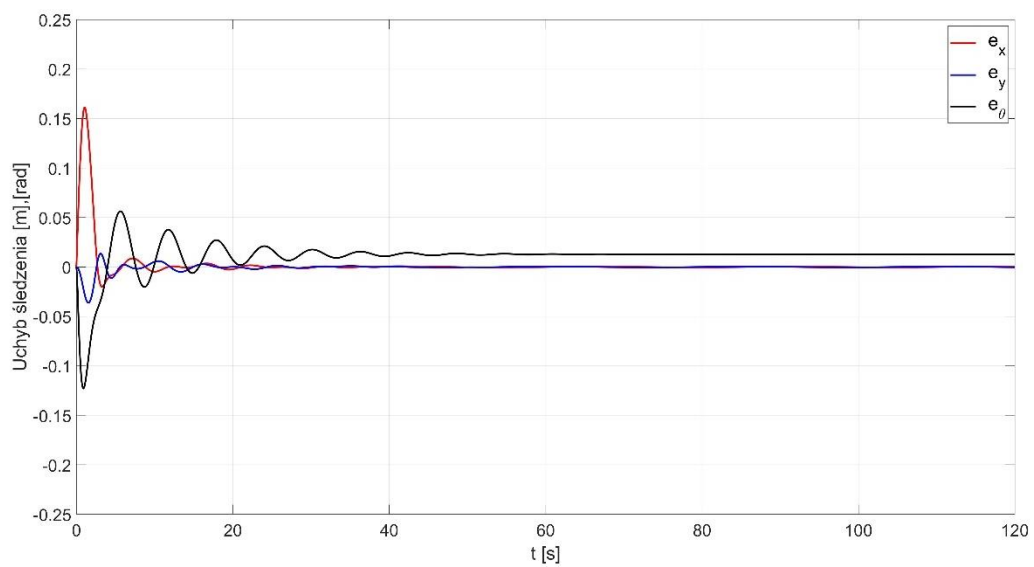
- Śledzenie trajektorii „okrąg”



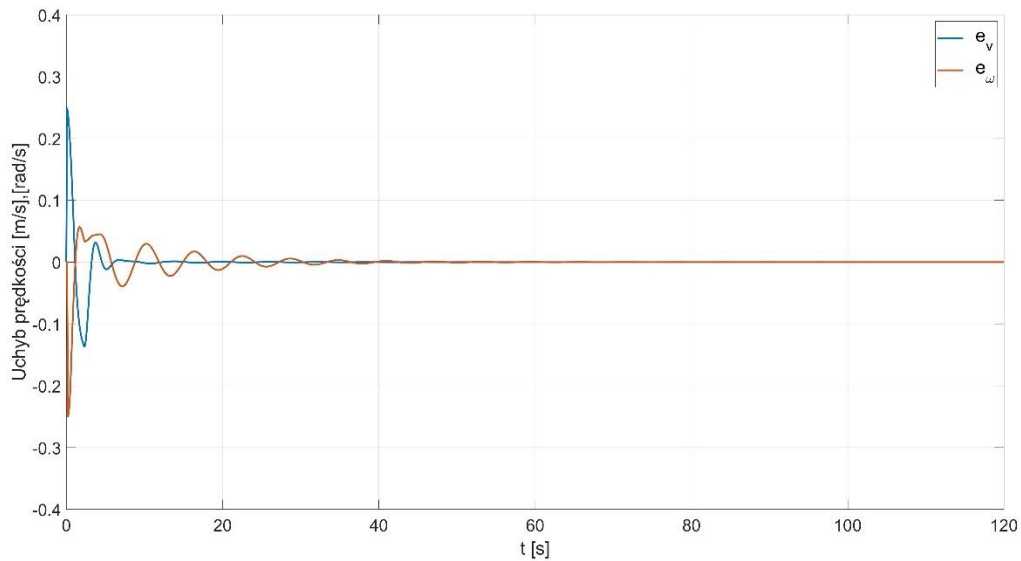
Rys. 5.4. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego stabilności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 5.5. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 5.6. Uchyby śledzenia e_x , e_y , e_{θ} w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego



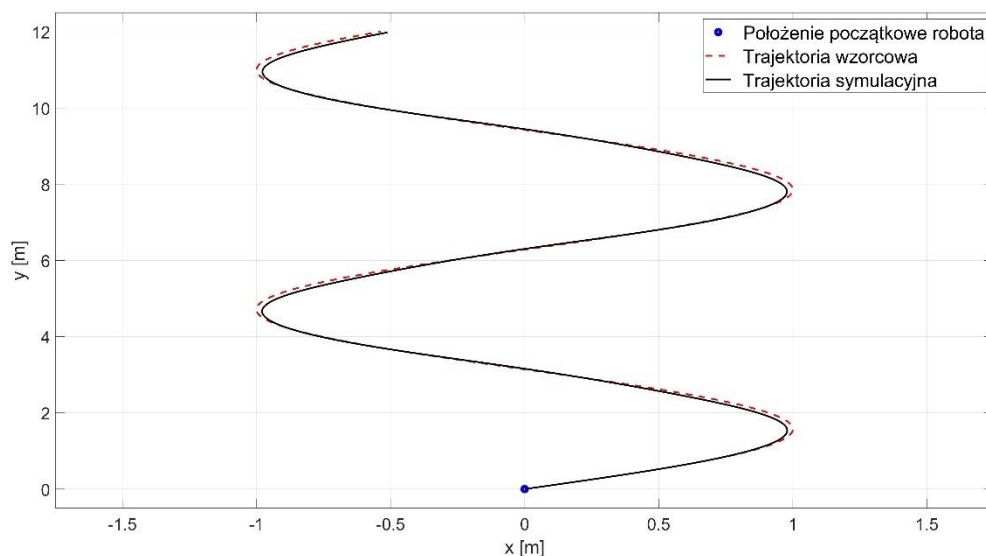
Rys. 5.7. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 5.6) oraz prędkości (Rys. 5.7) nie większymi niż:

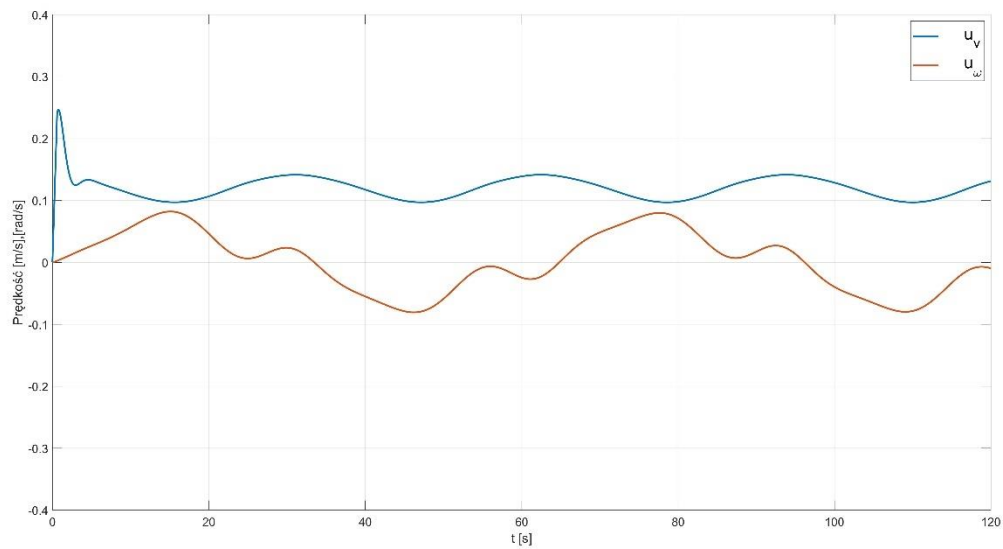
- w kierunku osi x : $e_x = 0.010$ m;
- w kierunku osi y : $e_y = 0.008$ m;
- orientacji: $e_\theta = 0.057$ rad;
- prędkości liniowej $e_v = 0.011 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.042 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 4.8$ s.

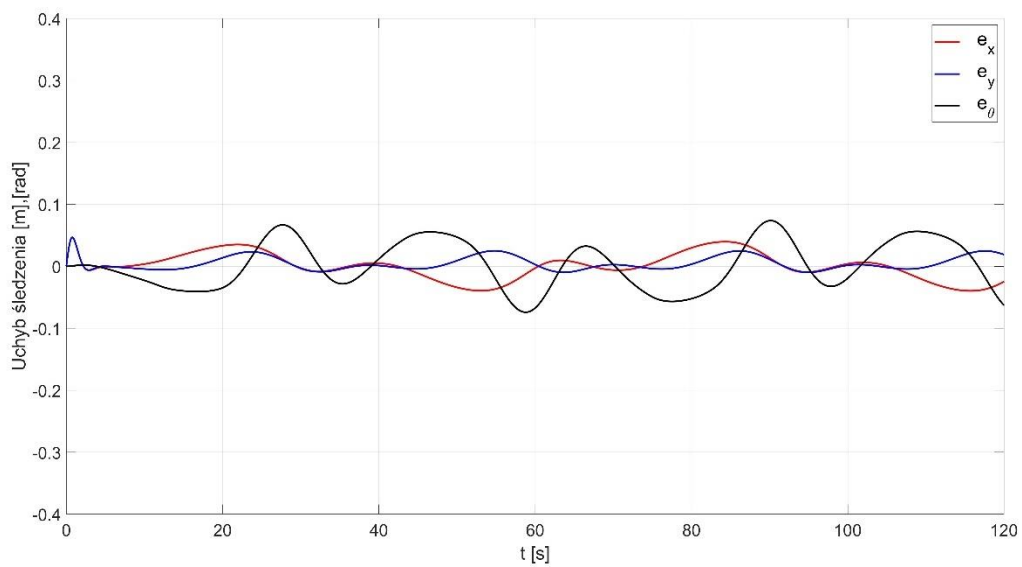
• Śledzenie trajektorii „sinus”



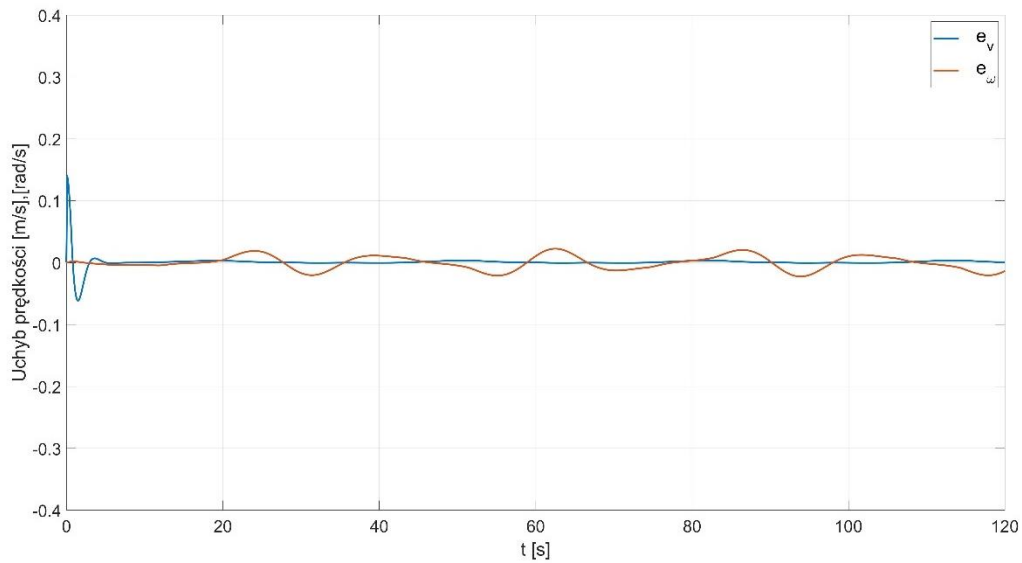
Rys. 5.8. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego stabilności śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 5.9. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 5.10. Uchyby śledzenia e_x , e_y , e_{θ} w symulacji śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego



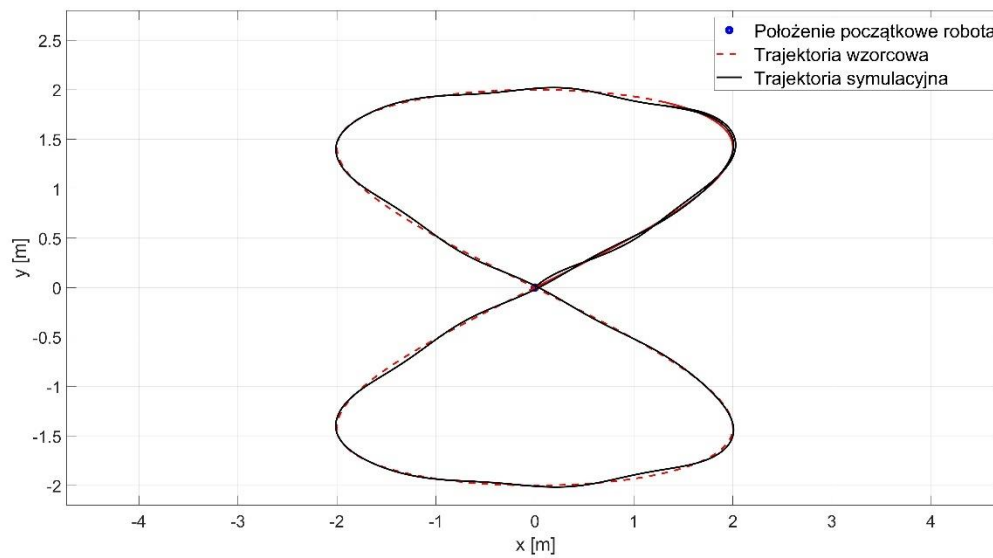
Rys. 5.11. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 5.10) oraz prędkości (Rys. 5.11) nie większymi niż:

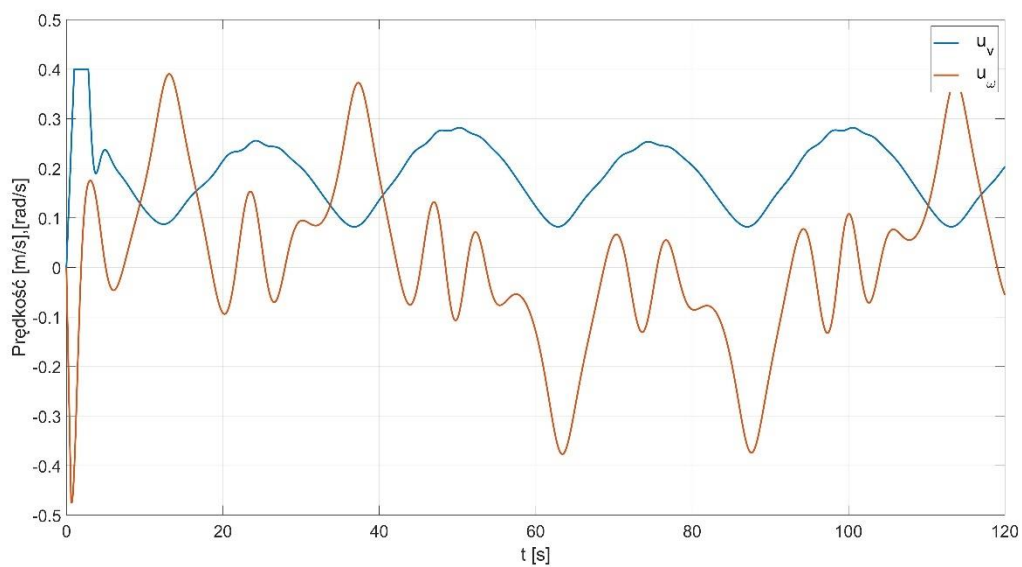
- w kierunku osi x : $e_x = 0.040$ m;
- w kierunku osi y : $e_y = 0.025$ m;
- orientacji: $e_\theta = 0.074$ rad;
- prędkości liniowej $e_v = 0.007 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.022 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 4.6$ s.

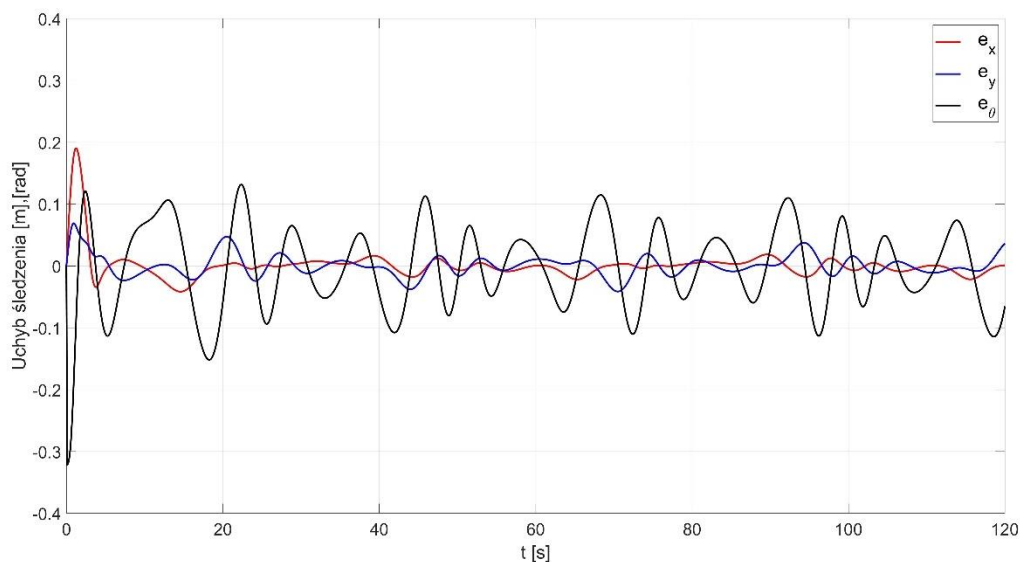
- **Śledzenie trajektorii „ósemka”**



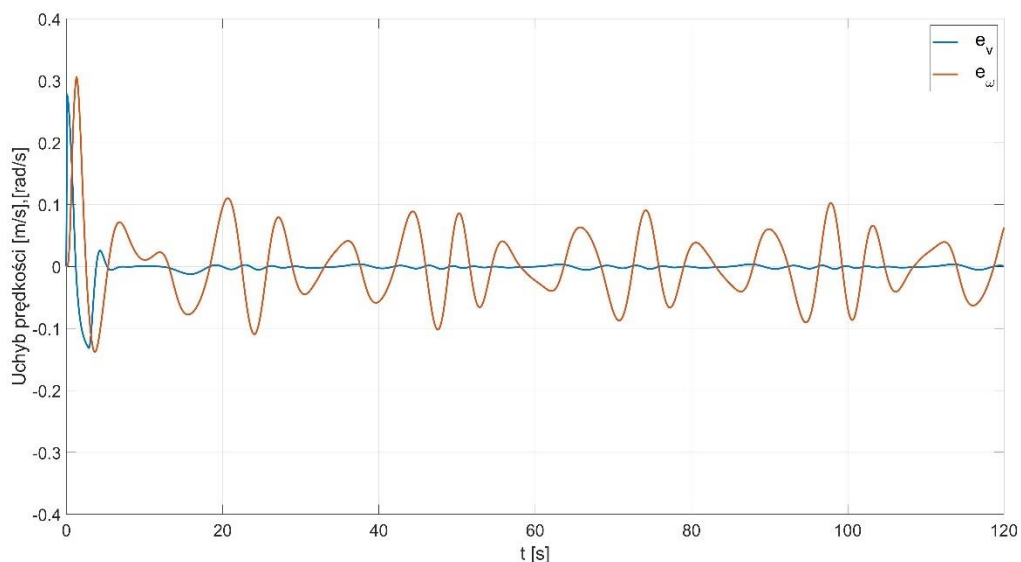
Rys. 5.12. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego stabilności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 5.13. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 5.14. Uchyby śledzenia e_x , e_y , e_θ w symulacji śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego



Rys. 5.15. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego

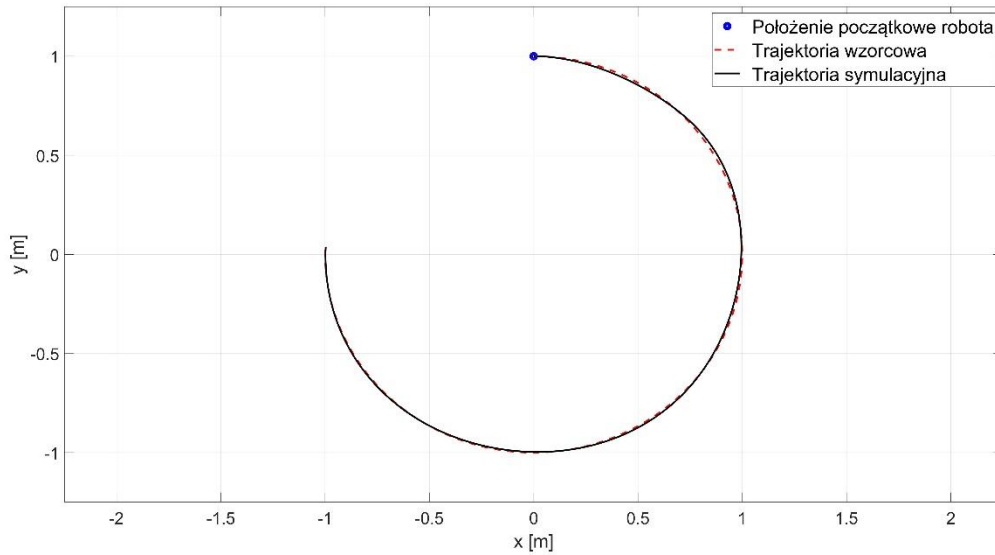
Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 5.14) oraz prędkości (Rys. 5.15) nie większymi niż:

- w kierunku osi x: $e_x = 0.041$ m;
- w kierunku osi y: $e_y = 0.047$ m;
- orientacji: $e_\theta = 0.152$ rad;
- prędkości liniowej $e_v = 0.012 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.111 \frac{\text{rad}}{\text{s}}$.

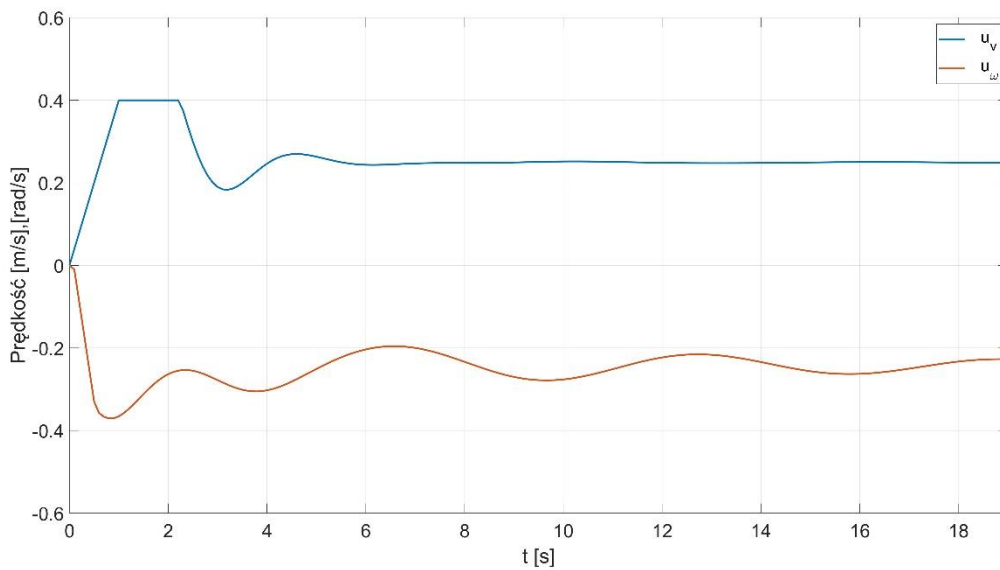
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 4.8$ s.

5.2.1.2. Kryterium ruchu po skończonej drodze dla kontrolera kinematycznego

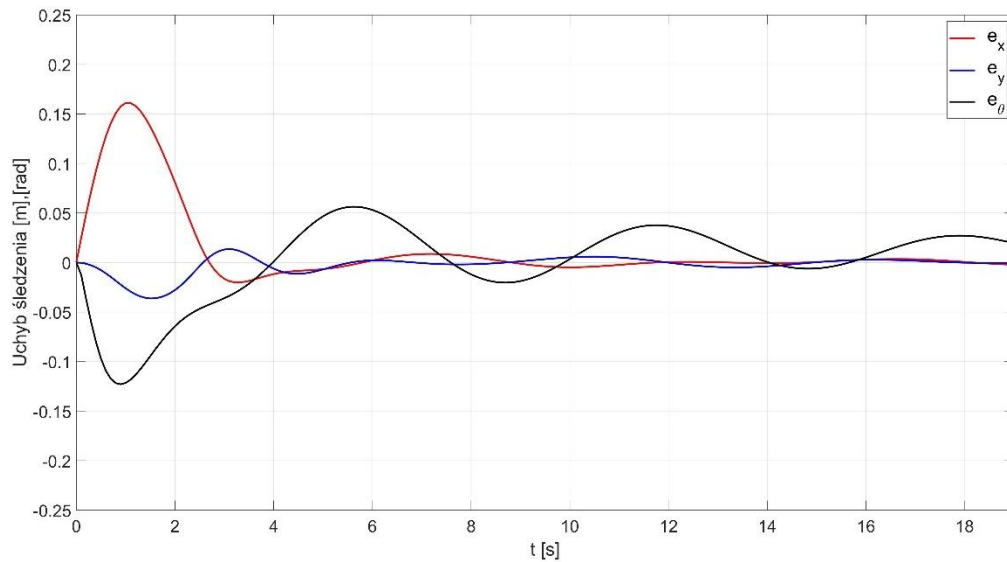
- Śledzenie trajektorii „okrąg”



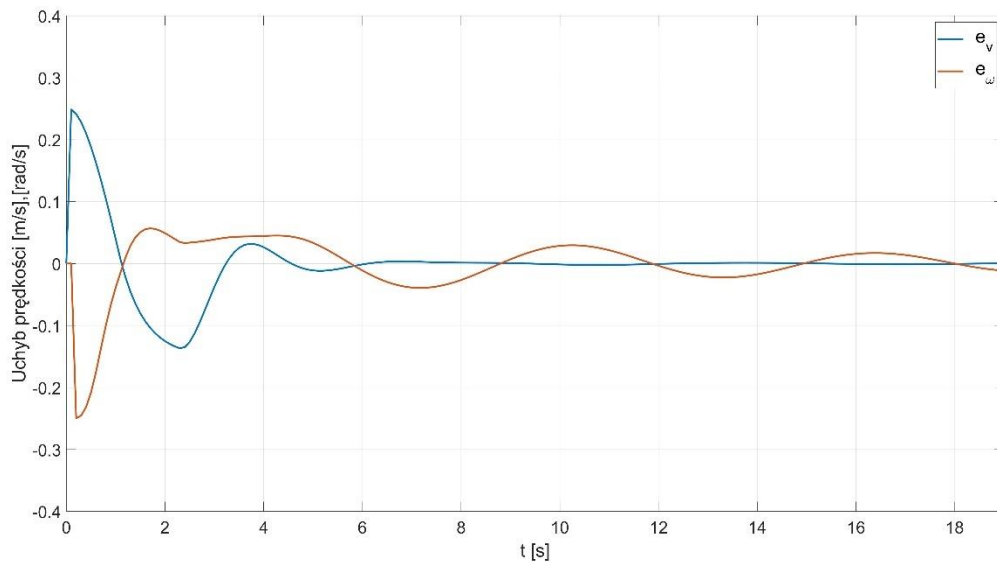
Rys. 5.16. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 5.17. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 5.18. Uchyby śledzenia e_x , e_y , e_θ w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego



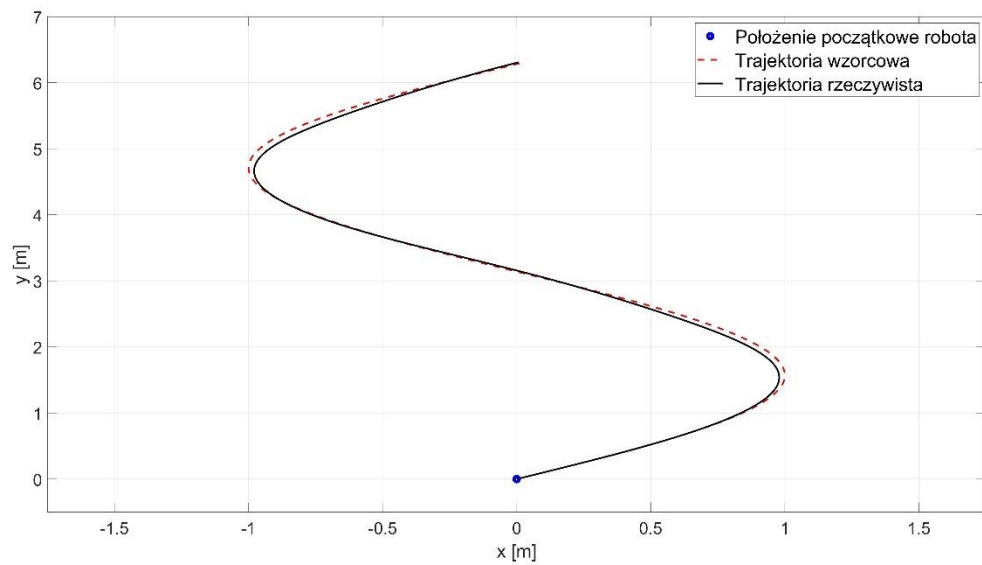
Rys. 5.19. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 5.18) oraz prędkości (Rys. 5.19) nie większymi niż:

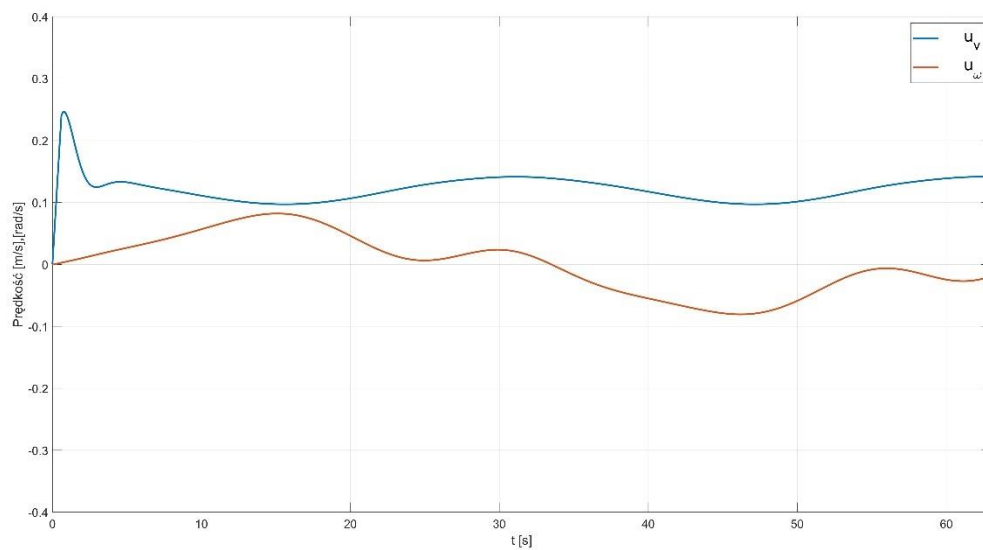
- w kierunku osi x: $e_x = 0.003$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.019$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.012 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 3.9$ s.

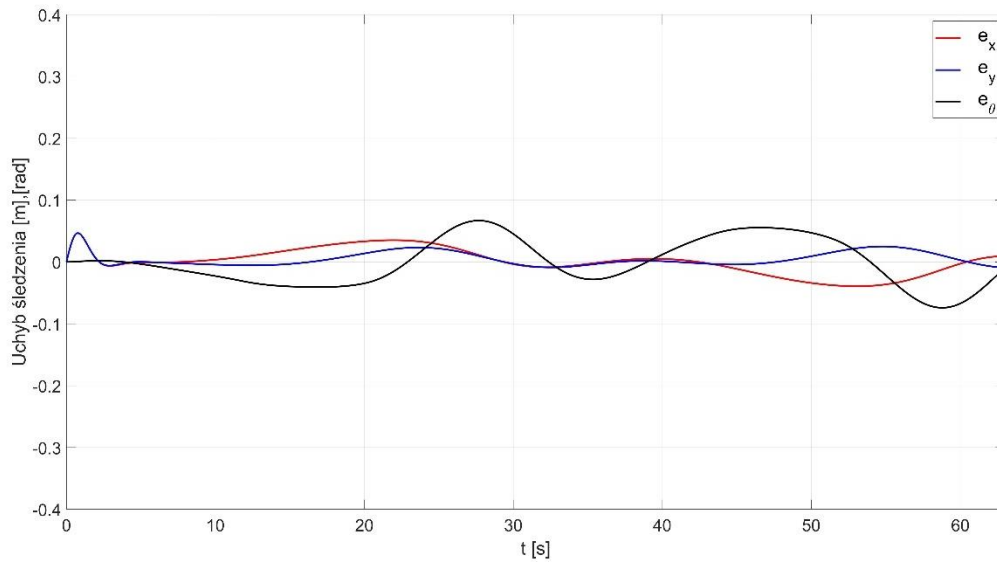
- Śledzenie trajektorii „sinus”



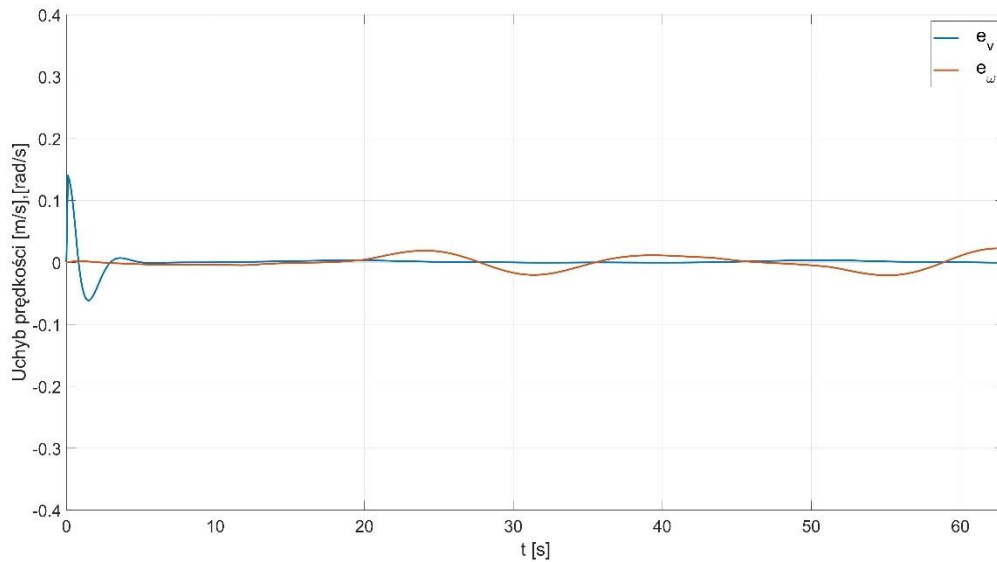
Rys. 5.20. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 5.21. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 5.22. Uchyby śledzenia e_x , e_y , e_θ w symulacji śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego



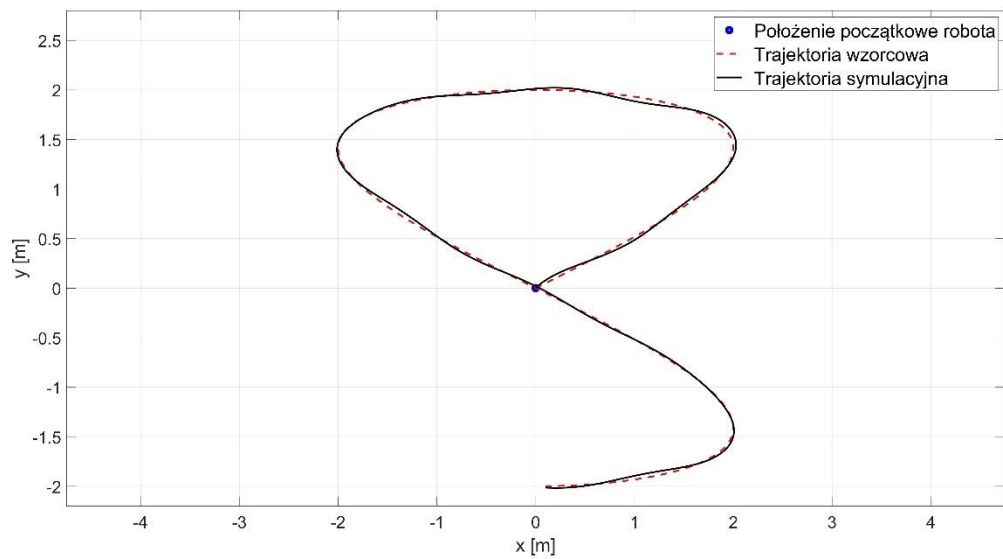
Rys. 5.23. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 5.22) oraz prędkości (Rys. 5.23) nie większymi niż:

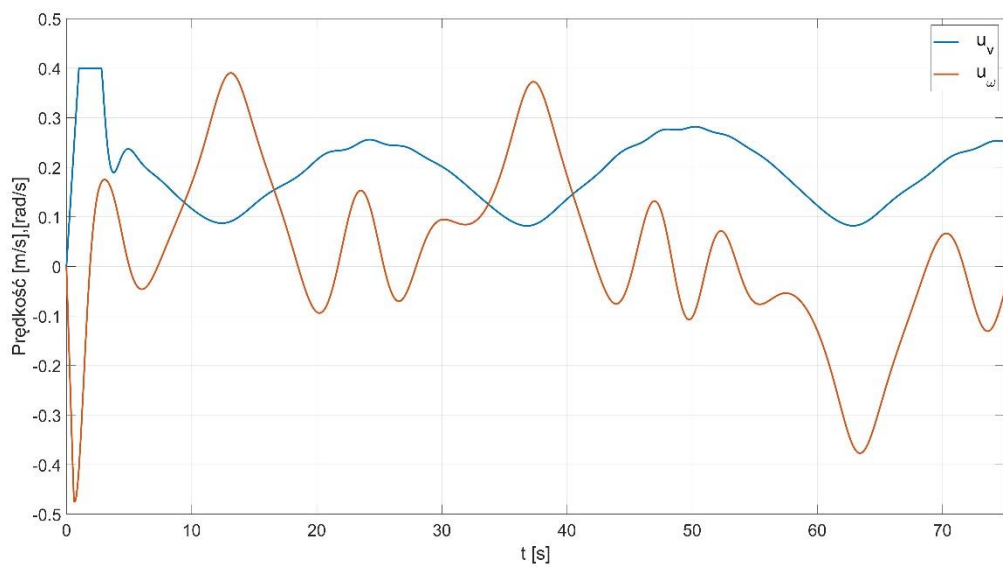
- w kierunku osi x: $e_x = 0.010$ m;
- w kierunku osi y: $e_y = 0.009$ m;
- orientacji: $e_\theta = 0.012$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.022 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 1.9$ s.

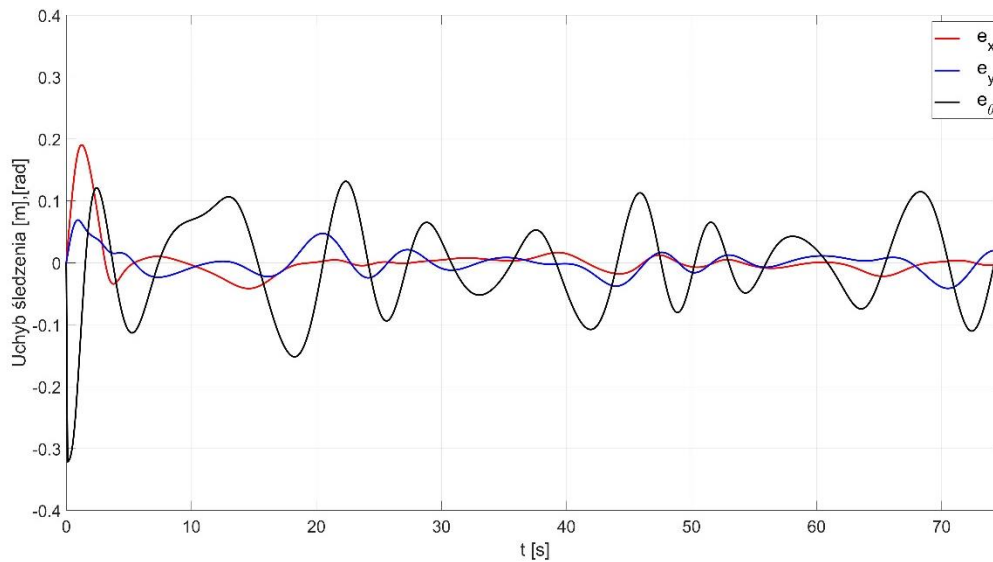
- Śledzenie trajektorii „ósemka”



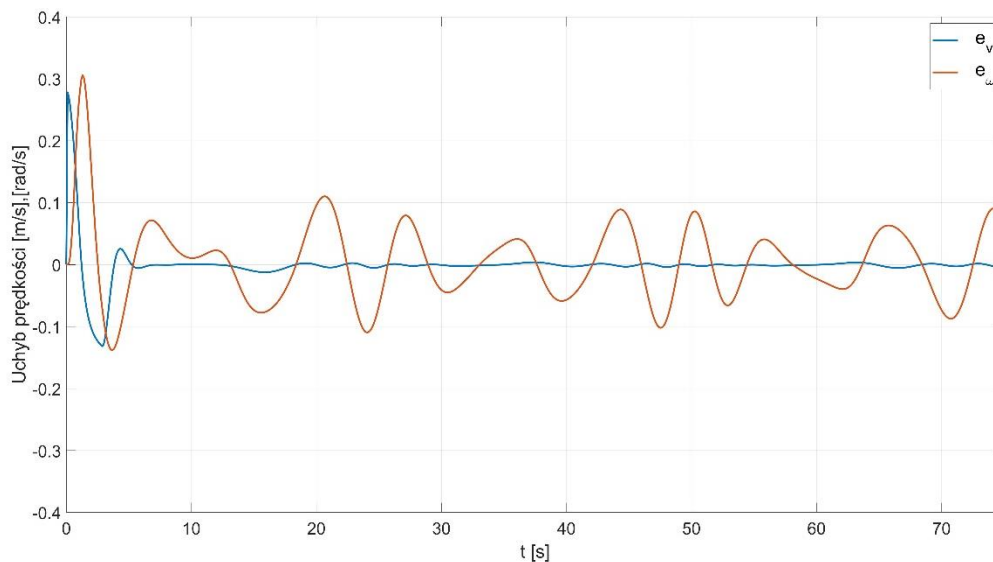
Rys. 5.24. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 5.25. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 5.26. Uchyby śledzenia e_x , e_y , e_θ w badaniu symulacyjnym dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 5.27. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 5.26) oraz prędkości (Rys. 5.27) nie większymi niż:

- w kierunku osi x: $e_x = 0.001$ m;
- w kierunku osi y: $e_y = 0.014$ m;
- orientacji: $e_\theta = 0.058$ rad;
- prędkości liniowej $e_v = 0.004 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.062 \frac{\text{rad}}{\text{s}}$.

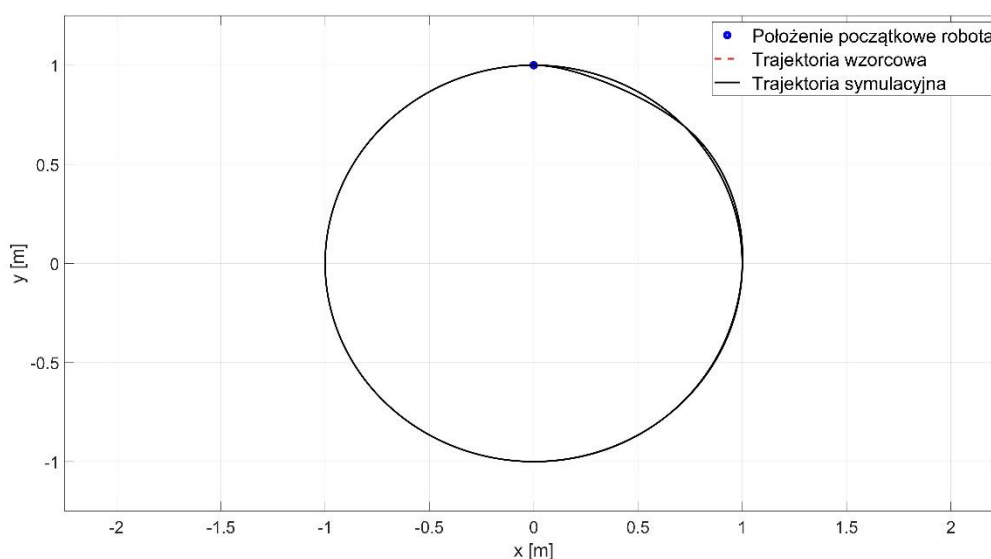
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 3.6$ s.

5.2.2. Badania kontrolera kinematyczno-dynamicznego

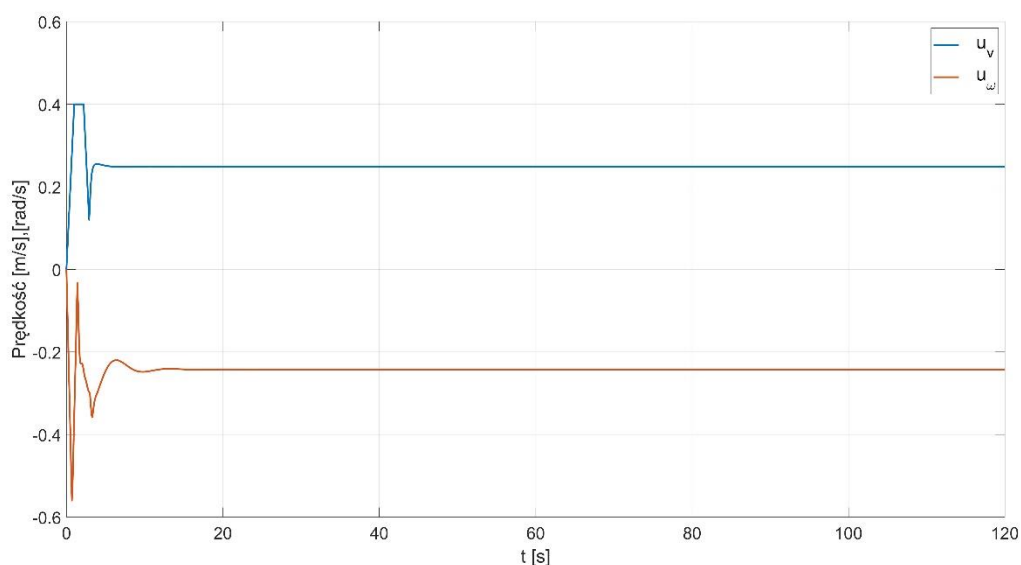
Wyniki badań symulacyjnych śledzenia trajektorii wzorcowej za pomocą kontrolera kinematyczno-dynamicznego przedstawiono na Rys. 5.28 - Rys. 5.51. Badania przeprowadzono według schematu przedstawionego na Rys. 5.2.

5.2.2.1. Kryterium ruchu przez określony czas dla kontrolera kinematyczno-dynamicznego

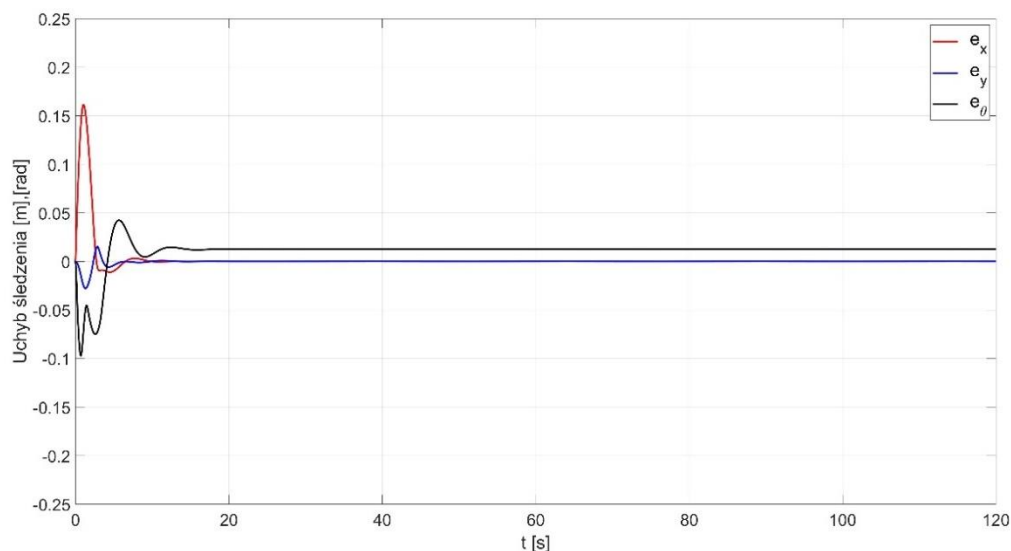
- Śledzenie trajektorii „okrąg”



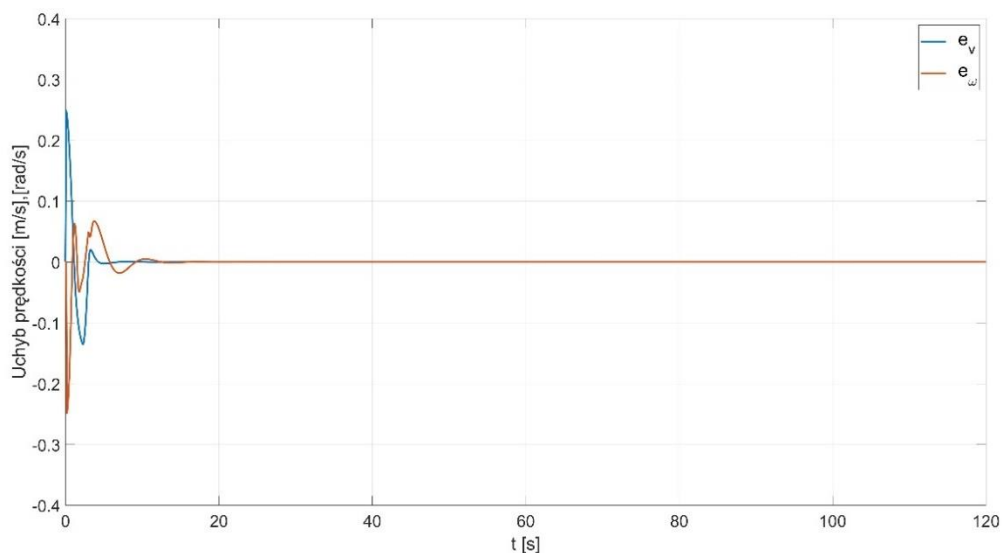
Rys. 5.28. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego stabilności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.29. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.30. Uchyby śledzenia e_x , e_y , e_θ w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



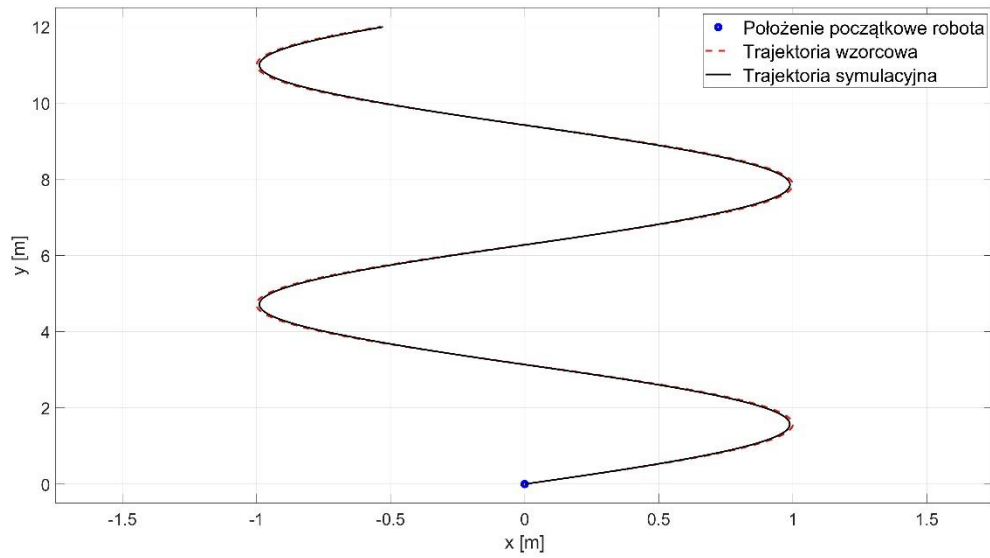
Rys. 5.31. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 5.30) oraz prędkości (Rys. 5.31) nie większymi niż:

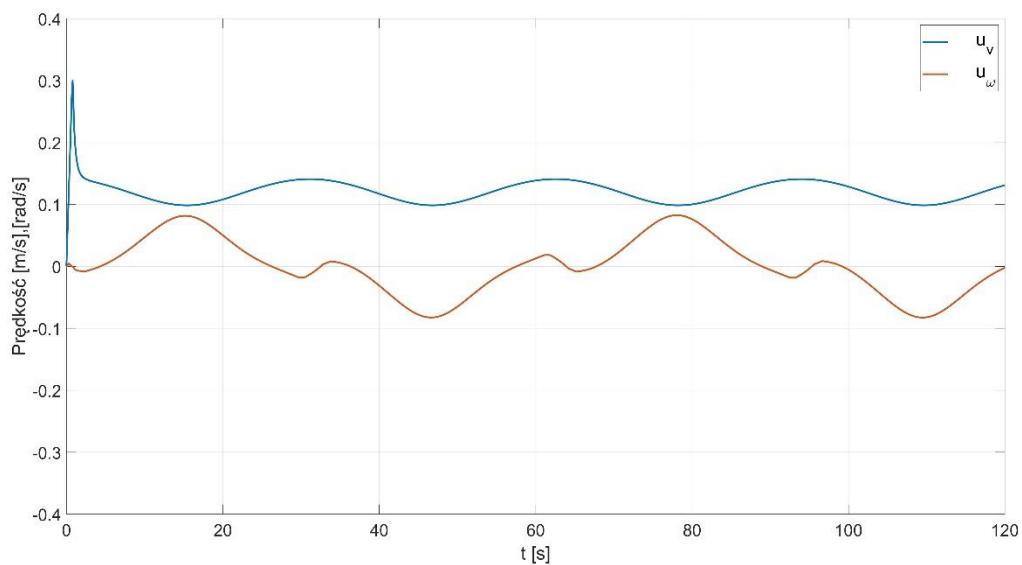
- w kierunku osi x: $e_x = 0.009$ m;
- w kierunku osi y: $e_y = 0.008$ m;
- orientacji: $e_\theta = 0.063$ rad;
- prędkości liniowej $e_v = 0.008 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.033 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 3.2$ s.

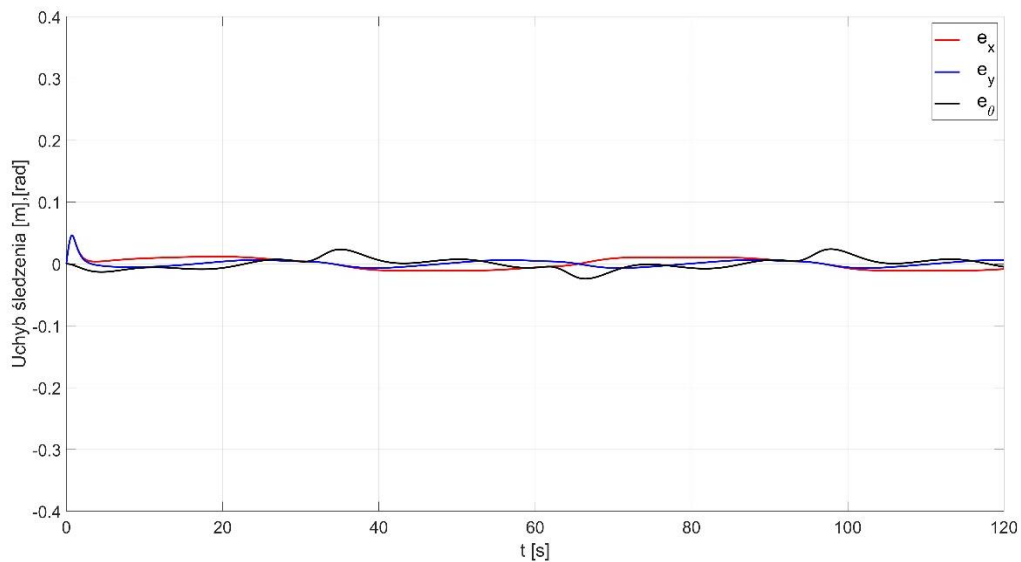
- Śledzenie trajektorii „sinus”



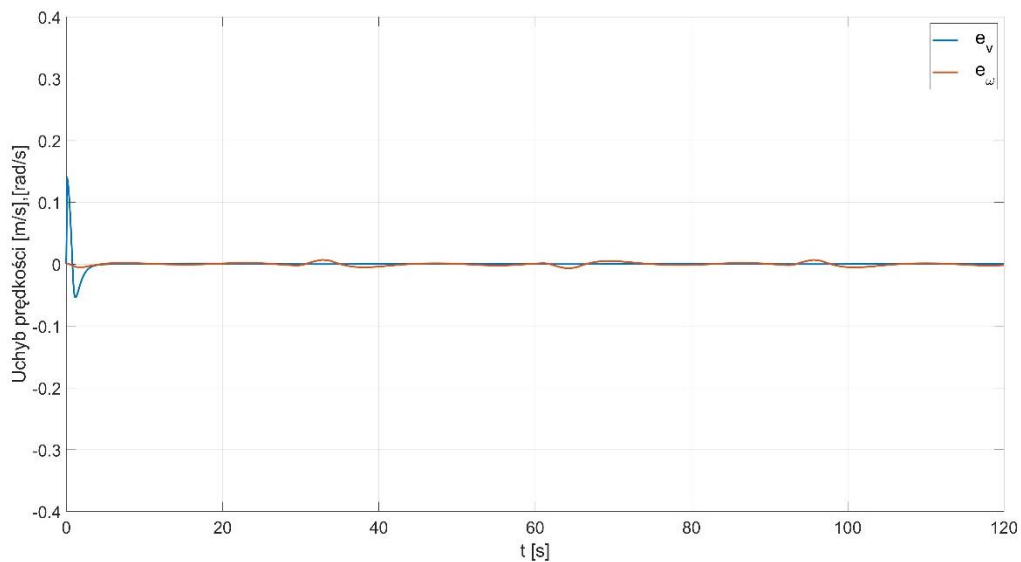
Rys. 5.32. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego stabilności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.33. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.34. Uchyby śledzenia e_x , e_y , e_θ w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



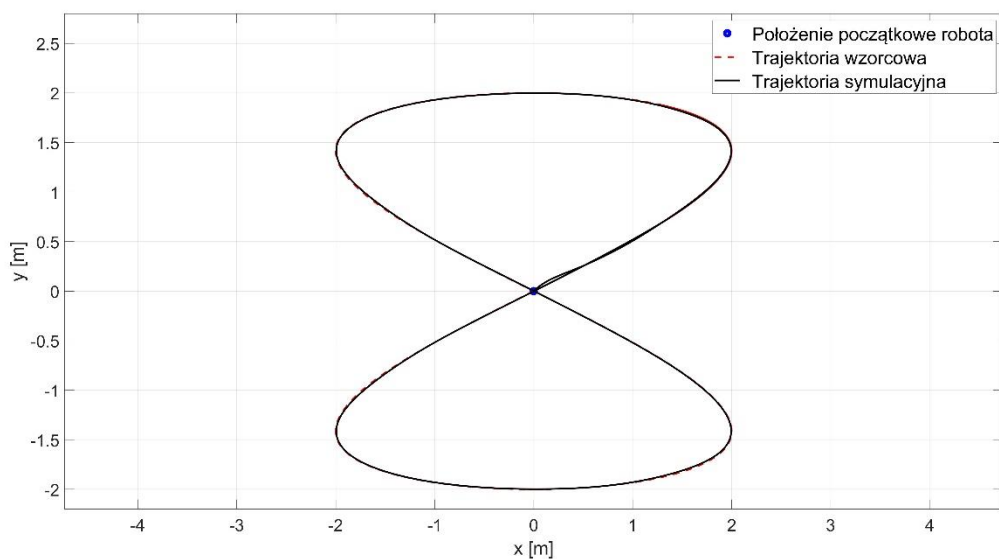
Rys. 5.35. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 5.34) oraz prędkości (Rys. 5.35) nie większymi niż:

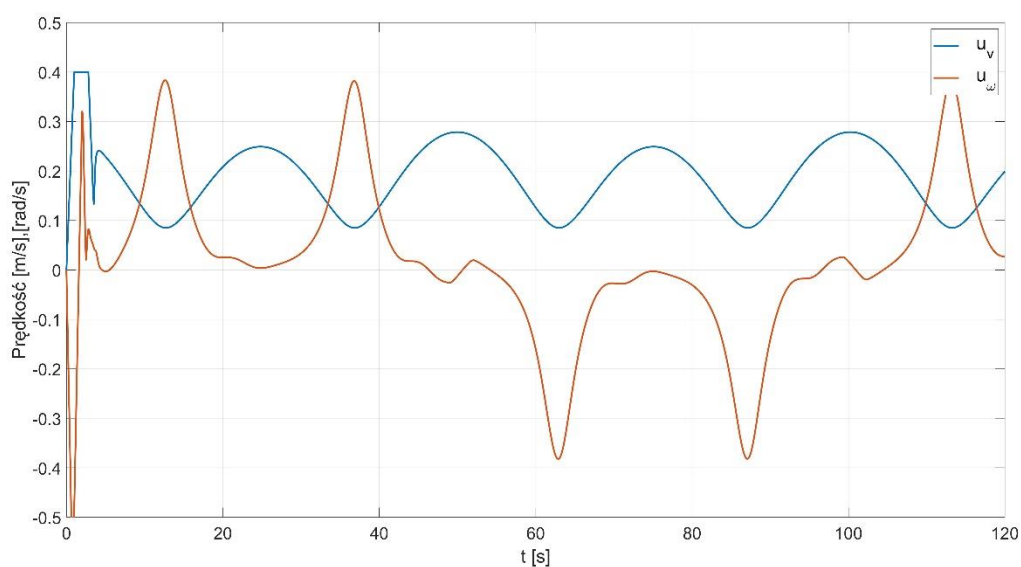
- w kierunku osi x: $e_x = 0.012$ m;
- w kierunku osi y: $e_y = 0.007$ m;
- orientacji: $e_\theta = 0.024$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.007 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 2.3$ s.

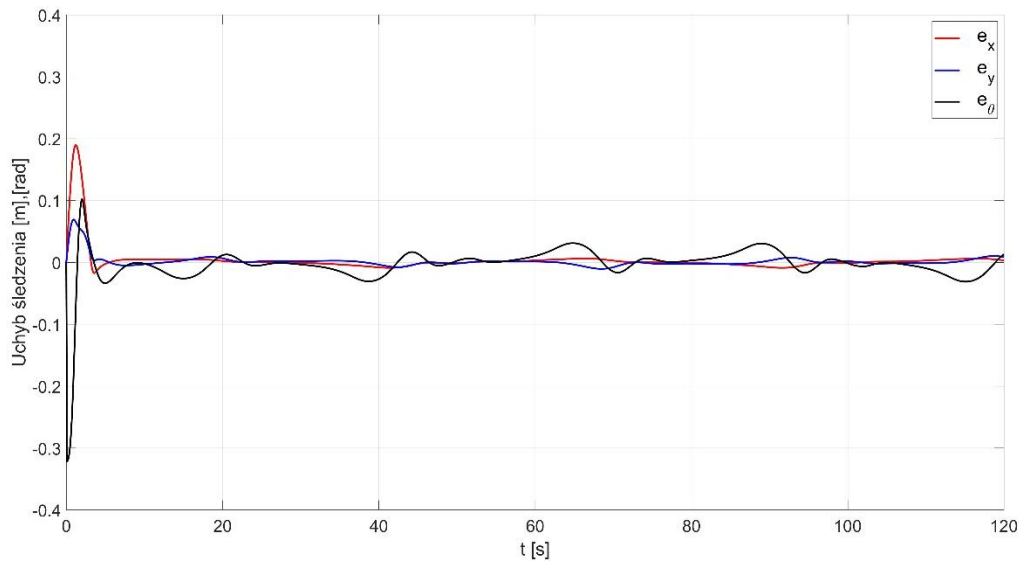
- Śledzenie trajektorii „ósemka”



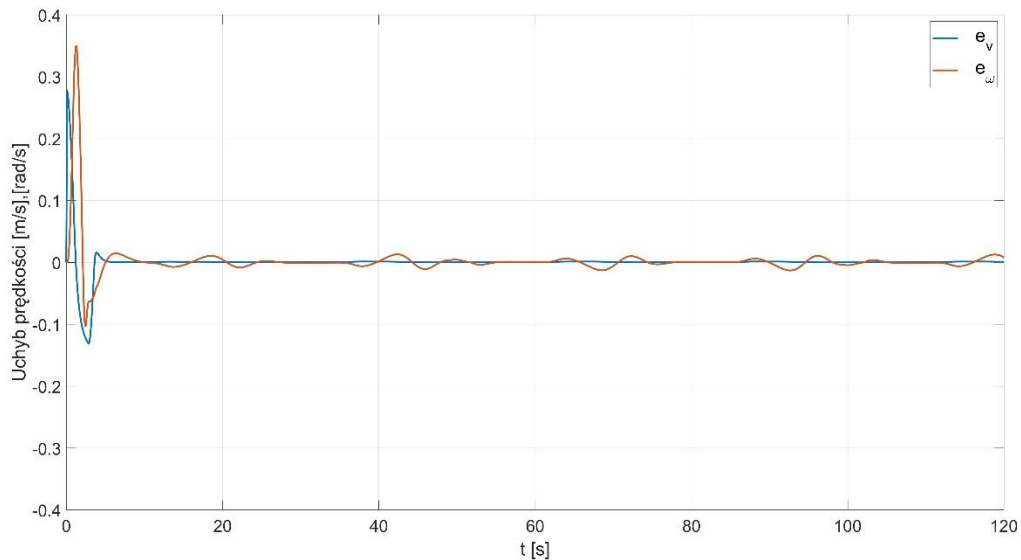
Rys. 5.36. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego stabilności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.37. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.38. Uchyby śledzenia e_x , e_y , e_θ w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.39. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

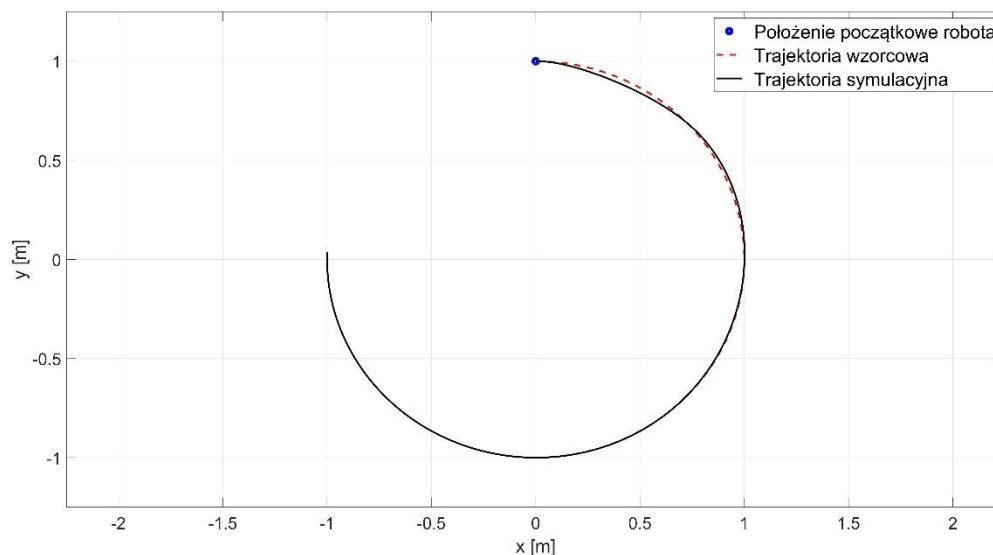
Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler dynamiczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 5.38) oraz prędkości (Rys. 5.39) nie większymi niż:

- w kierunku osi x: $e_x = 0.009$ m;
- w kierunku osi y: $e_y = 0.010$ m;
- orientacji: $e_\theta = 0.031$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.013 \frac{\text{rad}}{\text{s}}$.

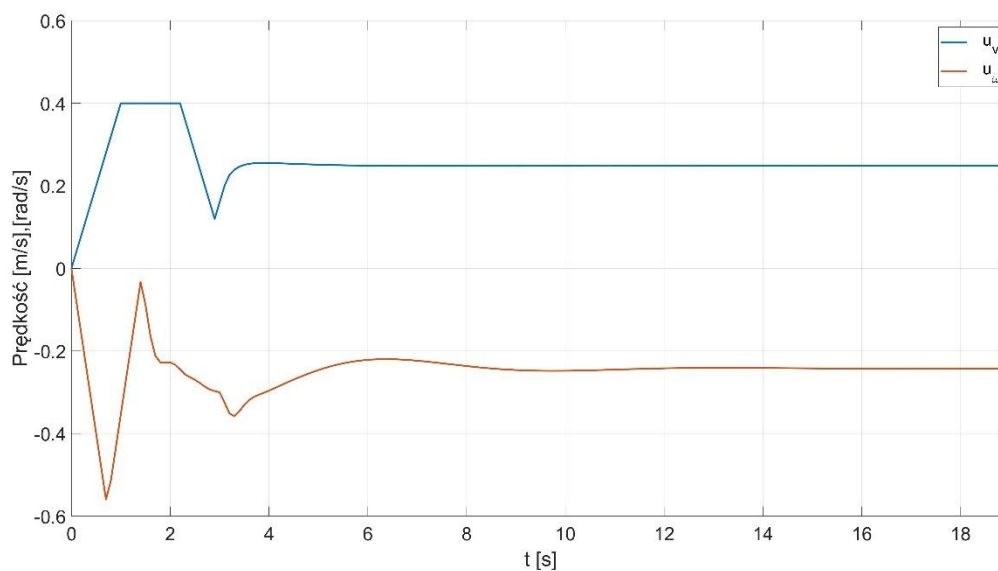
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 3.4$ s.

5.2.2.2. Kryterium ruchu po skończonej drodze dla kontrolera kinematyczno-dynamicznego

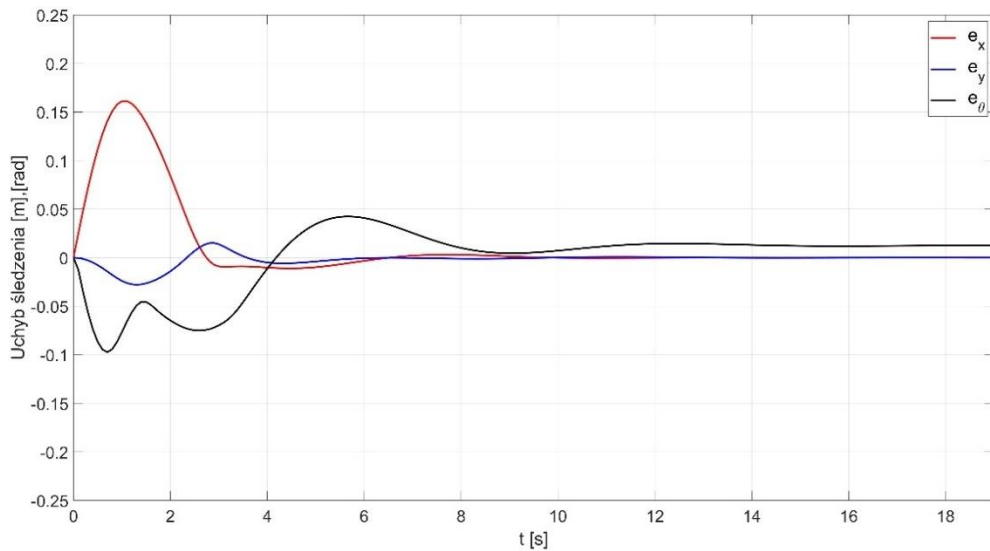
- Śledzenie trajektorii „okrąg”



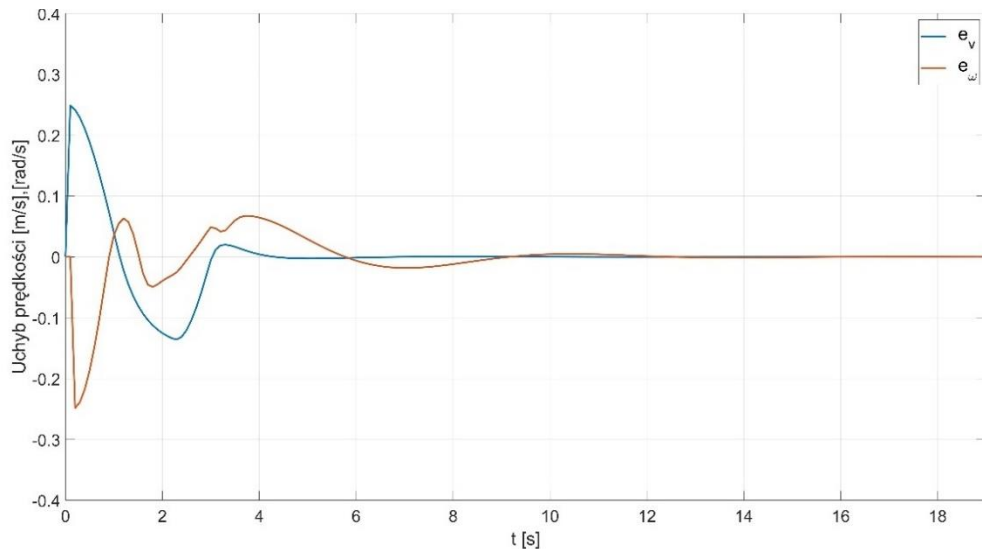
Rys. 5.40. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.41. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.42. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



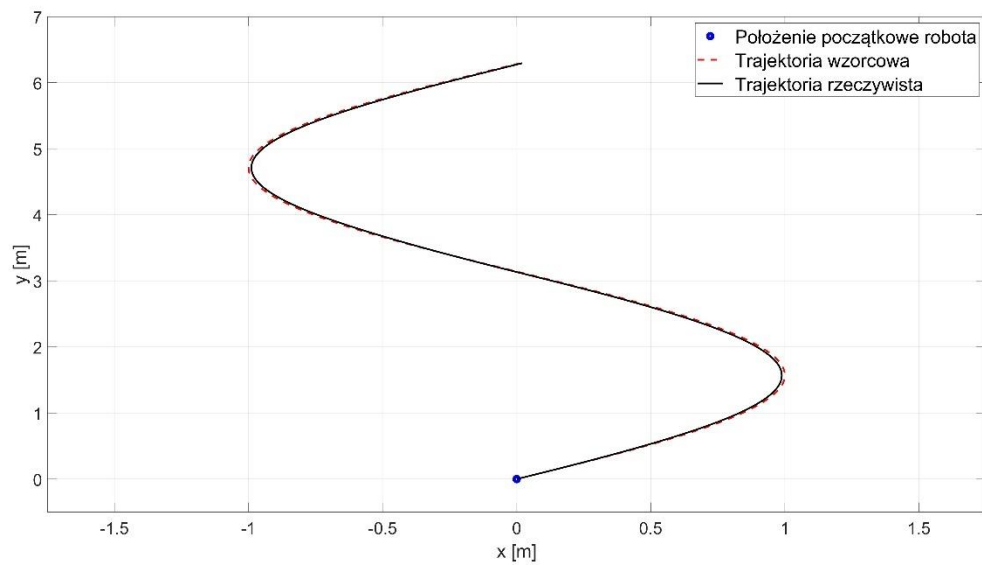
Rys. 5.43. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 5.42) oraz prędkości (Rys. 5.43) nie większymi niż:

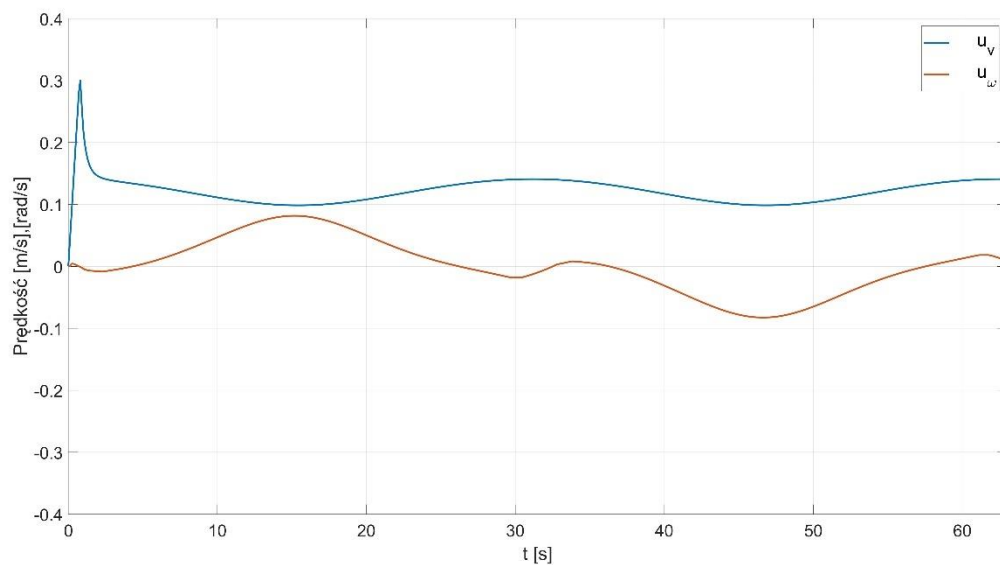
- w kierunku osi x: $e_x = 0.001$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.012$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.001 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 3.1$ s.

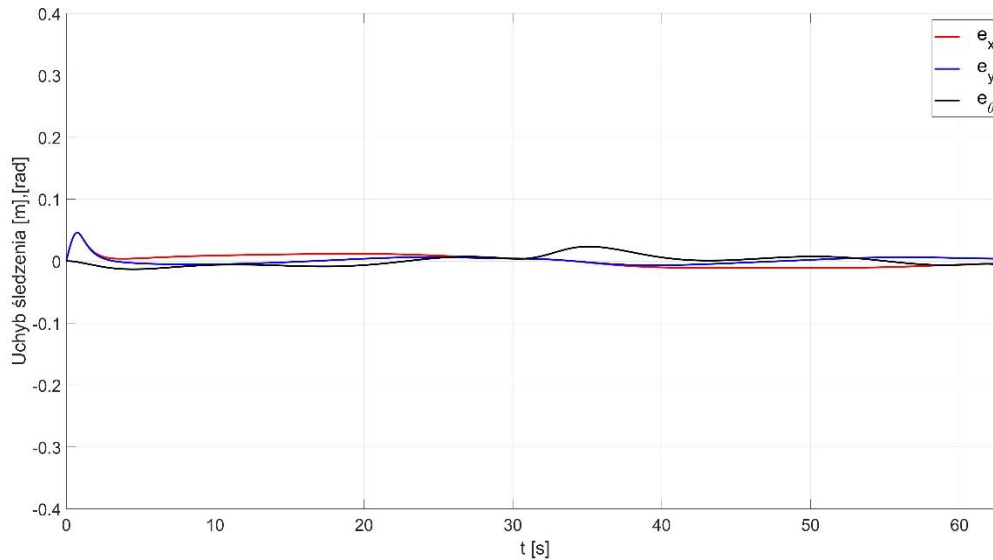
- Śledzenie trajektorii „sinus”



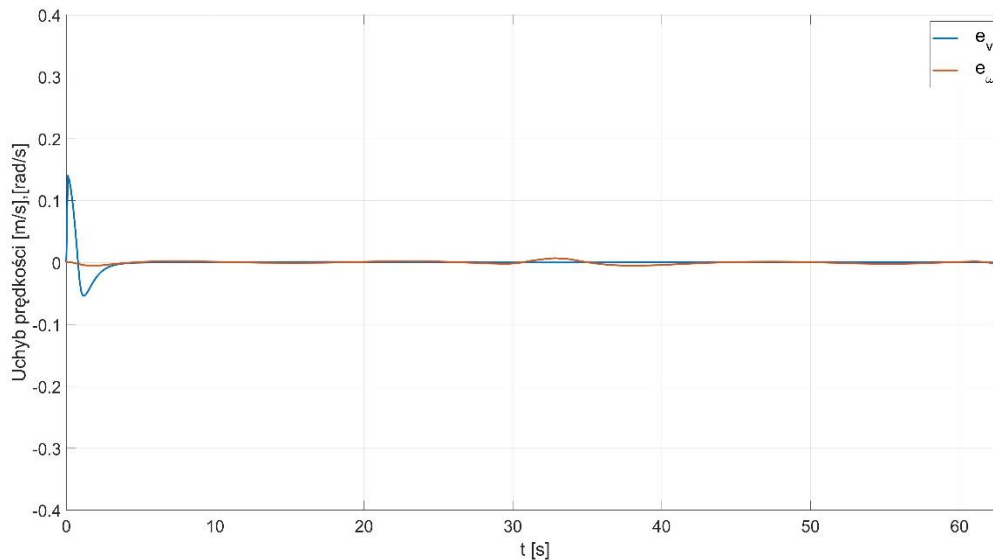
Rys. 5.44. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.45. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.46. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



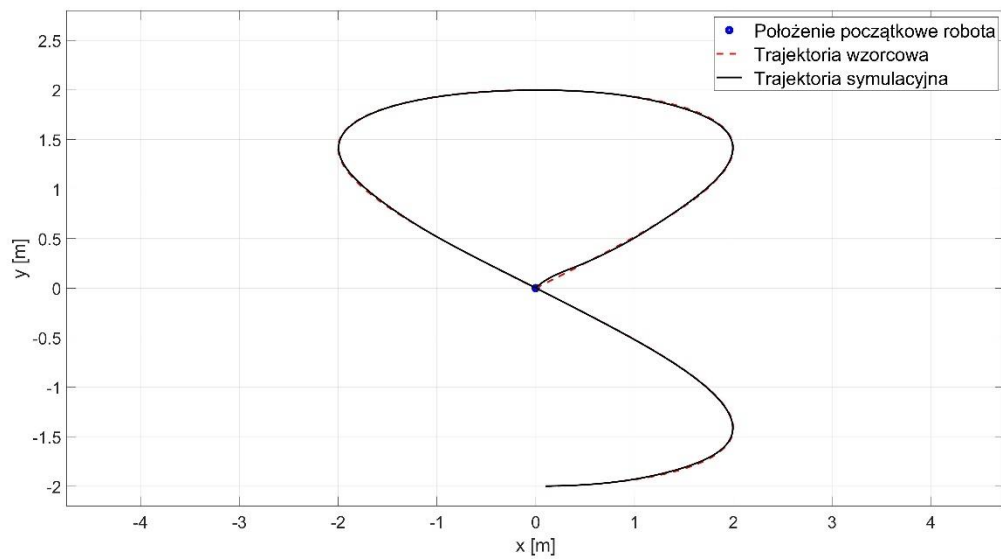
Rys. 5.47. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 5.46) oraz prędkości (Rys. 5.47) nie większymi niż:

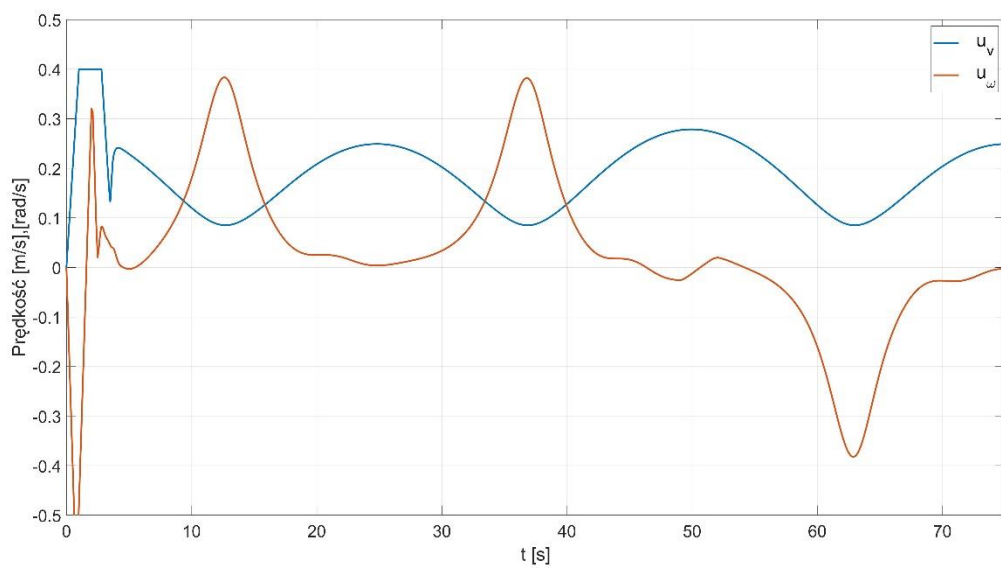
- w kierunku osi x: $e_x = 0.004$ m;
- w kierunku osi y: $e_y = 0.004$ m;
- orientacji: $e_\theta = 0.007$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.004 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 1.4$ s.

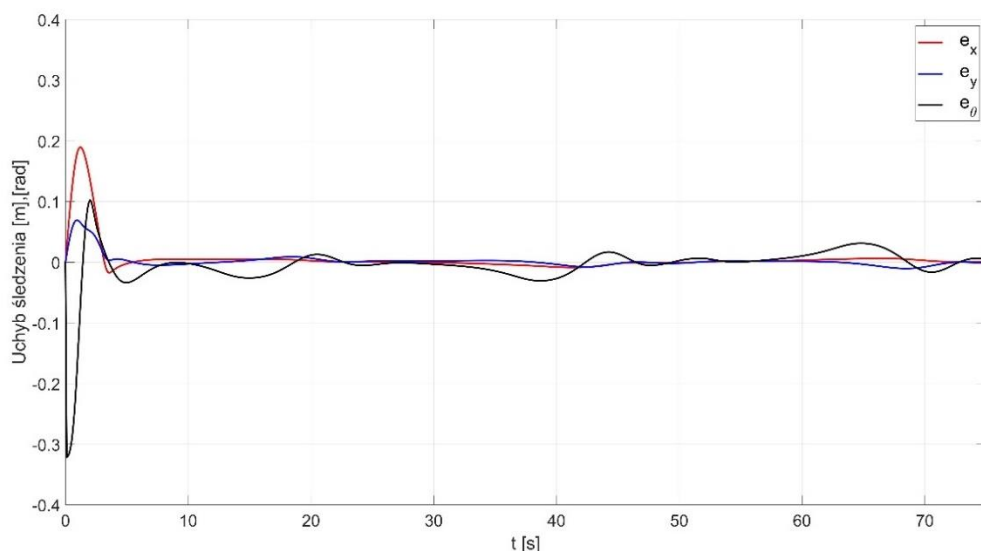
- Śledzenie trajektorii „ósemka”



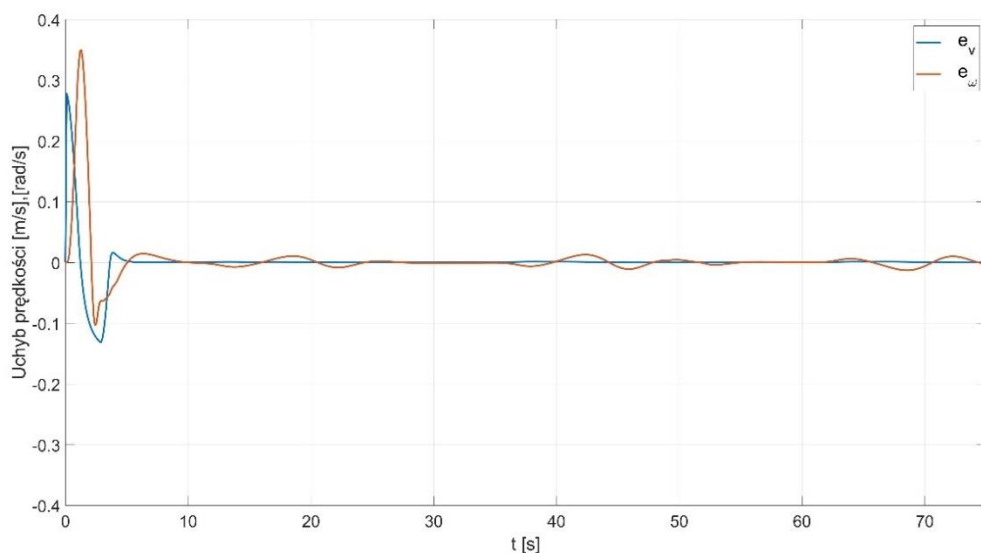
Rys. 5.48. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.49. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.50. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 5.51. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 5.50) oraz prędkości (Rys. 5.51) nie większymi niż:

- w kierunku osi x: $e_x = 0.001$ m;
- w kierunku osi y: $e_y = 0.002$ m;
- orientacji: $e_\theta = 0.006$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.002 \frac{\text{rad}}{\text{s}}$.

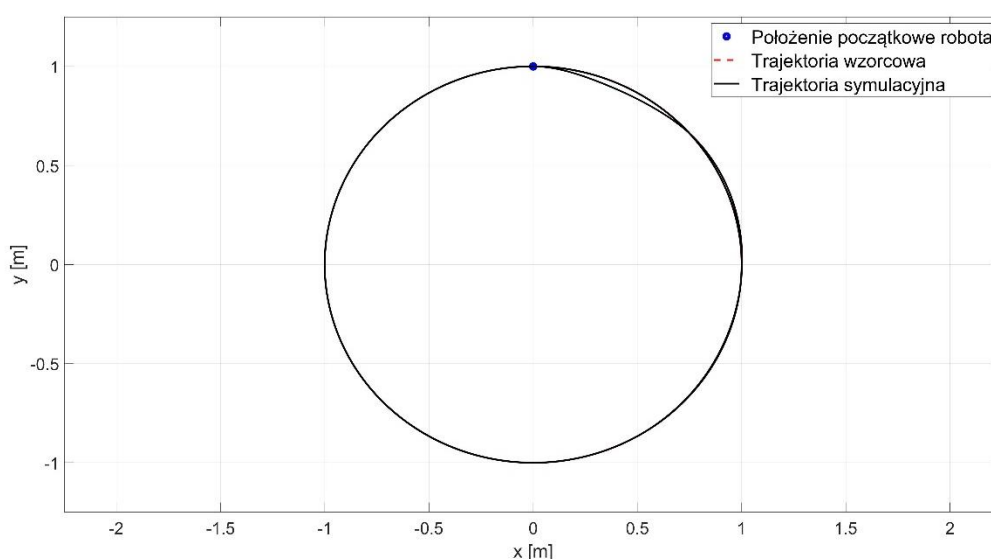
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 3.1$ s.

5.2.3. Badania kontrolera kinematyczno-dynamicznego z adaptacją

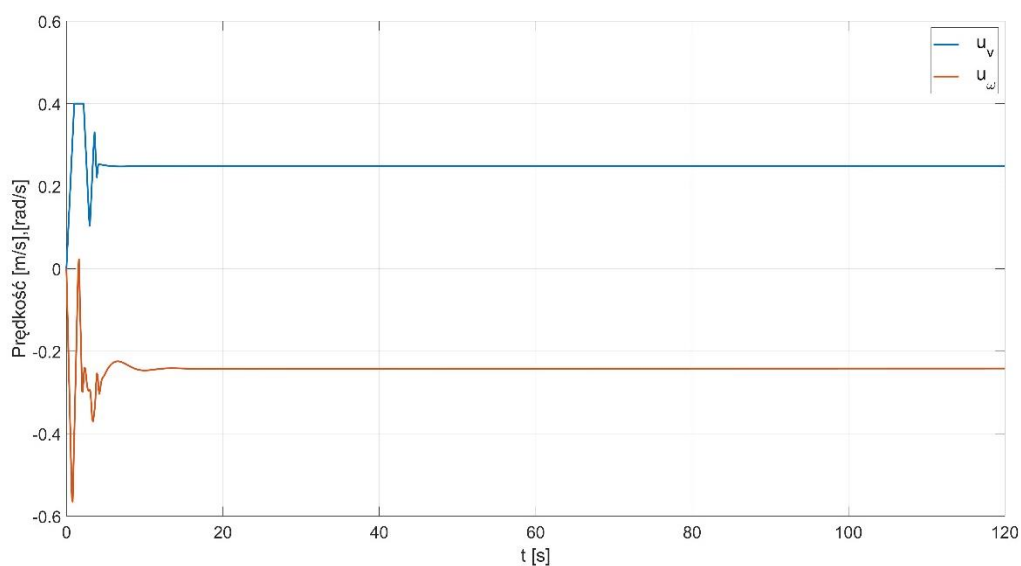
Wyniki badań symulacyjnych śledzenia trajektorii wzorcowej za pomocą kontrolera kinematyczno-dynamicznego z adaptacją przedstawiono na Rys. 5.52 - Rys. 5.75. Badania przeprowadzono według schematu przedstawionego na Rys. 5.3

5.2.3.1. Kryterium ruchu przez określony czas dla kontrolera kinematyczno-dynamicznego z adaptacją

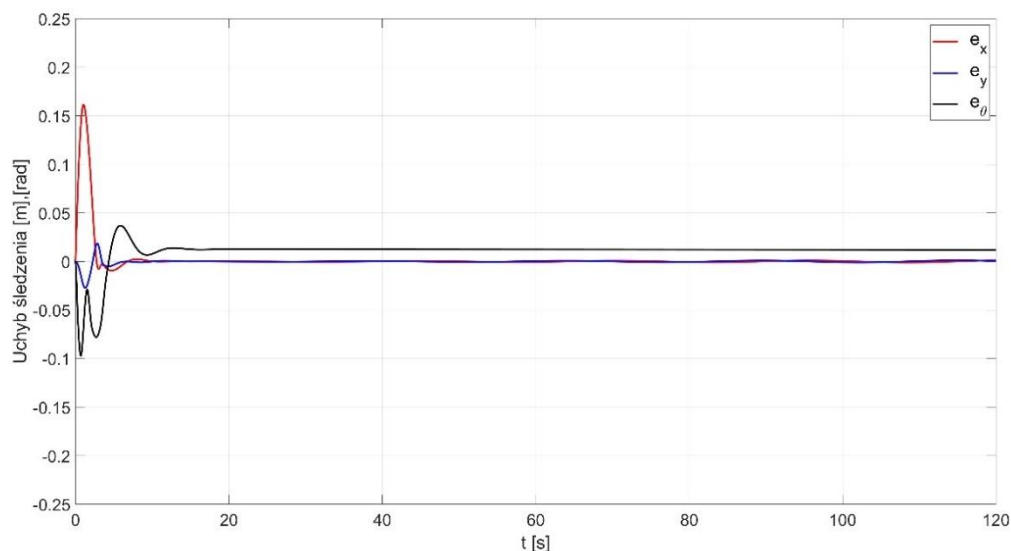
- Śledzenie trajektorii „okrąg”



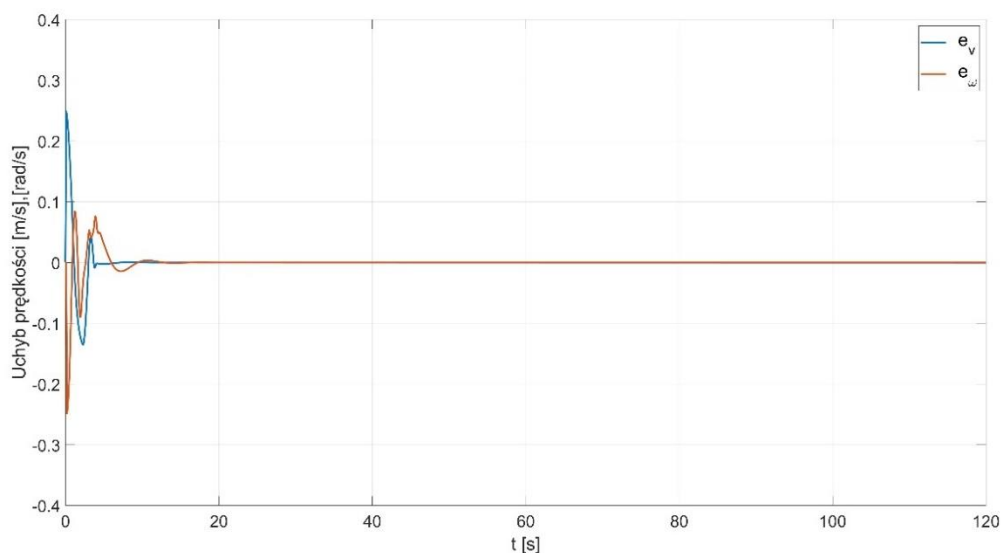
Rys. 5.52. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.53. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.54. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



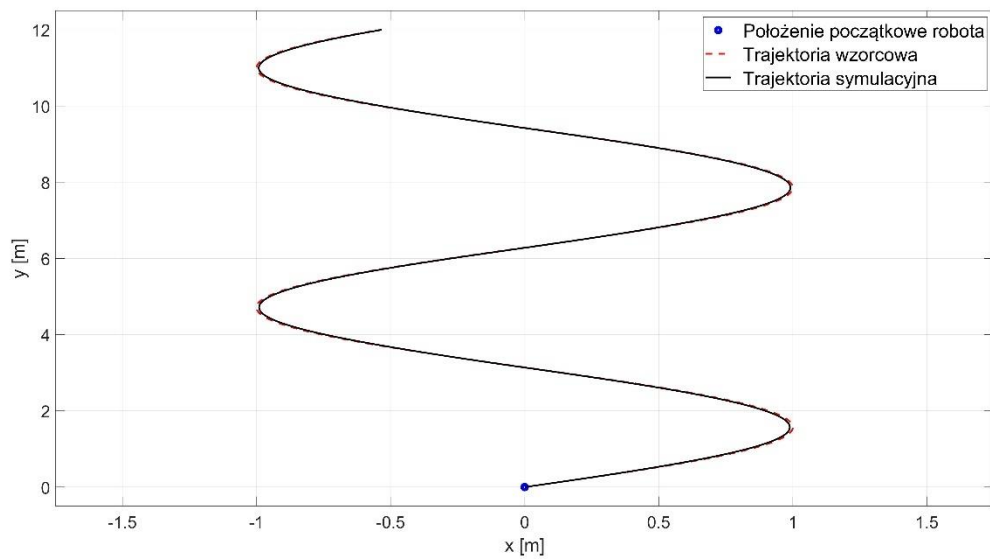
Rys. 5.55. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 5.54) oraz prędkości (Rys. 5.55) nie większymi niż:

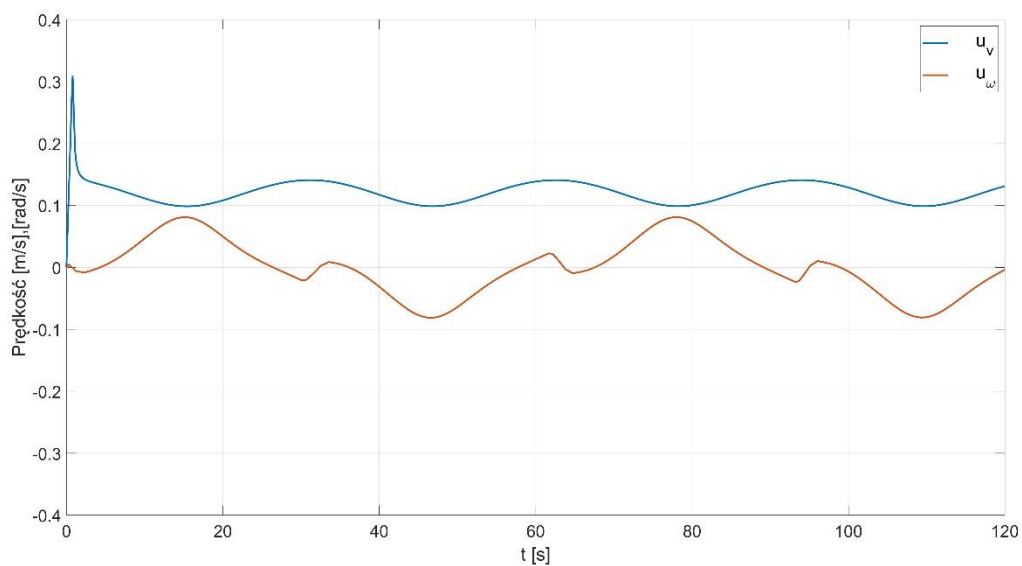
- w kierunku osi x: $e_x = 0.004$ m;
- w kierunku osi y: $e_y = 0.006$ m;
- orientacji: $e_\theta = 0.037$ rad;
- prędkości liniowej $e_v = 0.008 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.014 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 2.4$ s.

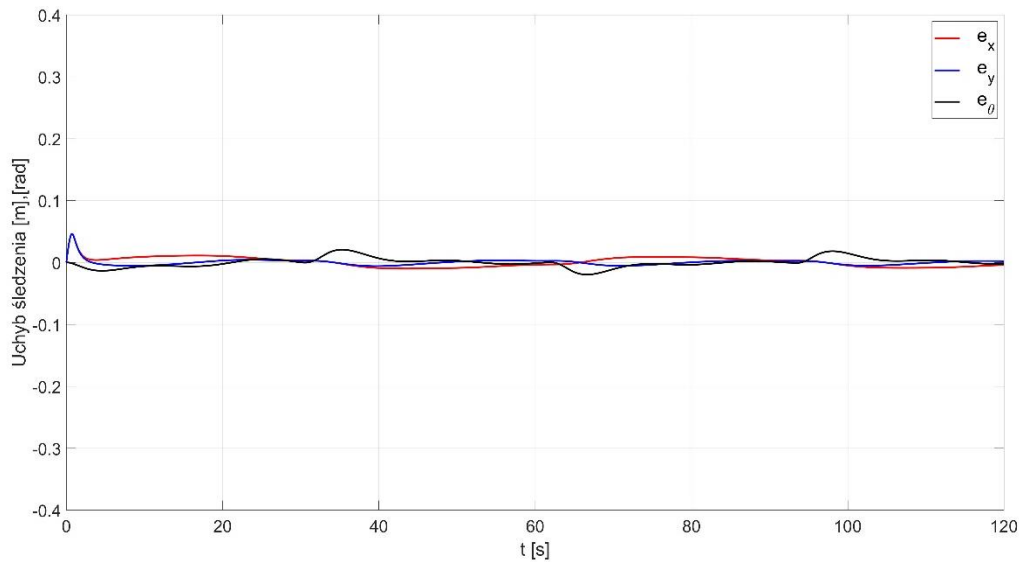
- Śledzenie trajektorii „sinus”



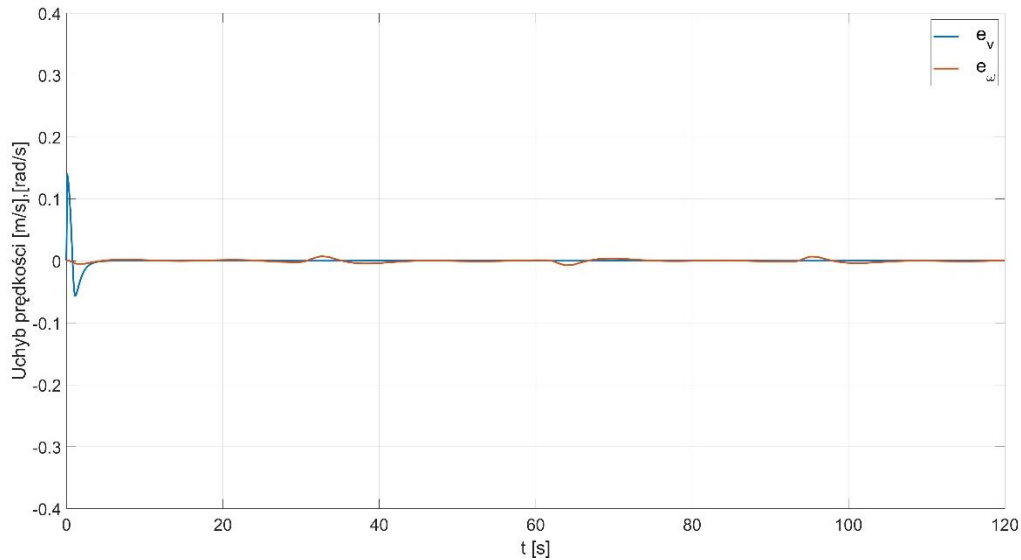
Rys. 5.56. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.57. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.58. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



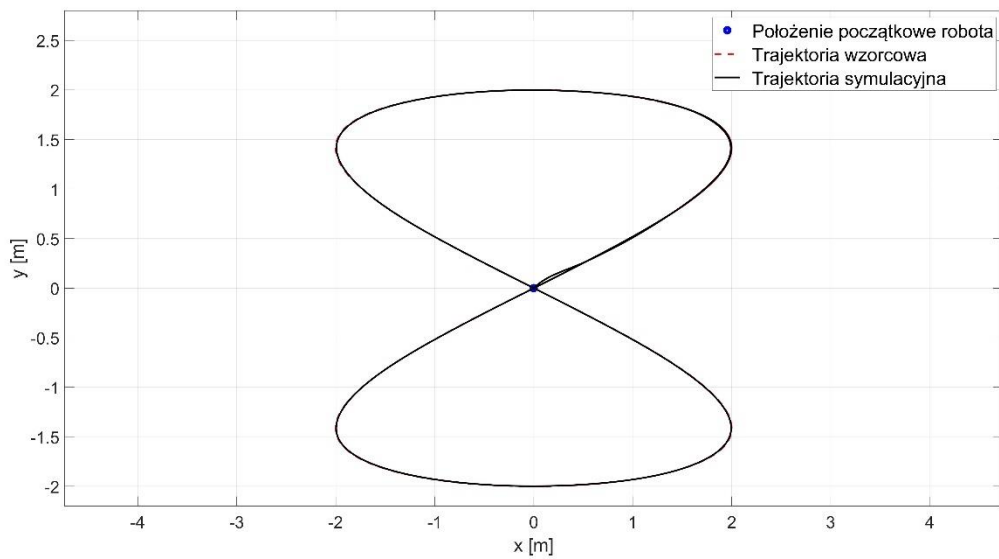
Rys. 5.59. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 5.58) oraz prędkości (Rys. 5.59) nie większymi niż:

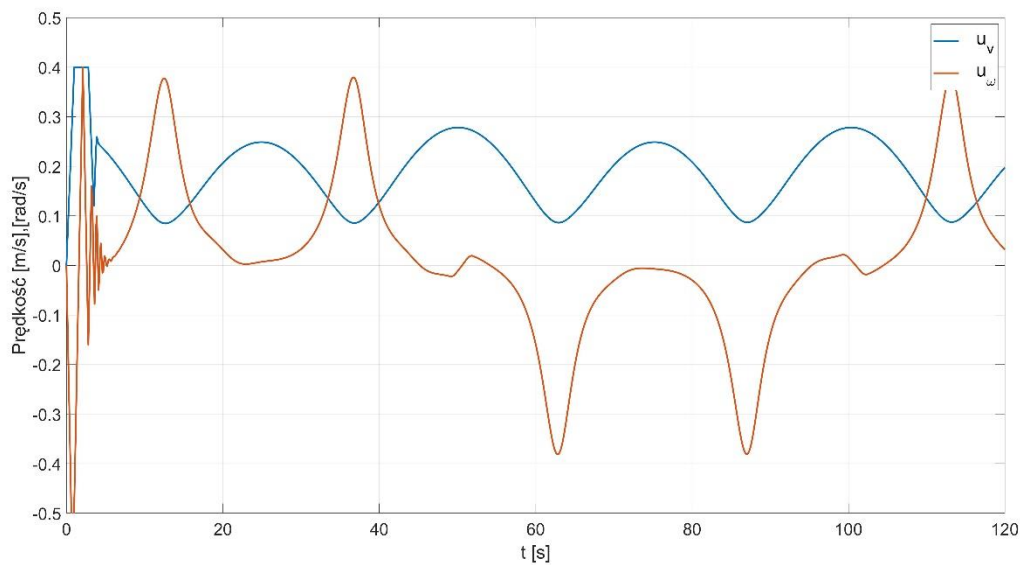
- w kierunku osi x: $e_x = 0.011$ m;
- w kierunku osi y: $e_y = 0.006$ m;
- orientacji: $e_\theta = 0.021$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.007 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 1.9$ s.

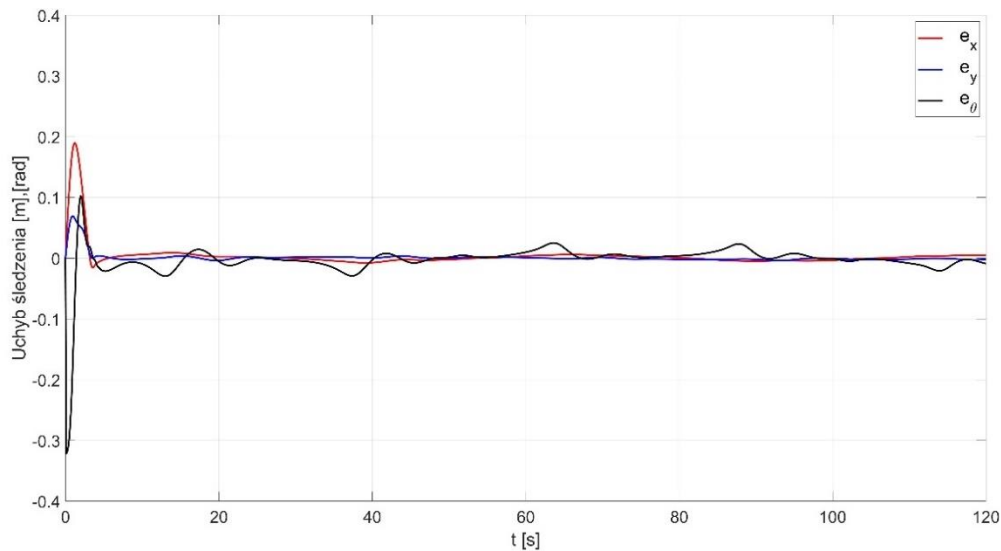
- Śledzenie trajektorii „ósemka”



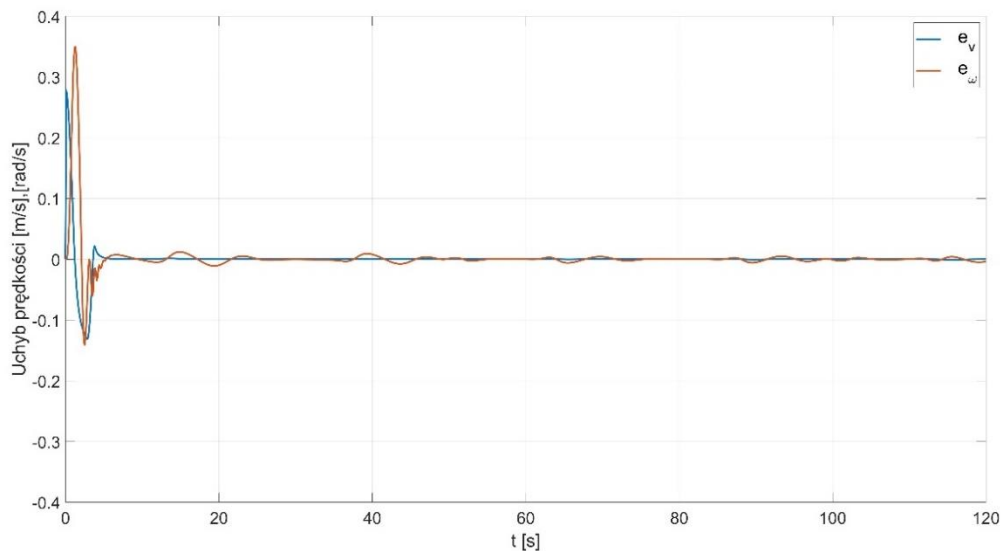
Rys. 5.60. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.61. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.62. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.63. Uchyby prędkości liniowej e_v i kątovej e_w w symulacji śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

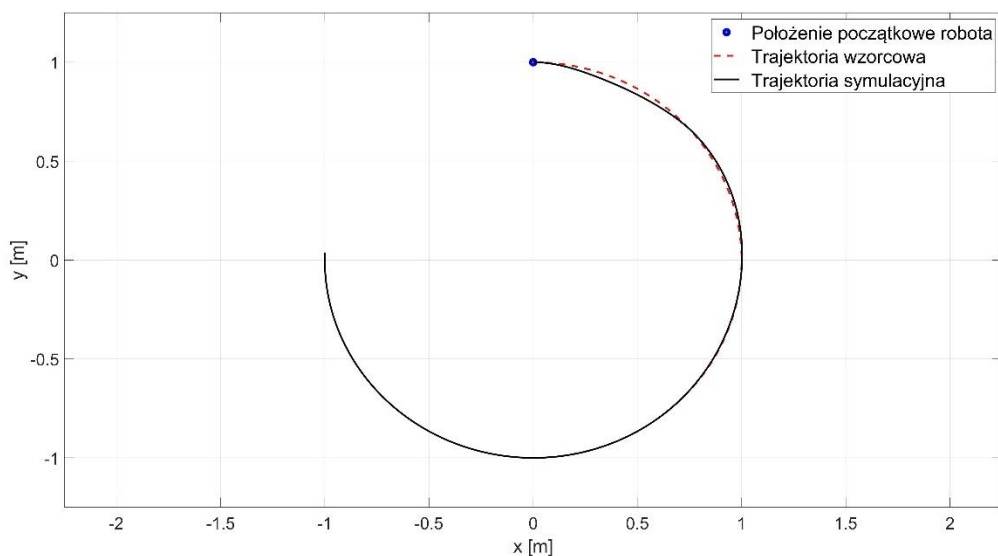
Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 5.62) oraz prędkości (Rys. 5.63) nie większymi niż:

- w kierunku osi x: $e_x = 0.009$ m;
- w kierunku osi y: $e_y = 0.004$ m;
- orientacji: $e_\theta = 0.029$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątovej $e_w = 0.011 \frac{\text{rad}}{\text{s}}$.

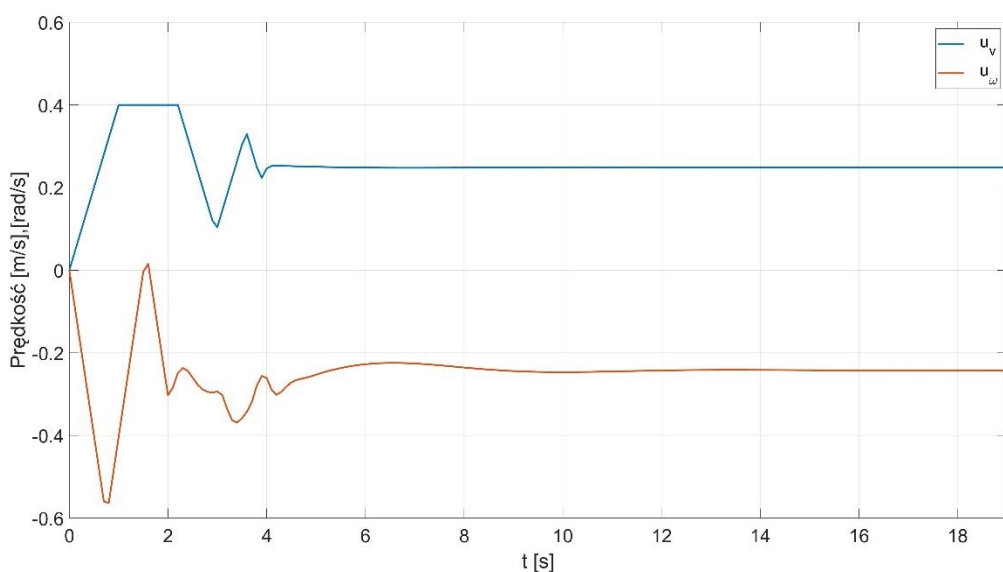
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 2.7$ s.

5.2.3.2. Kryterium ruchu po skończonej drodze dla kontrolera kinematyczno-dynamicznego z adaptacją

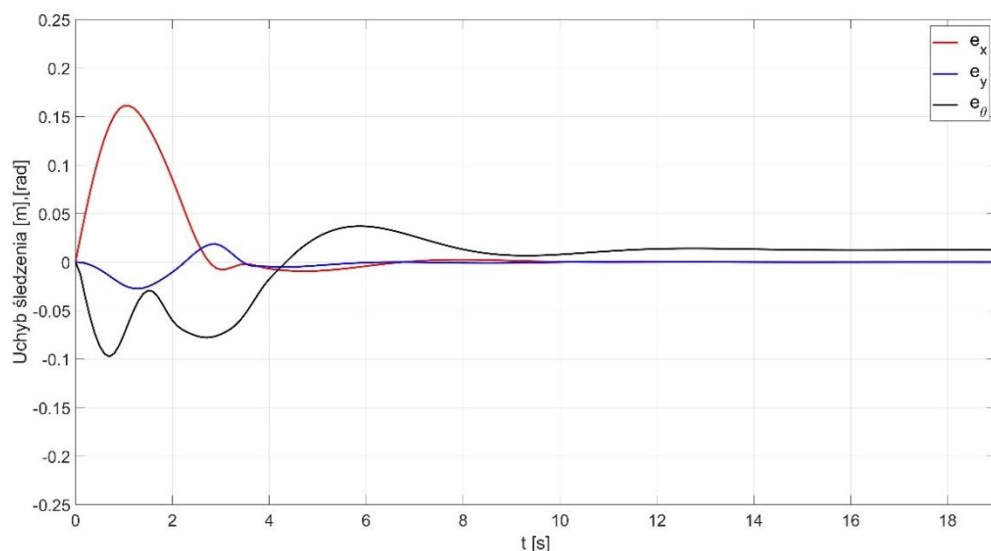
- Śledzenie trajektorii „okrąg”



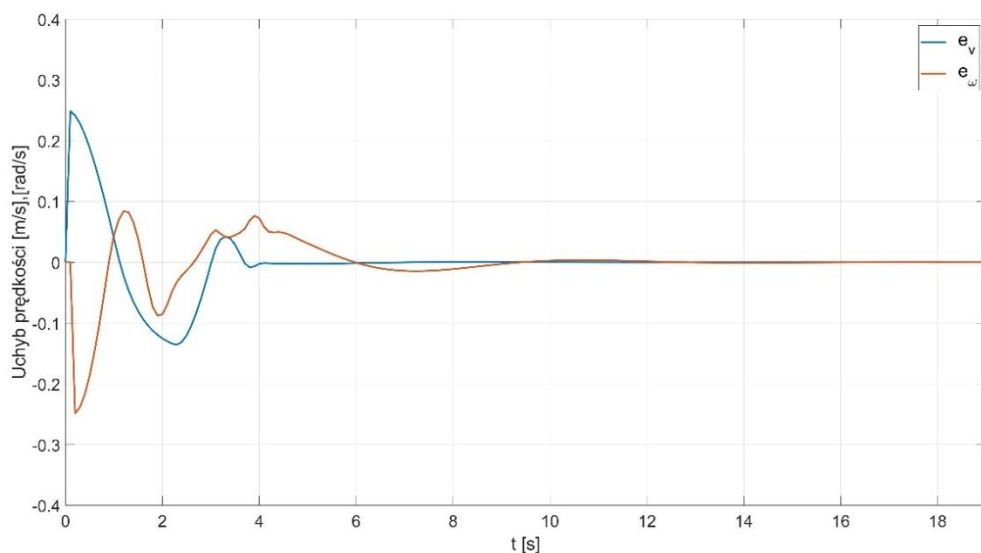
Rys. 5.64. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.65. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w symulacji śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.66. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



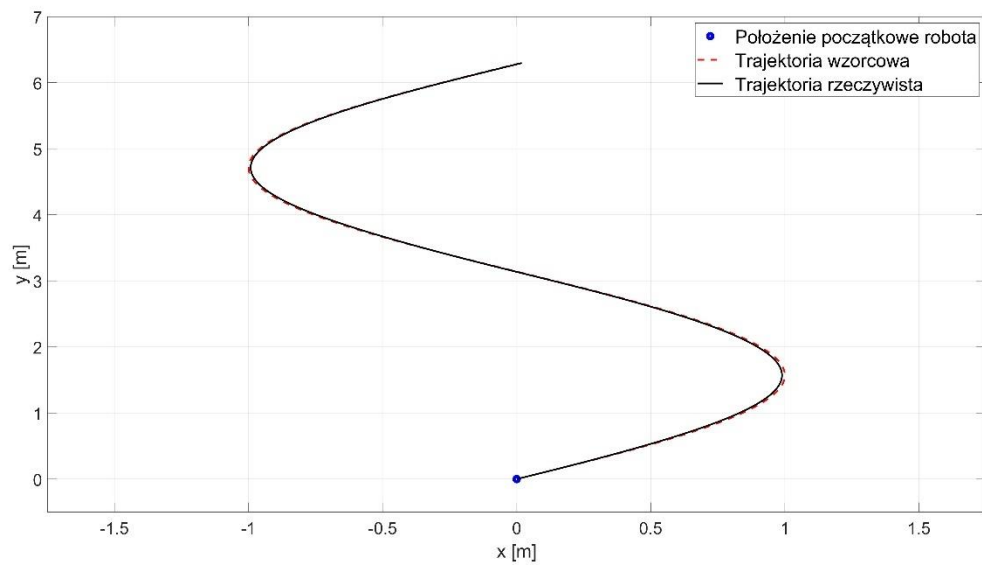
Rys. 5.67. Uchyby prędkości liniowej e_v i kątowej e_w w symulacji śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 5.66) oraz prędkości (Rys. 5.67) nie większymi niż:

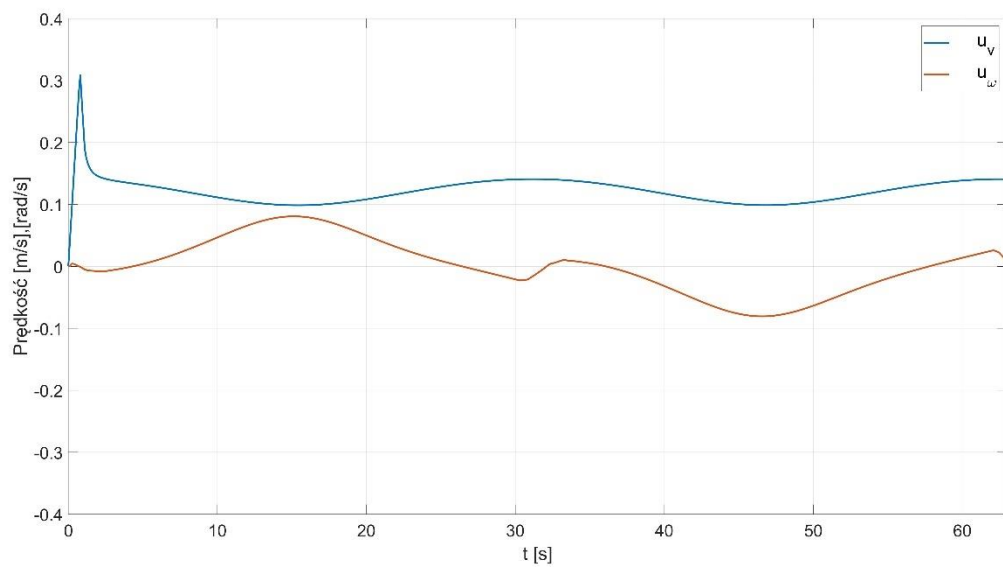
- w kierunku osi x: $e_x = 0.001$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.012$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.001 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 2.6$ s.

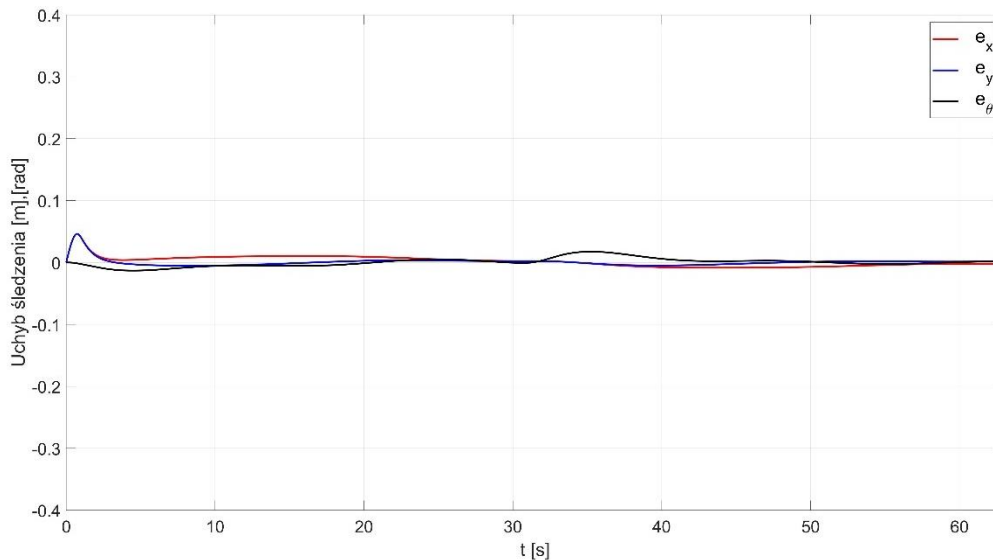
- Śledzenie trajektorii „sinus”



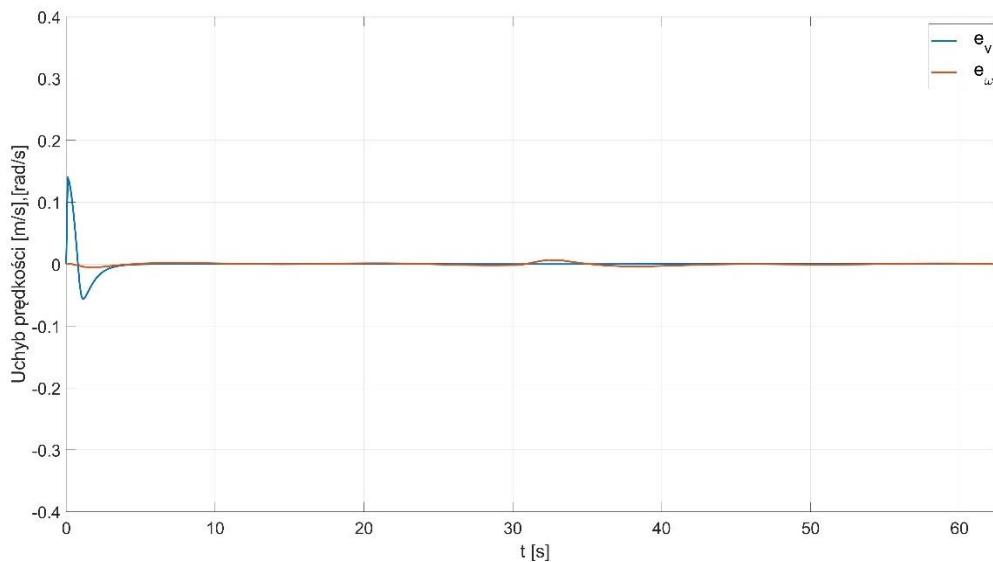
Rys. 5.68. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.69. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.70. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



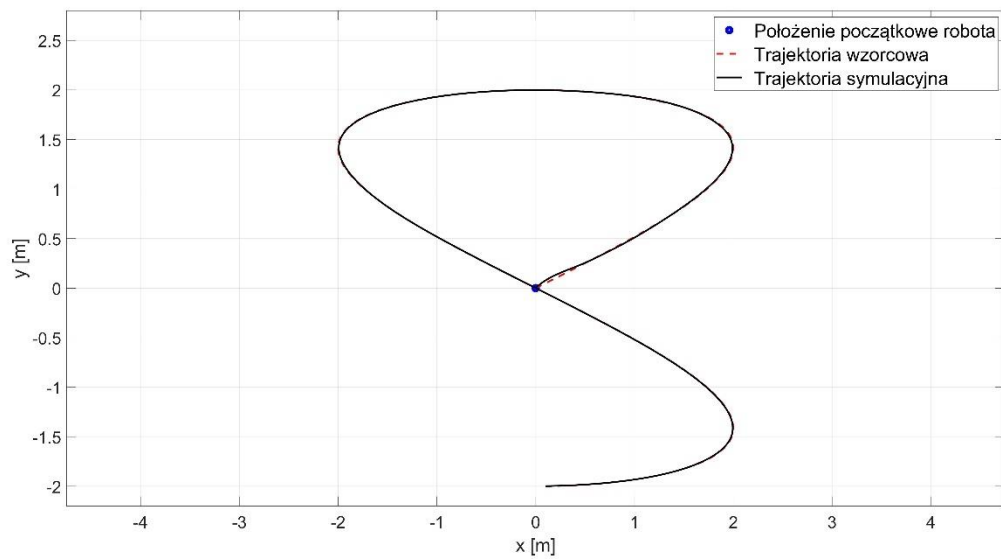
Rys. 5.71. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 5.70) oraz prędkości (Rys. 5.71) nie większymi niż:

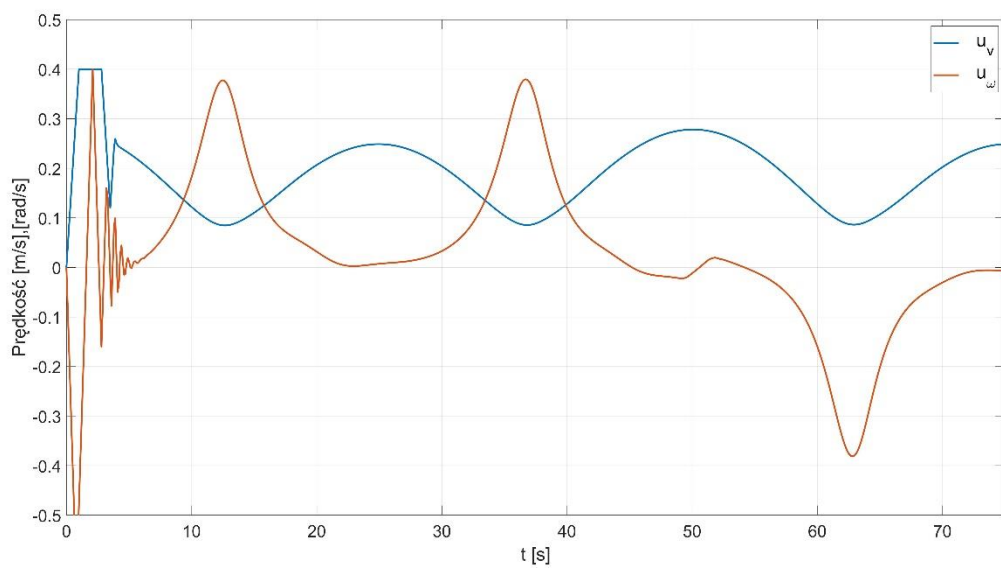
- w kierunku osi x : $e_x = 0.002$ m;
- w kierunku osi y : $e_y = 0.002$ m;
- orientacji: $e_\theta = 0.001$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.002 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 1.4$ s.

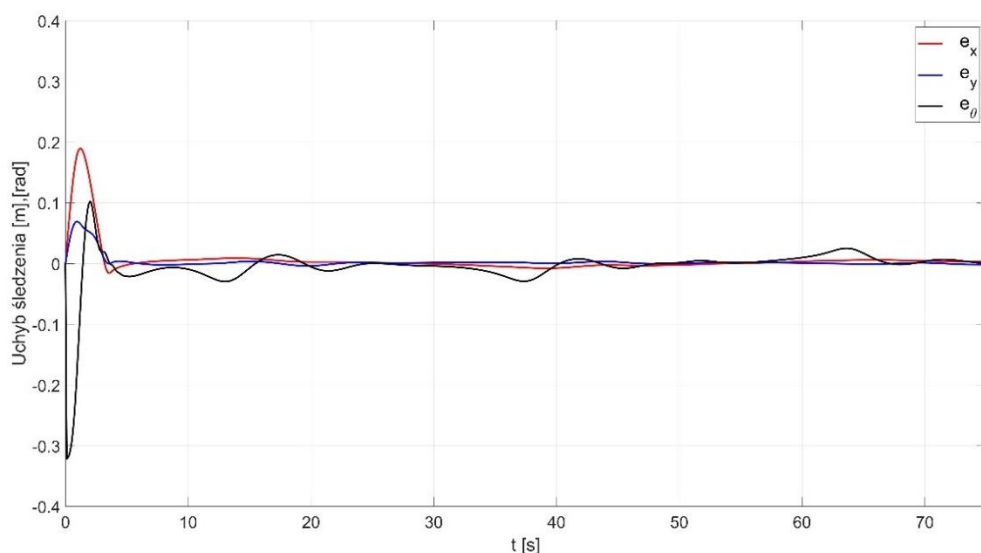
- Śledzenie trajektorii „ósemka”



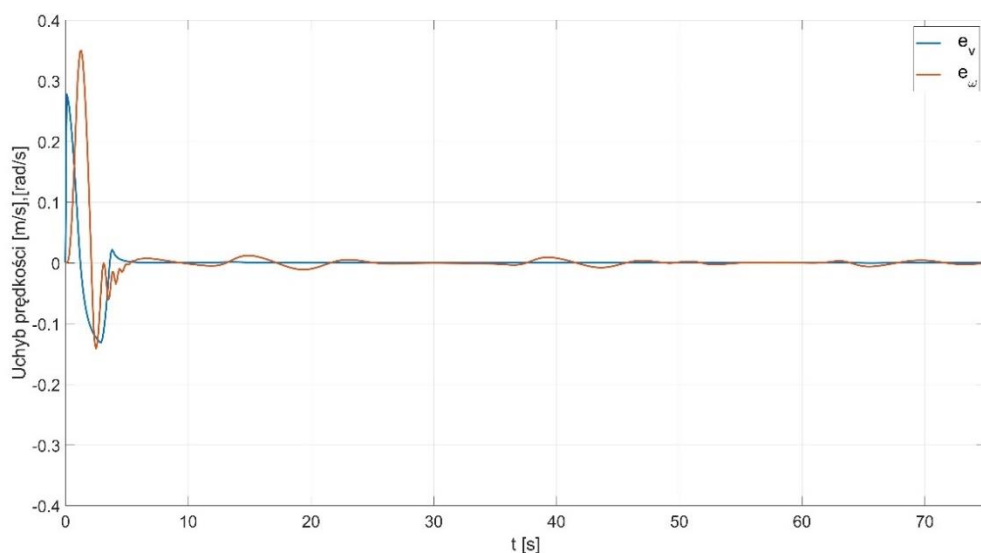
Rys. 5.72. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania symulacyjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.73. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w symulacji śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.74. Uchyby śledzenia e_x , e_y , e_θ wyznaczone w badaniu symulacyjnym dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 5.75. Uchyby prędkości liniowej e_v i kątowej e_ω w symulacji śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych symulacyjnie w badaniu dokładności śledzenia trajektorii, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 5.74) oraz prędkości (Rys. 5.75) nie większymi niż:

- w kierunku osi x: $e_x = 0.003$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.001$ rad;
- prędkości liniowej $e_v = 0.001 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.001 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 2.9$ s.

5.2.4. Podsumowanie badań teoretycznych warstwy wewnętrznej W2 algorytmu sterowania

Analizując uchyby położenia, orientacji i prędkości, wyznaczone w badaniach teoretycznych warstwy wewnętrznej W2 algorytmu sterowania, czyli śledzenia trajektorii przez jednego robota, można stwierdzić, że optymalnym rozwiązaniem pod względem minimalnego uchybu i czasu regulacji jest kontroler kinematyczno-dynamiczny z adaptacją, czyli połączenie kontrolera kinematycznego i dynamicznego z adaptacją. Połączenie to pozwoliło na uzyskanie najmniejszych wartości uchybów i najkrótszego czasu regulacji w stosunku do pozostałych konfiguracji układu sterowania:

- dla trajektorii okrąg, w stosunku do uchybów wyznaczonych dla kontrolera kinematycznego otrzymano wartości mniejsze o: 60% dla e_x , 25% dla e_y , 35% dla e_θ , 27% dla e_v , 67% dla e_ω , 50% dla t_r ; w stosunku do uchybów wyznaczonych dla kontrolera kinematyczno-dynamicznego otrzymano wartości mniejsze o: 56% dla e_x , 25% dla e_y , 41% dla e_θ , 27% dla e_v , 57% dla e_ω , 25% dla t_r ;
- dla trajektorii sinus, w stosunku do uchybów wyznaczonych dla kontrolera kinematycznego otrzymano wartości mniejsze o: 73% dla e_x , 76% dla e_y , 72% dla e_θ , 86% dla e_v , 68% dla e_ω , 59% dla t_r ; w stosunku do uchybów wyznaczonych dla kontrolera kinematyczno-dynamicznego otrzymano wartości mniejsze o: 8% dla e_x , 14% dla e_y , 13% dla e_θ , 0% dla e_v , 0% dla e_ω , 17% dla t_r ;
- dla trajektorii ósemka, w stosunku do uchybów wyznaczonych dla kontrolera kinematycznego otrzymano wartości mniejsze o: 78% dla e_x , 91% dla e_y , 81% dla e_θ , 92% dla e_v , 90% dla e_ω , 44% dla t_r ; w stosunku do uchybów wyznaczonych dla kontrolera kinematyczno-dynamicznego otrzymano wartości mniejsze o: 0% dla e_x , 60% dla e_y , 7% dla e_θ , 0% dla e_v , 15% dla e_ω , 21% dla t_r ;

Wartości uchybów wyznaczonych w badaniach teoretycznych warstwy wewnętrznej W2 algorytmu sterowania zgodnie z kryterium ruchu przez określony czas zestawiono w Tabeli 14. Wartości uchybów wyznaczonych w badaniach teoretycznych warstwy wewnętrznej W2 algorytmu sterowania zgodnie z kryterium ruchu po skończonej drodze zestawiono w Tabeli 15.

Tabela 14. Wartości uchybów wyznaczonych w badaniu teoretycznym warstwy wewnętrznej W2 zgodnie z kryterium ruchu przez określony czas

		Kontroler kinematyczny	Kontroler kinematyczno-dynamiczny	Kontroler kinematyczno-dynamiczny z adaptacją
Trajektoria „okrąg”	e_x [m]	0.010	0.009	0.004
	e_y [m]	0.008	0.008	0.006
	e_θ [rad]	0.057	0.063	0.037
	e_v $\left[\frac{m}{s}\right]$	0.011	0.008	0.008
	e_ω $\left[\frac{rad}{s}\right]$	0.042	0.033	0.014
	t_r [s]	4.8	3.2	2.4
Trajektoria „sinus”	e_x [m]	0.040	0.012	0.011
	e_y [m]	0.025	0.007	0.006
	e_θ [rad]	0.074	0.024	0.021
	e_v $\left[\frac{m}{s}\right]$	0.007	0.001	0.001
	e_ω $\left[\frac{rad}{s}\right]$	0.022	0.007	0.007
	t_r [s]	4.6	2.3	1.9
Trajektoria „ósemka”	e_x [m]	0.041	0.009	0.009
	e_y [m]	0.047	0.010	0.004
	e_θ [rad]	0.152	0.031	0.029
	e_v $\left[\frac{m}{s}\right]$	0.012	0.001	0.001
	e_ω $\left[\frac{rad}{s}\right]$	0.111	0.013	0.011
	t_r [s]	4.8	3.4	2.7

Tabela 15. Wartości uchybów wyznaczonych w badaniu teoretycznym warstwy wewnętrznej W2 zgodnie z kryterium ruchu po skończonej drodze

		Kontroler kinematyczny	Kontroler kinematyczno-dynamiczny	Kontroler kinematyczno-dynamiczny z adaptacją
Trajektoria „okrąg”	e_x [m]	0.003	0.001	0.001
	e_y [m]	0.001	0.001	0.001
	e_θ [rad]	0.019	0.012	0.012
	e_v $\left[\frac{m}{s}\right]$	0.001	0.001	0.001
	e_ω $\left[\frac{rad}{s}\right]$	0.012	0.001	0.001
	t_r [s]	3.9	3.1	2.6
Trajektoria „sinus”	e_x [m]	0.010	0.004	0.002
	e_y [m]	0.009	0.004	0.002
	e_θ [rad]	0.012	0.007	0.001
	e_v $\left[\frac{m}{s}\right]$	0.001	0.001	0.001
	e_ω $\left[\frac{rad}{s}\right]$	0.022	0.004	0.002
	t_r [s]	1.9	1.4	1.4
Trajektoria „ósemka”	e_x [m]	0.001	0.001	0.003
	e_y [m]	0.014	0.002	0.001
	e_θ [rad]	0.058	0.006	0.001
	e_v $\left[\frac{m}{s}\right]$	0.004	0.001	0.001
	e_ω $\left[\frac{rad}{s}\right]$	0.062	0.002	0.001
	t_r [s]	3.6	3.1	2.9

Wyznaczone dla kontrolera kinematyczno-dynamicznego z adaptacją uchyby i czas regulacji są na akceptowalnym poziomie (biorąc pod uwagę konstrukcję zastosowanego robota i odnosząc się do wskazanych w rozdziale 1.2 stosowanych rozwiązań kontrolerów), który pozwala na zastosowanie tego zestawienia w dalszych badaniach.

5.3. Zakres i metodyka badań teoretycznych warstwy zewnętrznej W1 algorytmu sterowania

Badania warstwy zewnętrznej W1 systemu sterowania przedstawionego na Rys. 4.1 przeprowadzono w celu sprawdzenia odporności algorytmu sterowania na opóźnienia czasowe sygnałów sterujących i zakłócenia wynikające z niesprawności robotów wchodzących w skład grupy. W badaniach warstwy zewnętrznej W1 zastosowano konfigurację układu sterowania warstwy W2 przedstawioną na Rys. 5.3, obejmującą kontroler kinematyczno-dynamiczny z adaptacją, opisany równaniami (4.18), (4.29) i (2.22), która wykazała najlepsze rezultaty sterowania w badaniach przedstawionych w punkcie 5.2 oraz modele matematyczne robotów wyrażone równaniami (2.9) i (2.17).

W badaniach rozpatrzono dwa przypadki (warianty):

- synchronizacja rozpoczęcia ruchu przez grupę,
- synchroniczny ruch grupy po zadanej trajektorii.

Odporność algorytmu sterowania na opóźnienia czasowe zbadano na podstawie synchronizacji rozpoczęcia ruchu grupy trzech robotów po linii prostej. W badaniu tym, grupa robotów miała przejechać drogę równą 1m z prędkością liniową równą $v_d = 0.1 \frac{m}{s}$. Na sygnał rozpoczęcia ruchu grupy nałożono opóźnienie czasowe z zakresu ustalonego w rozdziale 4. Wyniki przedstawiono dla minimalnych wartości wyznaczonego w badaniach rozpoznawczych zakresu opóźnień, odpowiednio $\Delta t_2 = 300$ ms dla Robota 2 i $\Delta t_3 = 500$ ms dla Robota 3 (w celu zobrazowania różnicy w położeniu). W pierwszej kolejności przeprowadzono badanie bez uwzględnienia warstwy W2, czyli kontrolera grupy opisanego w rozdziale 4.3. Następnie to samo badanie powtórzono dla kompletnej konfiguracji algorytmu sterowania obejmującej wszystkie elementy przedstawione na Rys. 4.1.

Odporność algorytmu sterowania na zakłócenia wynikające z niesprawności robotów wchodzących w skład grupy sprawdzono w badaniu synchronicznego ruchu po trajektorii kołowej. W tym przypadku zadaniem dla grupy był przejazd drogi równej $\frac{3}{4}$ obwodu okręgu o promieniu 2 m, z zadeklarowaną prędkością liniową $v_d = 0.1 \frac{m}{s}$. W 60-tej sekundzie ruchu wprowadzono zakłócenie w postaci nagłego zmniejszenia prędkości Robota 2 symulujące np. opory lub uszkodzenie układu napędowego, lub nagły spadek poziomu naładowania akumulatorów. W przedstawionych wynikach badań uwzględniono zmniejszenie prędkości Robota 2 o 5%. W pierwszej kolejności przeprowadzono badanie bez uwzględnienia warstwy

W2, czyli kontrolera konfiguracji grupy. Następnie to samo badanie powtórzono dla kompletnej konfiguracji algorytmu sterowania obejmującej wszystkie elementy przedstawione na Rys. 4.1.

5.4. Badania symulacyjne warstwy zewnętrznej W1 algorytmu sterowania

Badania symulacyjne przeprowadzono w środowisku MATLAB/Simulink. Formację grupy ustalono jako linię prostą. Roboty oddalono od siebie o odległość 0.5 m, a ich położenia początkowe opisano wektorami współrzędnych konfiguracyjnych w postaci:

- Robot 1 - $h_1(t_0) = [0, 0, 0, 0, 0]$;
- Robot 2 - $h_2(t_0) = [0, 0.5, 0, 0, 0]$;
- Robot 3 - $h_3(t_0) = [0, -0.5, 0, 0, 0]$;

Położenie środka geometrycznego wirtualnej struktury C_{vc} ustalono w środku geometrycznym Robotu 1. Orientację θ^{vc} układu współrzędnych wirtualnej struktury również ustalono zgodną z orientacją Robotu 1.

Wyniki badań przedstawiono w formie wykresów:

- porównania trajektorii wzorcowej i otrzymanej w wyniku symulacji,
- uchybów śledzenia trajektorii wyznaczonych z zależności:

$$eR_i = \sqrt{(x_{di} - x_{ri})^2 + (y_{di} - y_{ri})^2} \quad (5.5)$$

- uchybów odległości pomiędzy robotami wyznaczonych z zależności:

$$e_{dij} = L_{ij} + \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5.6)$$

- wykresów prędkości symulacyjnej.

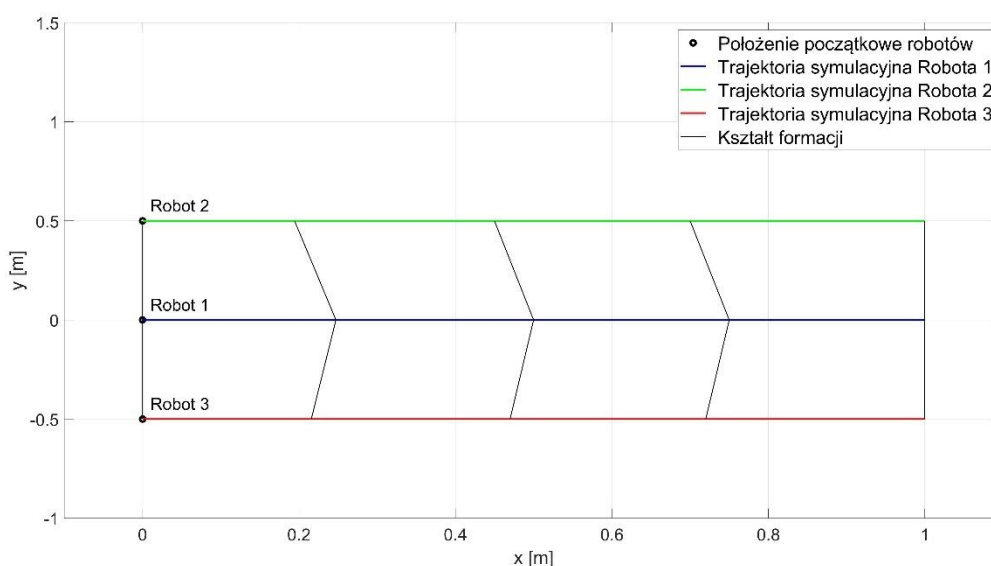
Jakość zaproponowanych układów regulacji oceniono na podstawie wartości uchybów opisanych wzorami (6.5) i (6.6).

5.4.1. Badania symulacyjne synchronizacji rozpoczęcia ruchu przez grupę robotów

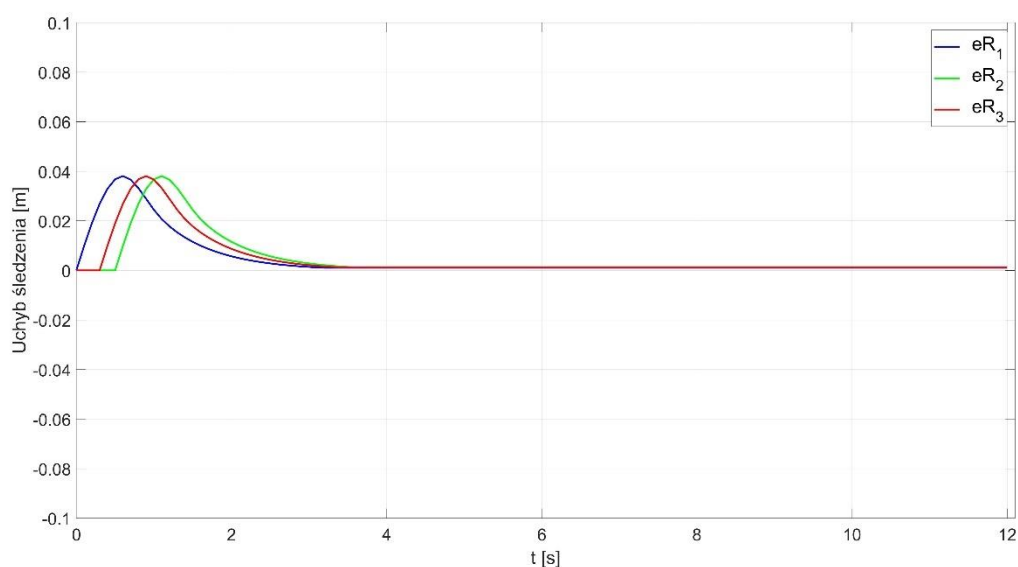
Wyniki badań symulacyjnych rozpoczęcia ruchu przez grupę robotów przedstawiono na Rys. 5.76 - Rys. 5.83. Badania przeprowadzono zgodnie z metodyką opisaną w rozdziale 5.3 i schematem przedstawionym na Rys. 4.1.

5.4.1.1. Badanie symulacyjne synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy

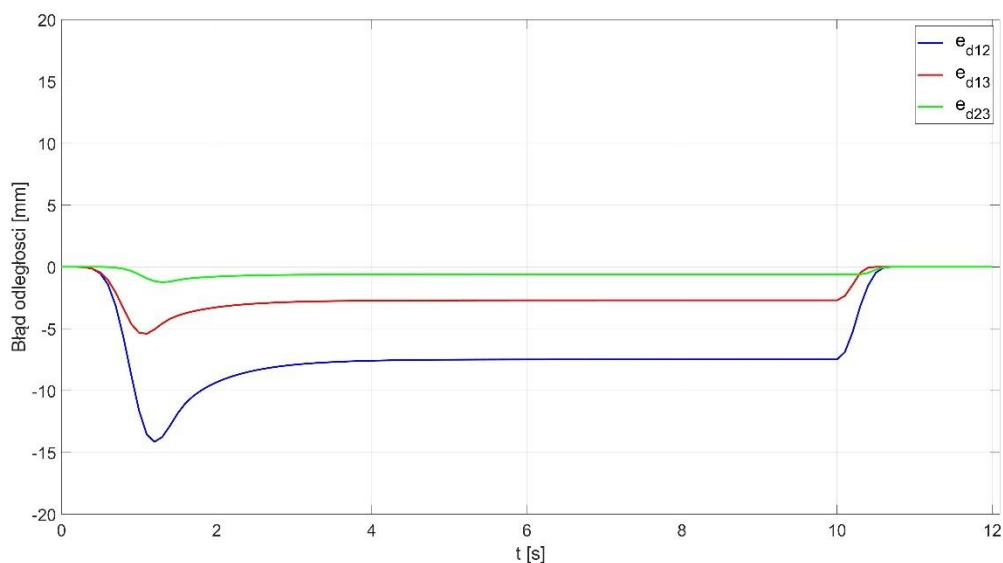
Badanie przeprowadzono dla grupy trzech robotów. Na sygnały sterujące dwóch robotów nałożono opóźnienie czasowe: $\Delta t_2 = 300$ ms dla Robota 2 i $\Delta t_3 = 500$ ms dla Robota 3. Wyniki badania przedstawiono na Rys. 5.76 - Rys. 5.79.



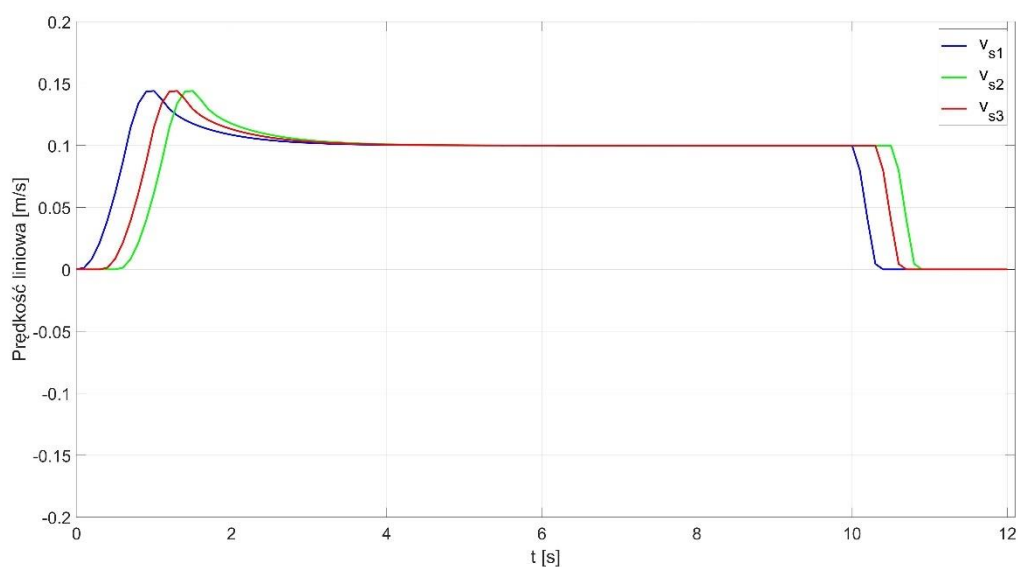
Rys. 5.76. Trajektorie symulacyjne robotów w badaniu synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy



Rys. 5.77. Uchyby śledzenia trajektorii wzorcowych w badaniu synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy



Rys. 5.78. Błędy odległości pomiędzy robotami w badaniu synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy



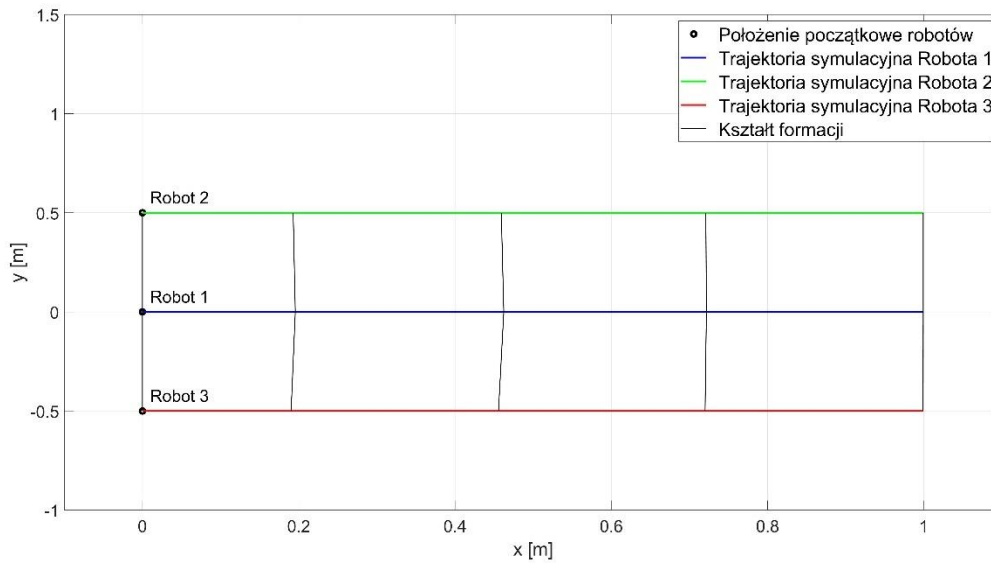
Rys. 5.79. Prędkości liniowe wyznaczone symulacyjnie w badaniu synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy

W badaniu symulacyjnym synchronizacji rozpoczęcia ruchu przez grupę robotów bez uwzględnienia warstwy sterowania W1 roboty przejechały zadaną drogę równą 1m jednak konfiguracja grupy podczas ruchu nie była utrzymywana co jest widoczne na Rys. 5.76. Analizując wykres prędkości (Rys. 5.79) zauważono, że roboty poruszały się z zadeklarowaną prędkością, jednak w wyniku opóźnionego startu dwóch robotów różny był też czas ich zatrzymania. Analizując wykresy uchybów śledzenia (Rys. 5.77) i uchybów odległości (Rys. 5.78) można zauważyć, że w tym przypadku priorytetem jest minimalizacja uchybów śledzenia,

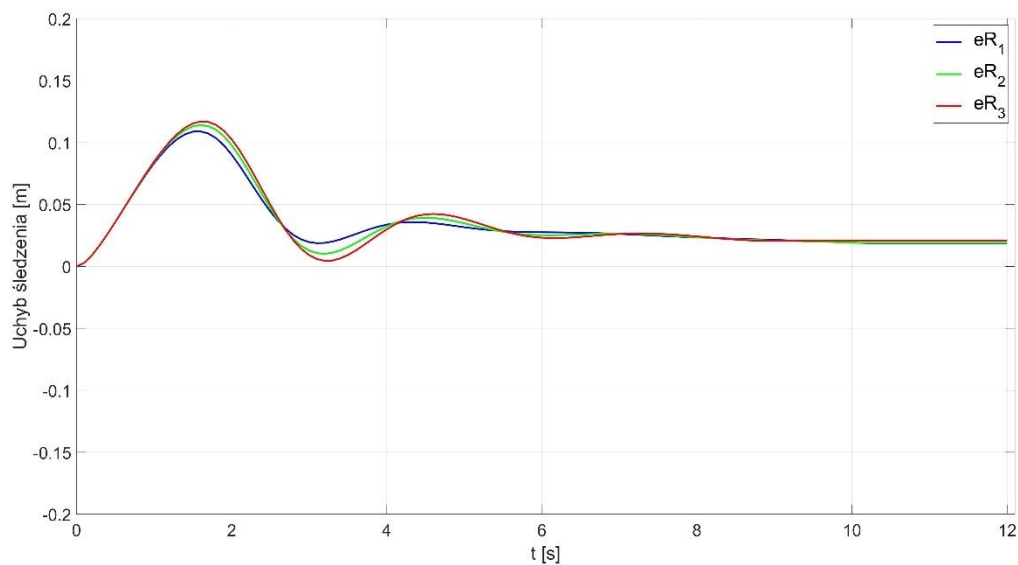
których wartość była nie większa niż $e_{R_i} = 0.001$ m, kosztem dużych uchybów odległości o wartościach nie większych niż: $e_{d12} = 14.2$ mm, $e_{d13} = 5.4$ mm, $e_{d23} = 1.3$ mm, które nie były minimalizowane.

5.4.1.2. Badanie symulacyjne synchronizacji rozpoczęcia ruchu z uwzględnieniem kontrolera konfiguracji grupy

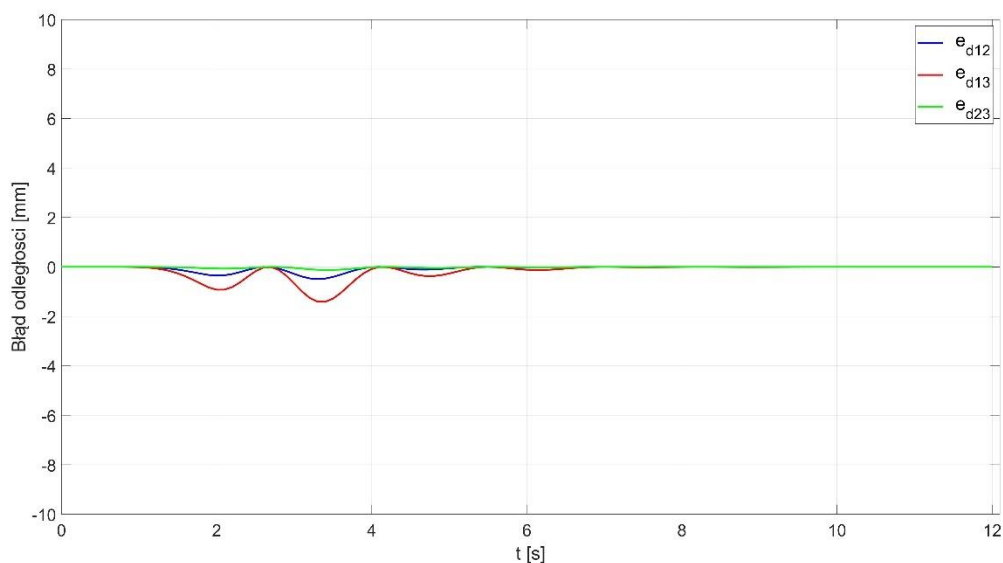
Badanie przeprowadzono dla grupy trzech robotów. Na sygnały sterujące dwóch robotów nałożono opóźnienie czasowe: $\Delta t_2 = 300$ ms dla Robotu 2 i $\Delta t_3 = 500$ ms dla Robotu 3. Wyniki badania przedstawiono na Rys. 5.80 - Rys. 5.83.



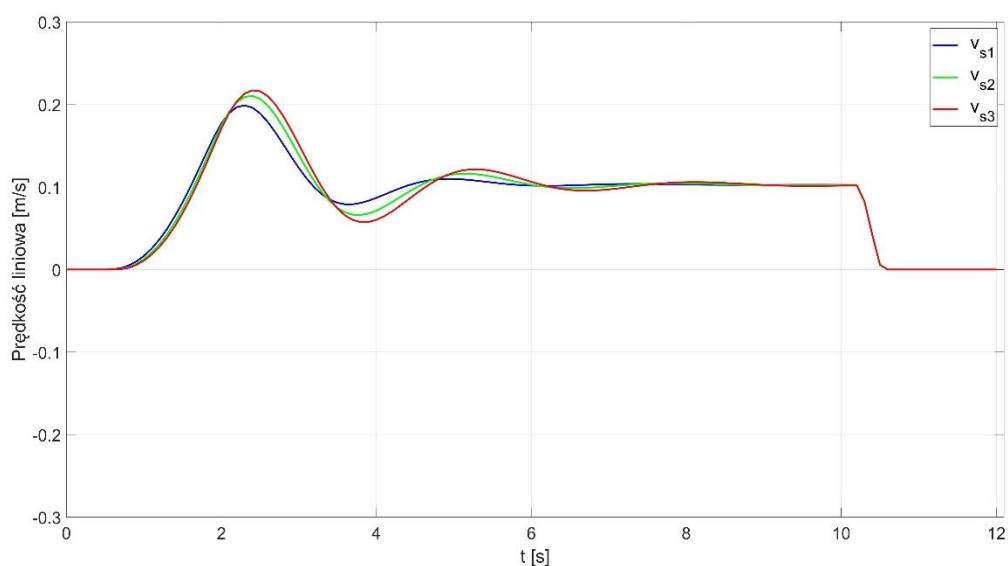
Rys. 5.80. Trajektorie symulacyjne robotów w badaniu synchronizacji rozpoczęcia ruchu z uwzględnieniem kontrolera konfiguracji grupy



Rys. 5.81. Uchyby śledzenia trajektorii wzorcowych w badaniu synchronizacji rozpoczęcia ruchu z uwzględnieniem kontrolera konfiguracji grupy



Rys. 5.82. Błędy odległości pomiędzy robotami w badaniu synchronizacji rozpoczęcia ruchu z uwzględnieniem kontrolera konfiguracji grupy



Rys. 5.83. Prędkości liniowe wyznaczone symulacyjnie w badaniu synchronizacji rozpoczęcia ruchu z uwzględnieniem kontrolera konfiguracji grupy

W badaniu symulacyjnym synchronizacji rozpoczęcia ruchu przez grupę robotów z uwzględnieniem warstwy sterowania W1 roboty przejechały zadaną drogę równą 1m. Pomimo początkowego zaburzenia konfiguracji grupy warstwa sterowania W1 minimalizowała błędy odległości pomiędzy robotami i podczas ruchu zadana konfiguracja była utrzymywana, co jest widoczne na Rys. 5.80. Analizując wykres prędkości (Rys. 5.83) zauważono, że roboty poruszały się z zadeklarowaną prędkością oraz że nastąpił synchroniczny start i zatrzymanie się grupy. Dodatkowo widoczne fluktuacje przebiegów prędkości wskazują na działanie

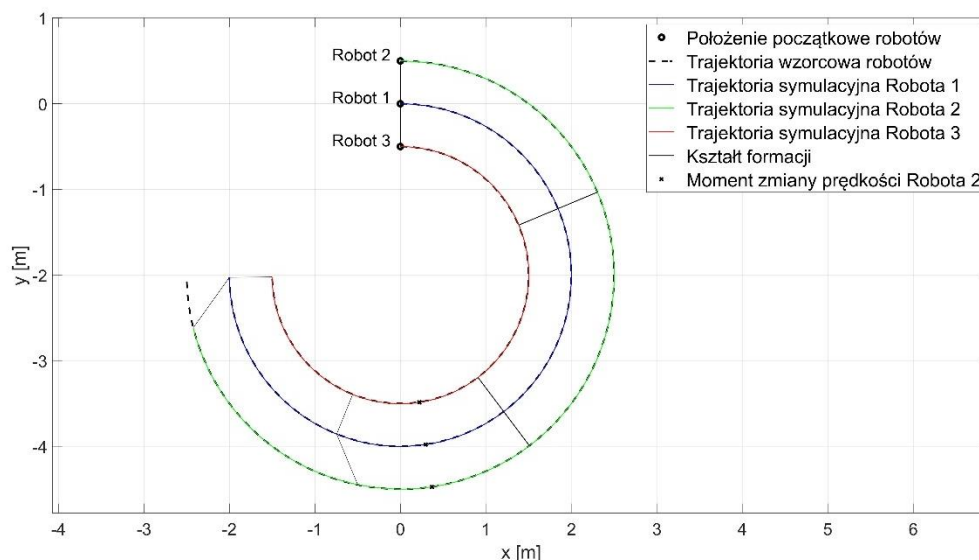
kontrolera grupy minimalizujące uchyb odległości pomiędzy robotami. Analizując wykresy uchybów śledzenia (Rys. 5.77) i uchybów odległości (Rys. 5.78) można zauważyć, że w tym przypadku priorytetem jest minimalizacja uchybów odległości pomiędzy robotami, których wartości były nie większe niż: $e_{d12} = 0.25$ mm, $e_{d13} = 0.7$ mm, $e_{d23} = 1.3$ mm kosztem dużych uchybów odległości o wartościach nie większych niż: $eR_1 = 0.11$ m, $eR_2 = 0.12$ m, $eR_3 = 0.13$ m, które nie były minimalizowane.

5.4.2. Badania symulacyjne synchronicznego ruchu grupy po zadanej trajektorii

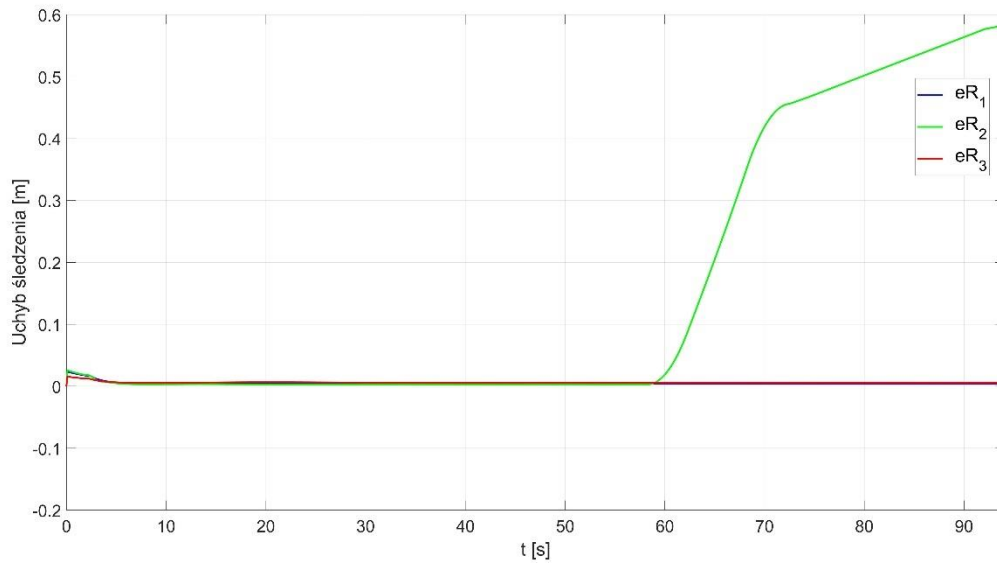
Wyniki badań symulacyjnych synchronicznego ruchu grupy po zadanej trajektorii przedstawiono na Rys. 5.84 - Rys. 5.91. Badania przeprowadzono zgodnie z metodyką opisaną w rozdziale 5.3 i schematem przedstawionym na Rys. 4.1.

5.4.2.1. Badanie symulacyjne synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy

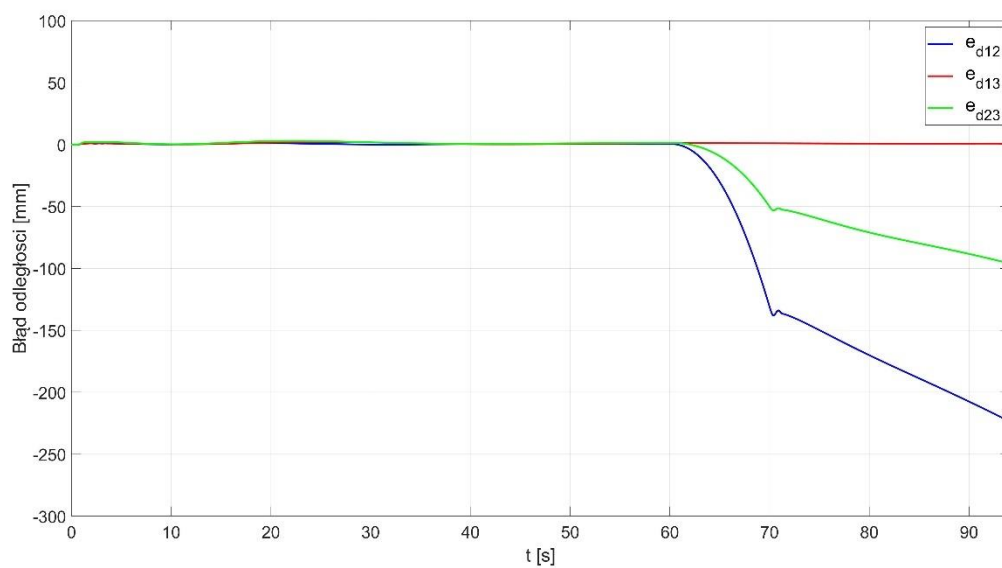
Badanie przeprowadzono dla grupy trzech robotów, której zadaniem był ruch po okręgu o promieniu 2 m i przejazd drogi równej $\frac{3}{4}$ obwodu. W 60-tej sekundzie ruchu wprowadzono zakłócenie w postaci nagłego zmniejszenia prędkości Robota 2 o 5%. Wyniki badania przedstawiono na Rys. 5.84 - Rys. 5.87.



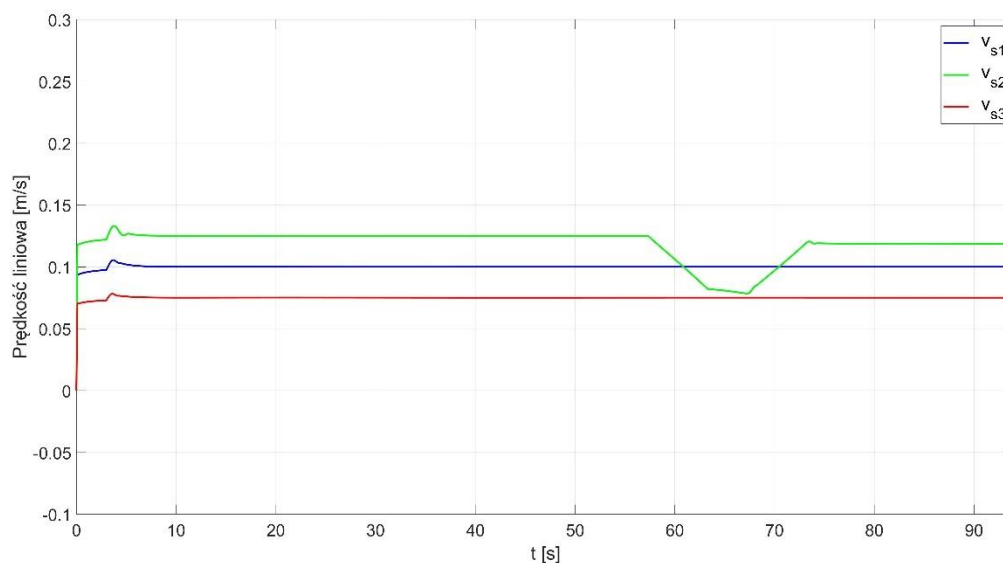
Rys. 5.84. Trajektorie symulacyjne robotów w badaniu synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy



Rys. 5.85. Uchyby śledzenia trajektorii wzorcowych w badaniu synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy



Rys. 5.86. Błędy odległości pomiędzy robotami w badaniu w badaniu synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy

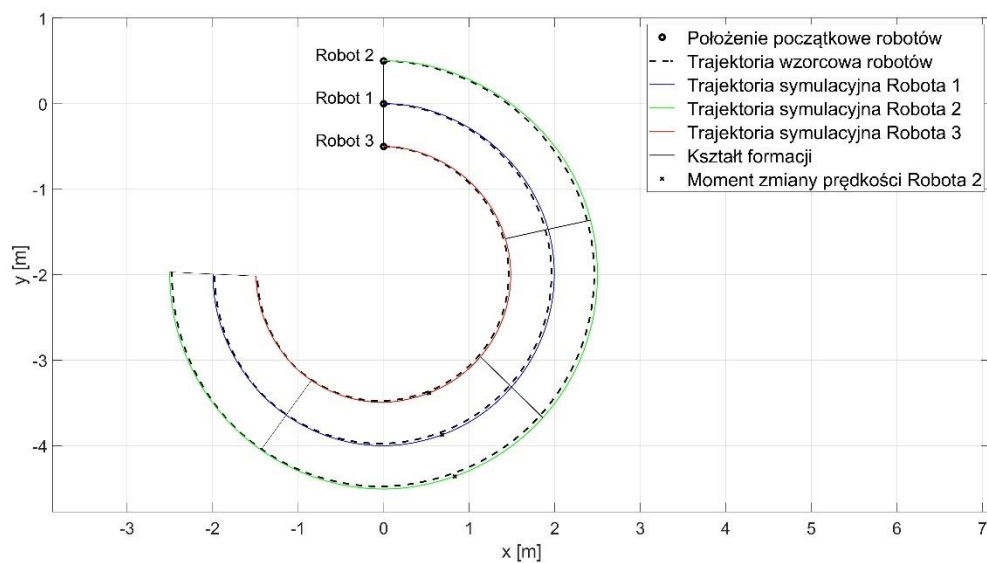


Rys. 5.87. Prędkości liniowe robotów wyznaczone symulacyjnie w badaniu synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy

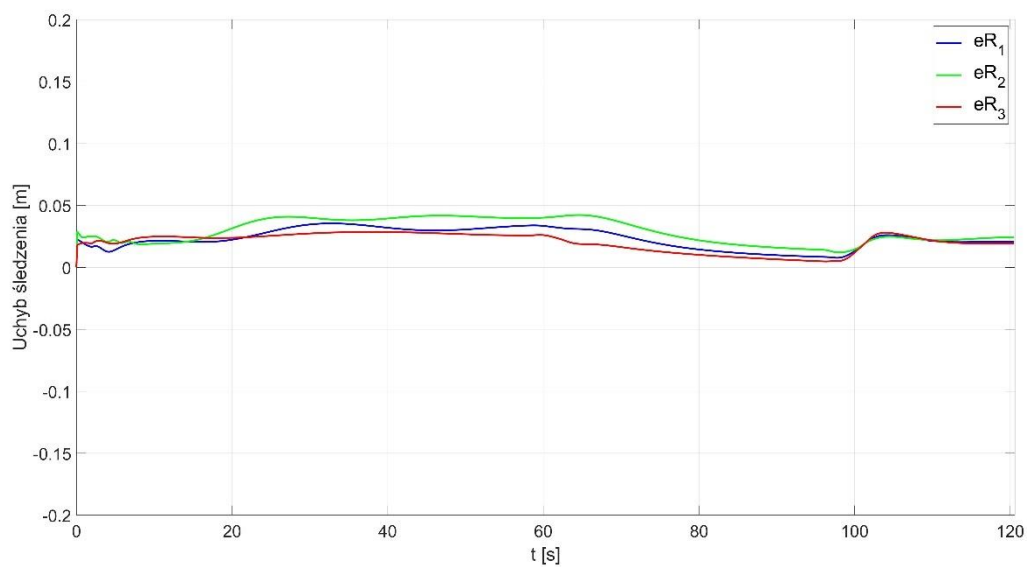
W badaniu symulacyjnym synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia warstwy sterowania W1, roboty poruszały się po zadanej trajektorii, jednak w wyniku niewielkiej zmiany prędkości Robota 2, konfiguracja grupy podczas ruchu nie była utrzymywana co jest widoczne na Rys. 5.84. Analizując wykres prędkości (Rys. 5.87) zauważono, że roboty poruszały się z zadeklarowaną prędkością, jednak w przypadku zmiany prędkości Robota 2 pozostałe roboty nie zareagowały i poruszały się dalej z zadeklarowaną prędkością, deformując ustalony szyk grupy. Analizując wykresy uchybów śledzenia (Rys. 5.85) i uchybów odległości (Rys. 5.86) można zauważyć, że w wyniku zmniejszenia o 5% prędkości Robota 2 pojawiły się duże uchyby odległości pomiędzy robotami, które ostatecznie osiągnęły maksymalne wartości: $e_{d12} = 55.8 \text{ mm}$, $e_{d13} = 1 \text{ mm}$, $e_{d23} = 23.9 \text{ mm}$ oraz duży uchyb śledzenia trajektorii zadanej przez Robota 2 o maksymalnej wartości $e_{R_2} = 0.58 \text{ m}$.

5.4.2.2. Badanie synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy

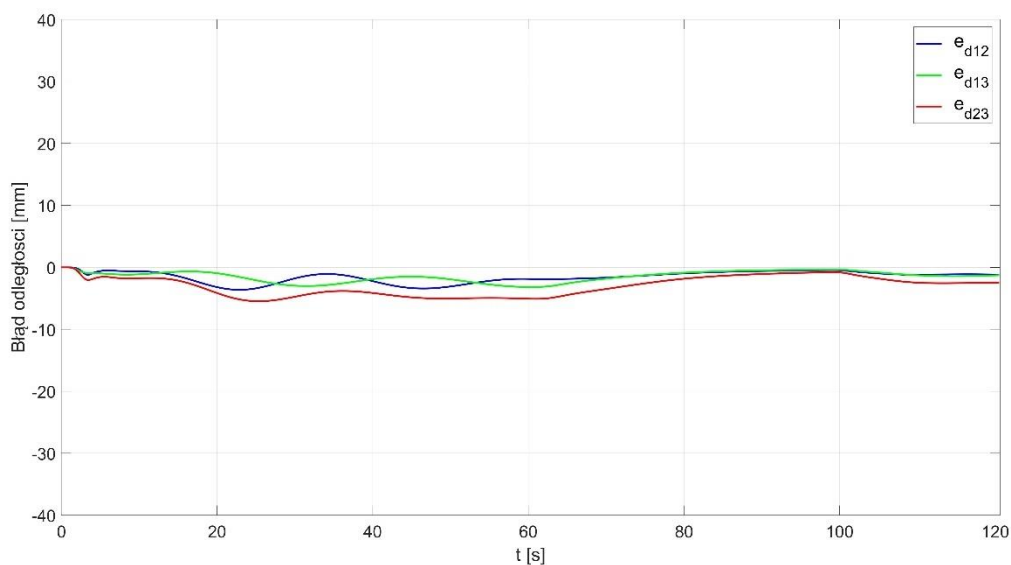
Badanie przeprowadzono dla grupy trzech robotów, której zadaniem był ruch po okręgu o promieniu 2 m i przejazd drogi równej $\frac{3}{4}$ obwodu. W 60-tej sekundzie ruchu wprowadzono zakłócenie w postaci nagłego zmniejszenia prędkości Robota 2 o 5%. Wyniki badania przedstawiono na Rys. 5.88 - Rys. 5.91.



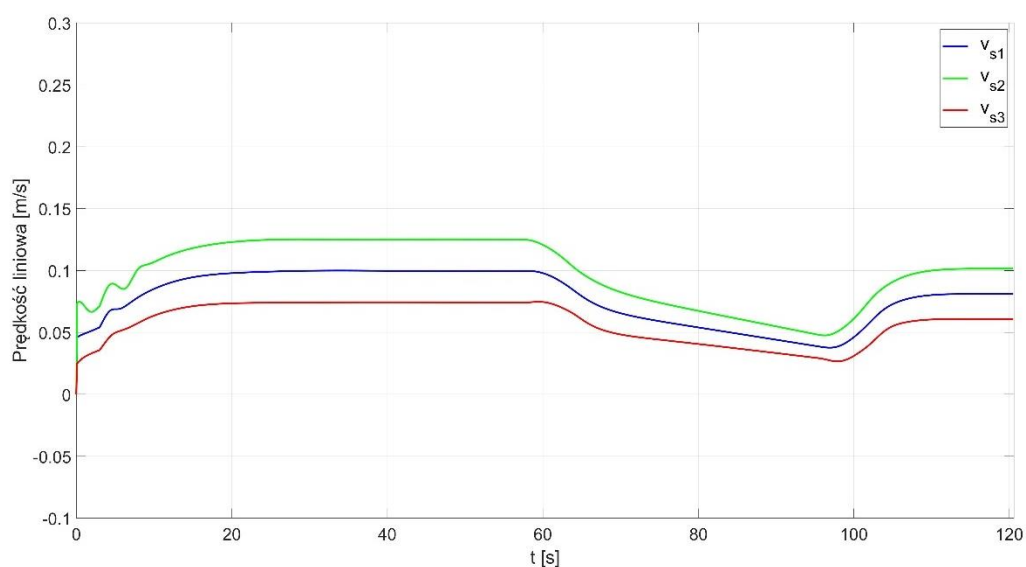
Rys. 5.88. Trajektorie symulacyjne robotów w badaniu synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy



Rys. 5.89. Uchyby śledzenia trajektorii wzorcowych w badaniu synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy



Rys. 5.90. Błędy odległości pomiędzy robotami w badaniu w badaniu synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy



Rys. 5.91. Prędkości liniowe robotów wyznaczone symulacyjnie w badaniu synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy

W badaniu symulacyjnym synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy, roboty poruszały się po zadanej trajektorii i pomimo niewielkiej zmiany prędkości Robotu 2, konfiguracja grupy podczas ruchu była utrzymywana co jest widoczne na Rys. 5.88. Analizując wykres prędkości (Rys. 5.91) zauważono, że roboty poruszały się z zadeklarowaną prędkością, a w przypadku zmiany prędkości Robotu 2 pozostałe roboty zareagowały i również dostosowały swoją prędkość tak aby zachować zdefiniowany szyk. Analizując wykresy uchybów śledzenia (Rys. 5.89) i uchybów odległości (Rys. 5.90) można zauważyć, że w trakcie całego ruchu uchyby

odległości pomiędzy robotami są minimalizowane i utrzymywane, a ich wartości są dopuszczalne i nie większe niż: $e_{d12} = 3.6 \text{ mm}$, $e_{d13} = 3 \text{ mm}$, $e_{d23} = 5.5 \text{ mm}$. Uchyby śledzenia trajektorii przez poszczególne roboty w tym badaniu osiągnęły wartości nie większe niż: $eR_1 = 0.034 \text{ m}$, $eR_2 = 0.042 \text{ m}$, $eR_3 = 0.035 \text{ m}$.

5.4.3. Podsumowanie badań teoretycznych warstwy zewnętrznej W1 algorytmu sterowania

Analizując uchyby położenia robotów i odległości pomiędzy robotami, wyznaczone w badaniach teoretycznych warstwy wewnętrznej W1 algorytmu sterowania, czyli badaniach synchronizacji rozpoczęcia ruchu i synchronicznego ruchu po zadanej trajektorii można stwierdzić, że zaproponowany autorski algorytm sterowania grupą robotów działa prawidłowo.

Wyznaczone w badaniach symulacyjnych wartości uchybów są akceptowalne (na podstawie analizy stosowanych metod sterowania grupami robotów przeprowadzonej w punkcie 0) z punktu widzenia zastosowania jako system transportowy, a ich wartości zestawiono w Tabeli 16.

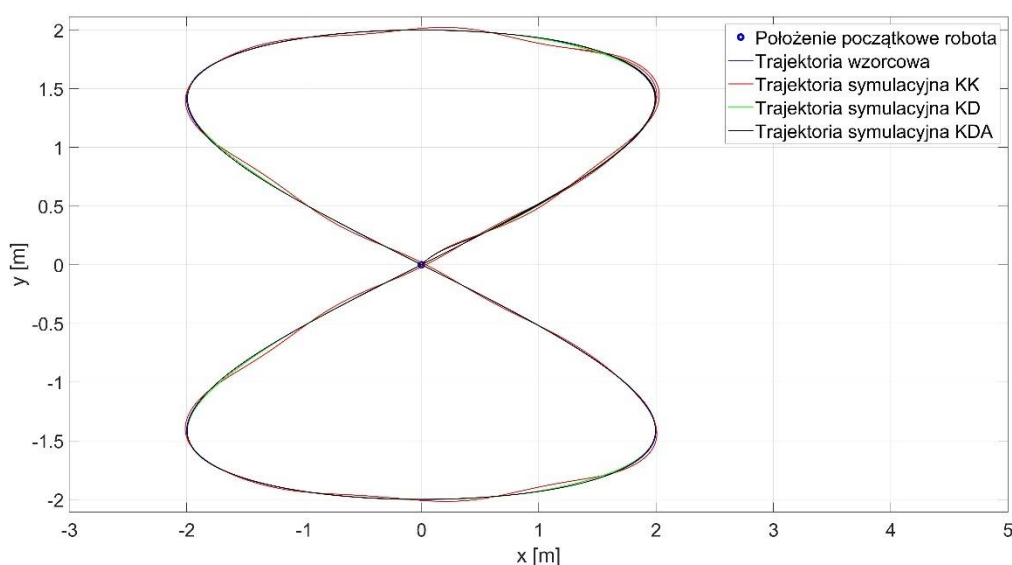
Tabela 16. Wartości uchybów wyznaczonych w badaniach teoretycznych warstwy zewnętrznej W1 algorytmu sterowania.

	Badanie symulacyjne synchronizacji rozpoczęcia ruchu		Badania symulacyjne synchronicznego ruchu grupy po zadanej trajektorii	
	Bez uwzględnienia kontrolera grupy	Z uwzględnieniem kontrolera grupy	Bez uwzględnienia kontrolera grupy	Z uwzględnieniem kontrolera grupy
$eR_1 \text{ [m]}$	0.001	0.11	0.001	0.034
$eR_2 \text{ [m]}$	0.001	0.12	0.58	0.042
$eR_3 \text{ [m]}$	0.001	0.13	0.001	0.035
$e_{d12} \text{ [mm]}$	14.2	0.25	55.8	3.6
$e_{d13} \text{ [mm]}$	5.4	0.7	1	3
$e_{d23} \text{ [mm]}$	1.3	1.3	23.9	5.5

5.5. Podsumowanie badań teoretycznych

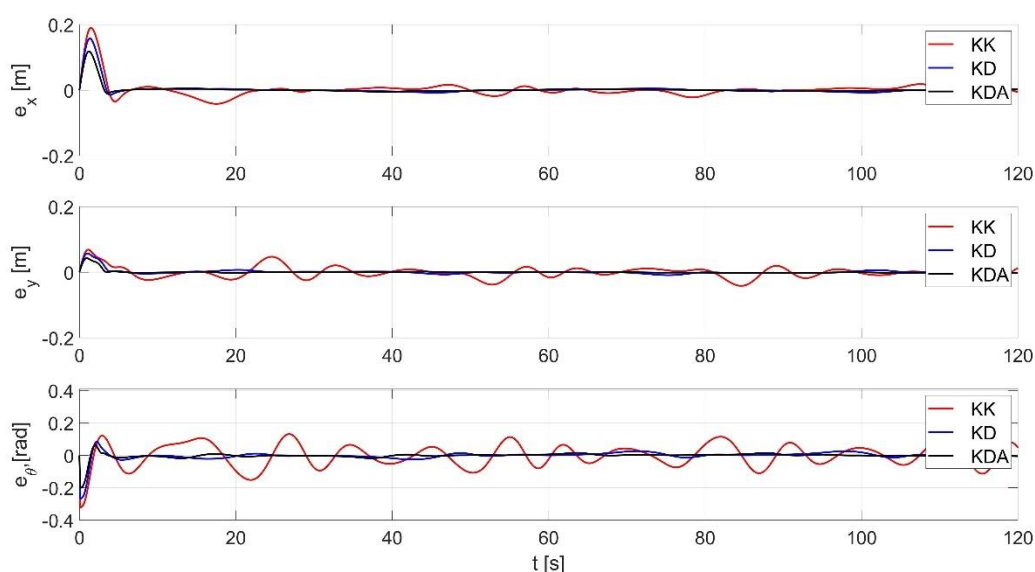
Na podstawie przeprowadzonych badań teoretycznych mających na celu sprawdzenie jakości sterowania warstwy wewnętrznej W2 struktury przedstawionej na Rys. 4.1 można stwierdzić, że zaproponowane kontrolery nakierowują robota na zadaną trajektorię, a jej kształt jest odwzorowany z bardzo dużą dokładnością. Zadanie śledzenia trajektorii jest prawidłowo realizowane dla prostej (w aspekcie zmienności sygnałów sterujących)) trajektorii „okrąg” jak i bardziej skomplikowanych trajektorii „sinus” i „ósemka”. Najmniejsze wartości uchybów

i najkrótszy czas regulacji osiągnięto przy zastosowaniu kontrolera kinematyczno-dynamicznego z adaptacją. Fakt ten potwierdza porównanie trajektorii (Rys. 5.92), uchybów śledzenia (Rys. 5.93) i prędkości (Rys. 5.94), wyznaczonych w badaniach symulacyjnych dla wszystkich trzech konfiguracji układu sterowania. W związku z tym do badań warstwy zewnętrznej W1 algorytmu sterowania zastosowano właśnie tę konfigurację kontrolerów w warstwie W2.

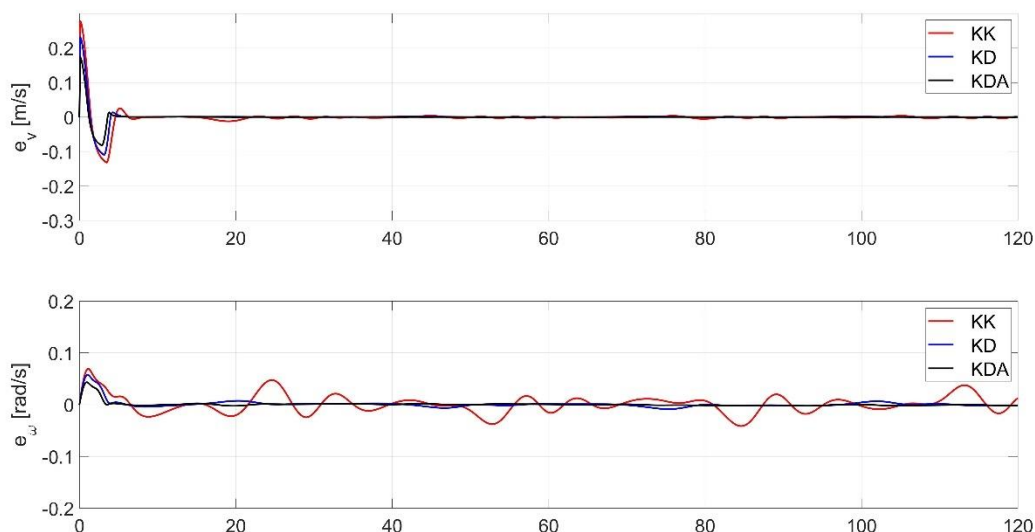


Rys. 5.92. Porównanie trajektorii wyznaczonych w badaniach symulacyjnych trzech konfiguracji układu sterowania warstwy wewnętrznej W2

KK – kontroler kinematyczny, KD – kontroler kinematyczno-dynamiczny, KDA – kontroler kinematyczno-dynamiczny z adaptacją



Rys. 5.93. Porównanie uchybów śledzenia e_x , e_y , e_θ wyznaczonych w badaniach symulacyjnych trzech konfiguracji układu sterowania warstwy wewnętrznej W2



Rys. 5.94. Porównanie uchybów prędkości liniowej e_v i kątowej e_ω wyznaczonych w badaniach symulacyjnych trzech konfiguracji układu sterowania warstwy wewnętrznej W2

Na podstawie przeprowadzonych badań teoretycznych warstwy zewnętrznej W1 algorytmu sterowania, mających na celu sprawdzenie jego odporności na opóźnienia czasowe sygnałów sterujących i zakłócenia wynikające z niesprawności robotów wchodzących w skład grupy, można stwierdzić, że zaproponowany autorski algorytm sterowania realizuje synchroniczny start i ruch grupy po zadanej trajektorii z zachowaniem ustalonego kształtu formacji.

Zastosowanie synchronizacji czasowej sygnałów sterujących pozwoliło na zmniejszenie błędów odległości pomiędzy robotami wywołanych opóźnieniami czasowymi (odpowiednio $\Delta t_2 = 300$ ms dla Robota 2 i $\Delta t_3 = 500$ ms dla Robota 3) oraz zakłóceniem w postaci nagłego zmniejszenia prędkości Robota 2 o 5%. W poszczególnych wariantach badań wartości zmniejszenia błędów były następujące:

- w badaniu synchronizacji rozpoczęcia ruchu: o 98% dla e_{d12} , o 87% dla e_{d13} oraz o 0% dla e_{d23} ;
- w badaniu synchronicznego ruchu grupy po zadanej trajektorii o 94% dla e_{d12} , o 66% dla e_{d13} oraz o 77% dla e_{d23} .

Rozdział 6

BADANIA DOŚWIADCZALNE ALGORYTMU SYNCHRONICZNEGO STEROWANIA GRUPĄ ROBOTÓW MOBILNYCH O STRUKTURZE ROZPROSZONEJ

Badania symulacyjne systemu sterowania przedstawione w rozdziale 5 wykazały poprawność działania algorytmu realizującego zadanie śledzenia trajektorii. W kolejnym etapie algorytm sterowania zaimplementowano na rzeczywistych robotach TURTLEBOT 2 i przebadano w warunkach laboratoryjnych. Ze względu na wybrany obiekt badań opisany w rozdziale 2 konieczne było opracowanie systemu sterowania z wykorzystaniem platformy Robot Operating System (ROS), z którą współpracują roboty TURTLEBOT 2. Przedmiotowy system sterowania opracowano przy użyciu języka C++.

Badania doświadczalne również przeprowadzono w dwóch etapach. W pierwszej kolejności zrealizowano badania śledzenia trajektorii wzorcowych przez jednego robota, za co odpowiedzialna jest warstwa wewnętrzna W2 algorytmu sterowania przedstawionego na Rys. 4.1. Następnie zbadano warstwę zewnętrzną W1 algorytmu sterowania przedstawionego na Rys. 4.1, która odpowiada za synchronizację i utrzymanie zadanej formacji grupy robotów. Program badań warstwy W2 przedstawiono w rozdziale 6.1, a ich wyniki w rozdziale 6.2. Program badań warstwy W1 przedstawiono w rozdziale 6.3, a ich wyniki w rozdziale 6.4.

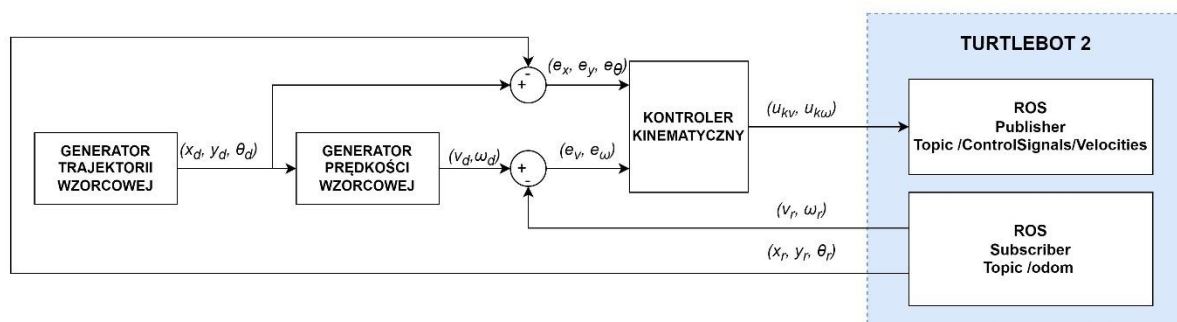
6.1. Zakres i metodyka badań doświadczalnych warstwy wewnętrznej W2 algorytmu sterowania

Zakres i metodyka badań doświadczalnych warstwy wewnętrznej W2 są zgodne z zakresem i metodyką badań symulacyjnych (opisaną w rozdziale 5). W celu porównania wyników badań symulacyjnych z doświadczalnymi przyjęto takie same parametry początkowe, nastawy kontrolerów i warianty badań. W badaniach wykorzystano system sterowania uwzględniający kontroler kinematyczny (4.18) i dynamiczny (4.28). Komunikację pomiędzy wątkami systemu sterowania zrealizowano wykorzystując dwa narzędzia platformy ROS: Publisher - do publikowania określonego typu wiadomości w zadeklarowanym kanale

komunikacyjnym oraz Subscriber – do odbierania wiadomości z systemu robota nadawanych w zadeklarowanym węźle komunikacyjnym. Za pomocą Publishera do systemu robota wysyłane były sygnały sterujące w postaci prędkości liniowej i kątowej. Za pomocą Subscribiera były odbierane z systemu robota pozycja i orientacja.

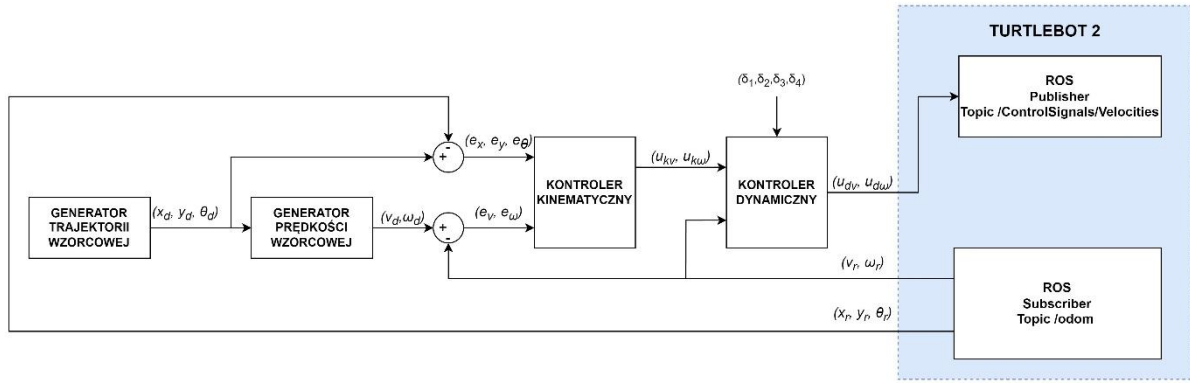
W badaniach doświadczalnych wykorzystano trajektorie testowe: sinusoidalną, kołową oraz trajektorię w kształcie cyfry osiem, opisane wzorami (6.1 – 6.3). Dla wybranych trajektorii w badaniach doświadczalnych przyjęto następujące parametry: $x_0 = 0$, $y_0 = 0$, $R = 1$ m, $\omega = 0.1 \frac{\text{rad}}{\text{s}}$, $k = (0, 1)$ oraz prędkości liniową i kątową wyznaczone na podstawie wzorów (5.2) i (5.3). Badania przeprowadzono dla trzech konfiguracji układu sterowania: śledzenia trajektorii za pomocą kontrolera kinematycznego, śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego oraz śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego z adaptacją.

Badanie doświadczalne śledzenia trajektorii za pomocą kontrolera kinematycznego przeprowadzono zgodnie ze schematem przedstawionym na Rys. 6.1. W celu porównania wyników badań doświadczalnych z wynikami badań teoretycznych zastosowano takie same wartości wzmocnień k_1, k_2 kontrolera kinematycznego (4.18) wyznaczone ze wzoru (4.19) dla $\varepsilon = 0.7$ oraz $g = 65$.



Rys. 6.1. Schemat badania doświadczalnego śledzenia trajektorii za pomocą kontrolera kinematycznego

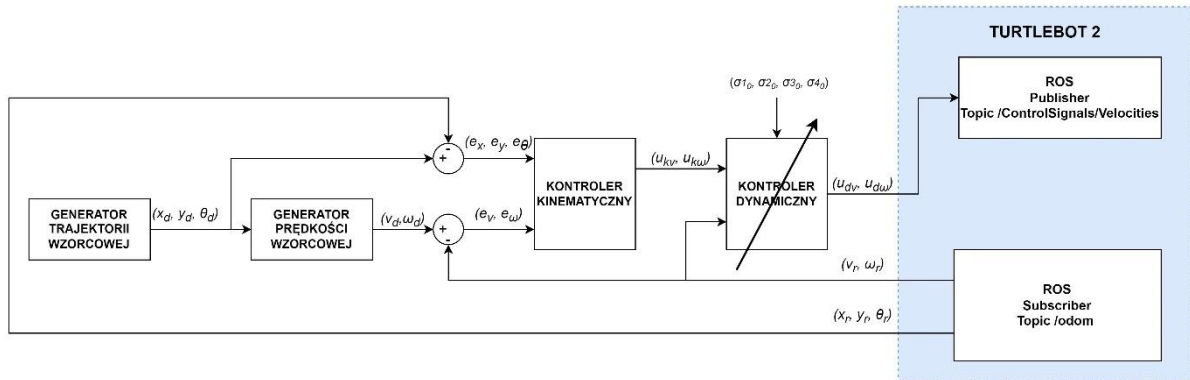
Badanie śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego przeprowadzono zgodnie ze schematem przedstawionym na Rys. 6.2. W celu porównania wyników badań doświadczalnych z wynikami badań teoretycznych zastosowano takie same wartości wzmocnień k_1, k_2 , kontrolera kinematycznego (4.18) wyznaczone ze wzoru (4.19) dla $\varepsilon = 0.7$ oraz $g = 65$ oraz wartości wzmocnień kontrolera dynamicznego (4.28) $k_v = 1.2$, $k_\omega = 4$, $l_v = l_\omega = 0.5$.



Rys. 6.2. Schemat badania doświadczalnego śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego

Badanie śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego z adaptacją przeprowadzono zgodnie ze schematem przedstawionym na Rys. 6.3. W celu porównania wyników badań doświadczalnych z wynikami badań teoretycznych zastosowano takie same wartości wzmocnień k_1, k_2 , kontrolera kinematycznego (4.18), wyznaczone ze wzoru (4.19) dla $\varepsilon = 0.7$ oraz $g = 65$, wartości wzmocnień kontrolera dynamicznego $k_v = 1.2$, $k_\omega = 4$, $l_v = 0.5$, $l_\omega = 0.5$ oraz wartości wzmocnień dla adaptacji parametrów modelu:

$$\gamma = \begin{bmatrix} 4.5 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0.0005 & 0 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix} \quad (7.1)$$



Rys. 6.3. Schemat badania doświadczalnego śledzenia trajektorii za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

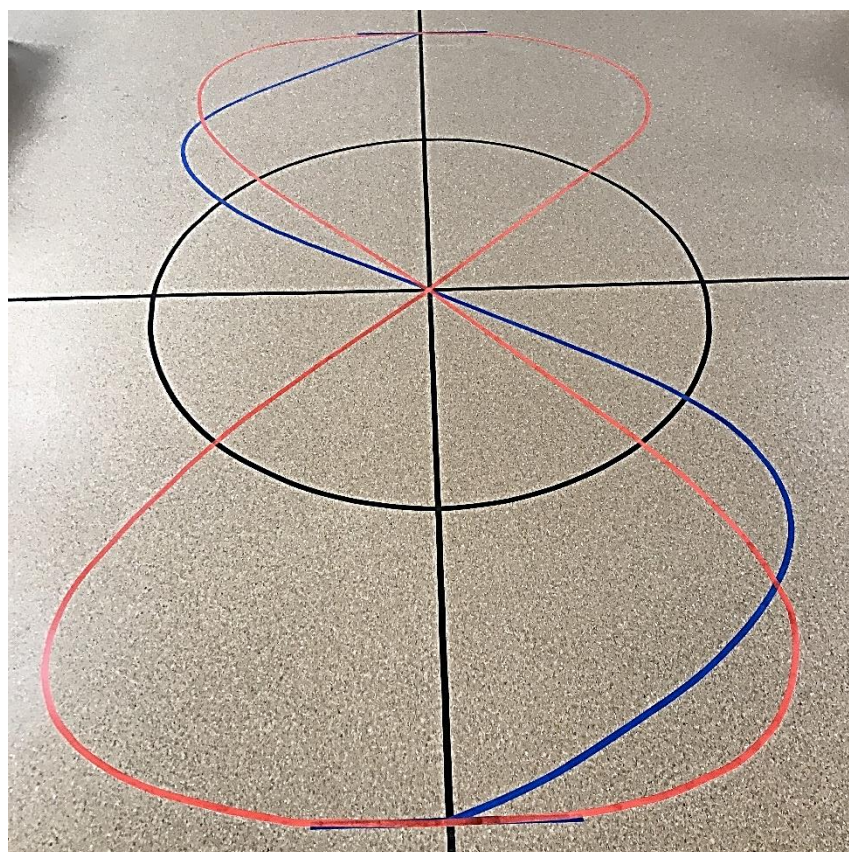
W badaniach założono pozycję początkową robota zgodną z początkowym punktem trajektorii ($x_{d0} = 0, y_{d0} = 0$ dla trajektorii sinus i „ósemka”, $x_{d0} = 0, y_{d0} = 1$ dla trajektorii „okrąg”) oraz prędkość liniową $v_d = 0.25 \frac{m}{s}$. W celu ograniczenia błędów pomiaru orientacji

powodowanych drganiami konstrukcji robota parametry ruchu ograniczono przyspieszeniami: liniowym $a_v = 0.4 \frac{m}{s^2}$ i kątowym $a_\omega = 0.8 \frac{rad}{s^2}$.

6.2. Badania doświadczalne warstwy wewnętrznej W2 algorytmu sterowania

Badania doświadczalne śledzenia trajektorii przez jednego robota przeprowadzono w laboratorium Instytutu Techniki Rakietowej i Mechatroniki na specjalnie przygotowanym placu manewrowym, przedstawionym na Rys. 6.4.

Wyniki badań przedstawiono w formie wykresów: porównania trajektorii wzorcowej i rzeczywistej, błędów pozycji i orientacji, błędów prędkości oraz sygnałów sterujących, którymi są prędkości liniowa i kątowa. Jakość zaproponowanych układów regulacji oceniono na podstawie wartości uchybów – różnicy pomiędzy położeniem, orientacją i prędkościami wzorcowymi, a rzeczywistymi



Rys. 6.4. Plac manewrowy do testowania algorytmów śledzenia trajektorii przez roboty mobilne

W badaniach laboratoryjnych początkowa pozycja i orientacja robota była opisana wektorem współrzędnych konfiguracyjnych, w postaci:

- dla trajektorii „okrąg” - $q_0 = [0, 1, 0]$;
- dla trajektorii „sinus” - $q_0 = [0, 0, 0]$;
- dla trajektorii „ósemka” - $q_0 = [0, 0, 0]$.

W przypadku trajektorii „sinus” i „ósemka” w pierwszych krokach realizacji algorytmu sterowania należało zapewnić warunki początkowe jak dla trajektorii „okrąg” czyli zgodność początkowej orientacji robota z orientacją w punkcie początkowym trajektorii wzorcowej. W związku z tym konieczna była reorientacja robota tak aby osiągnąć wektor współrzędnych konfiguracyjnych w postaci:

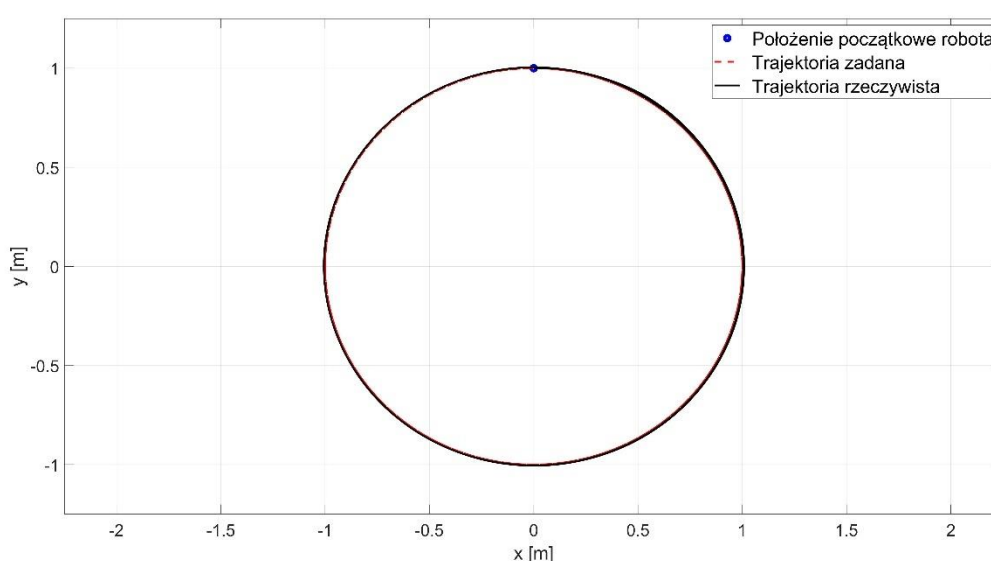
- dla trajektorii „sinus” - $q_0 = \left[0, 0, \frac{\pi}{6}\right]$;
- dla trajektorii „ósemka” - $q_0 = \left[0, 0, \frac{\pi}{4}\right]$.

6.2.1. Badania kontrolera kinematycznego

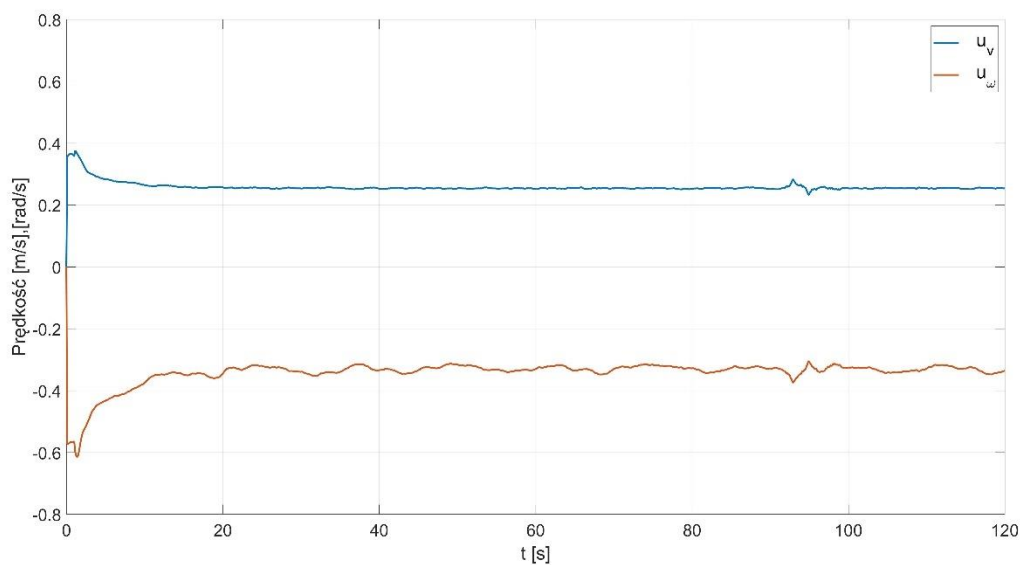
Wyniki badań doświadczalnych śledzenia trajektorii wzorcowej za pomocą kontrolera kinematycznego przedstawiono na Rys. 6.5 - Rys. 6.28. Badania przeprowadzono zgodnie ze schematem przedstawionym na Rys. 6.1.

6.2.1.1. Kryterium ruchu przez określony czas dla kontrolera kinematycznego

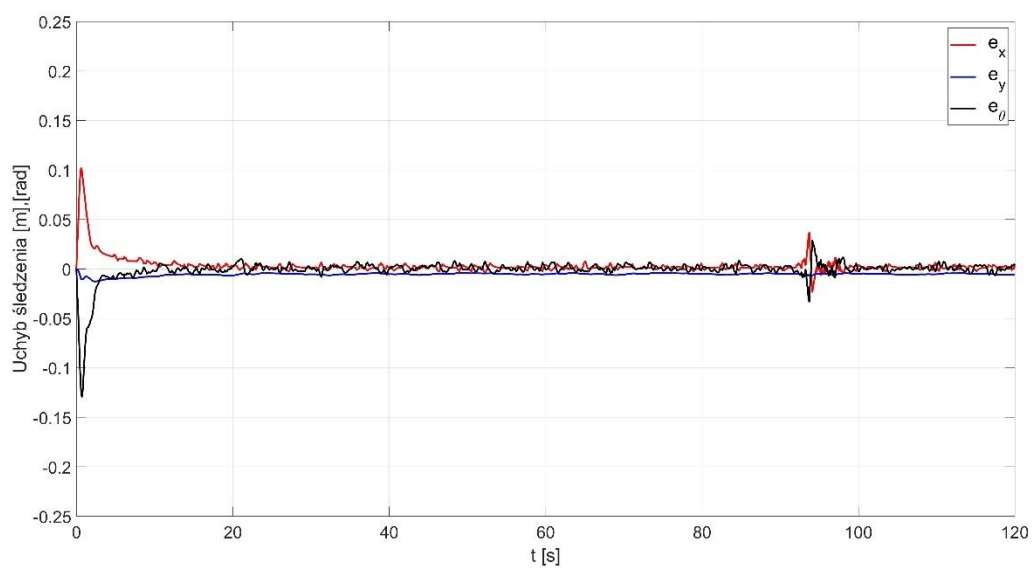
- Śledzenie trajektorii „okrąg”



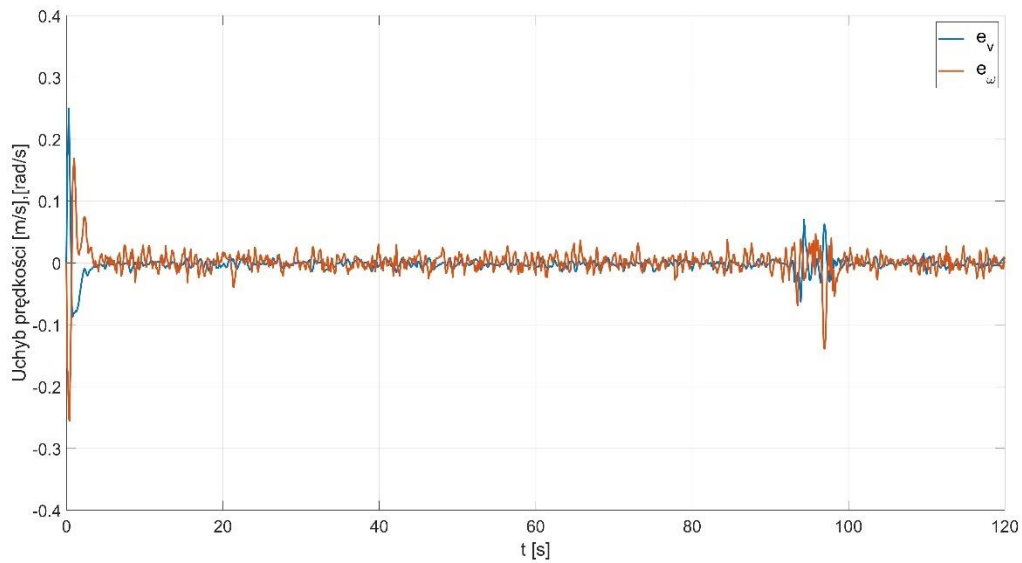
Rys. 6.5. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 6.6. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w badaniu laboratoryjnym śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 6.7. Uchyby śledzenia e_x , e_y , e_{θ} w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego



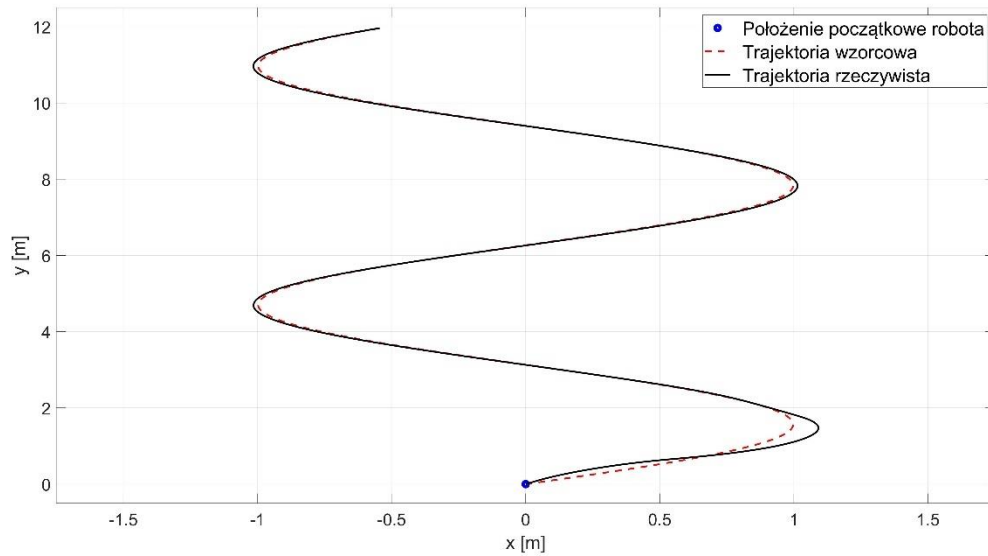
Rys. 6.8. Uchyby prędkości liniowej e_v i kątowej e_ω w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 6.7) oraz prędkości (Rys. 6.8) nie większymi niż:

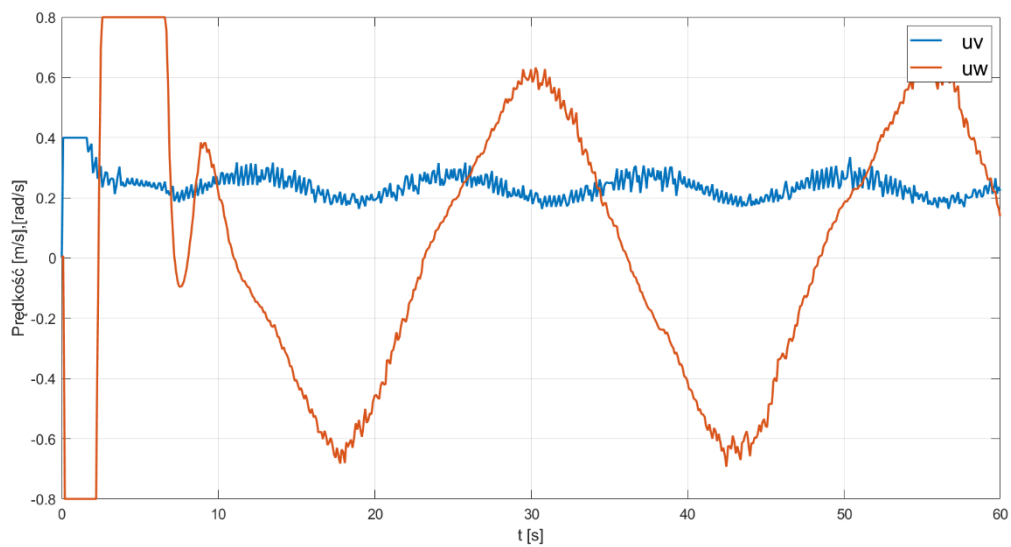
- w kierunku osi x: $e_x = 0.012$ m;
- w kierunku osi y: $e_y = 0.009$ m;
- orientacji: $e_\theta = 0.033$ rad;
- prędkości liniowej $e_v = 0.016 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.031 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 11$ s.

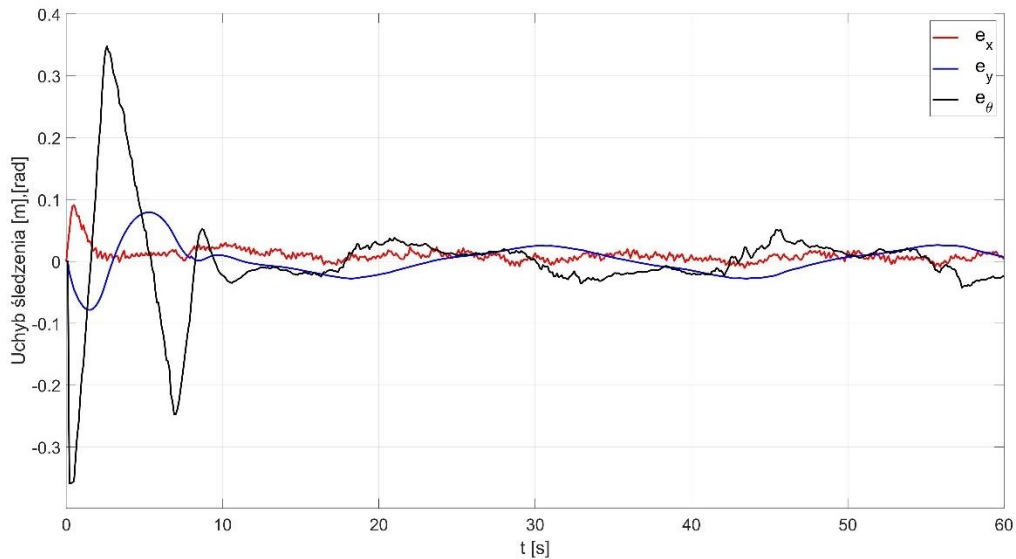
- Śledzenie trajektorii „sinus”



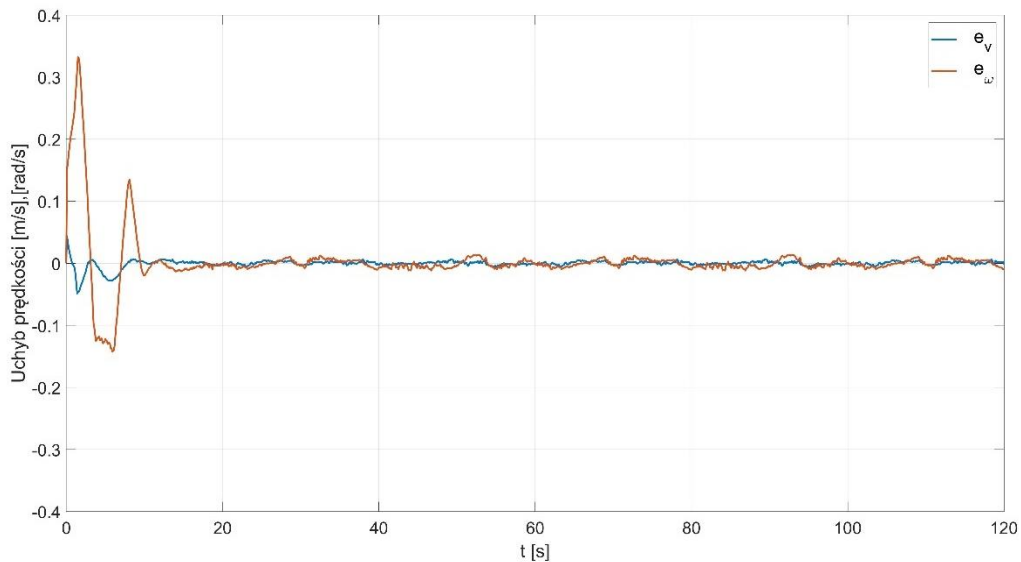
Rys. 6.9. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 6.10. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 6.11. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego



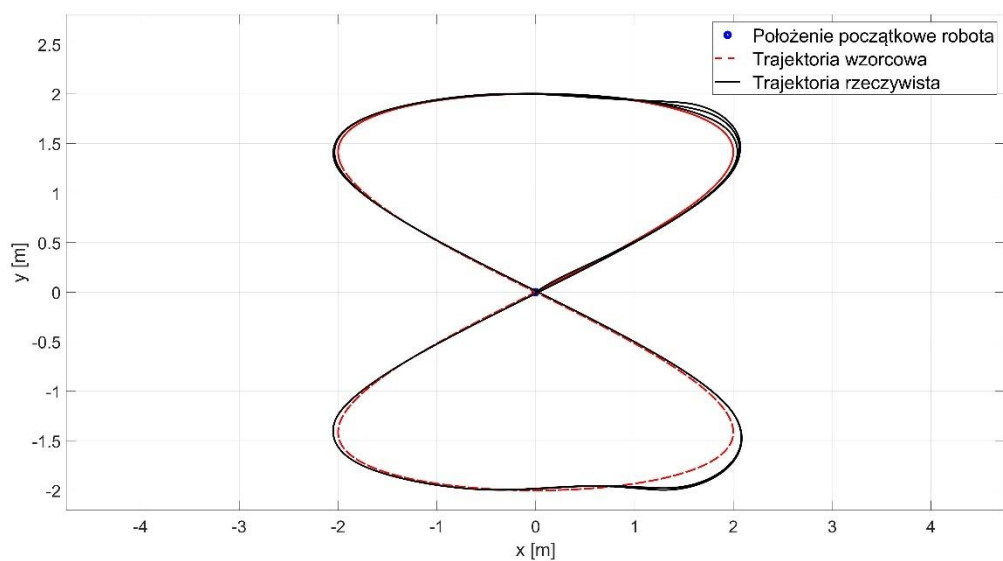
Rys. 6.12. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 6.11) oraz prędkości (Rys. 6.12) nie większymi niż:

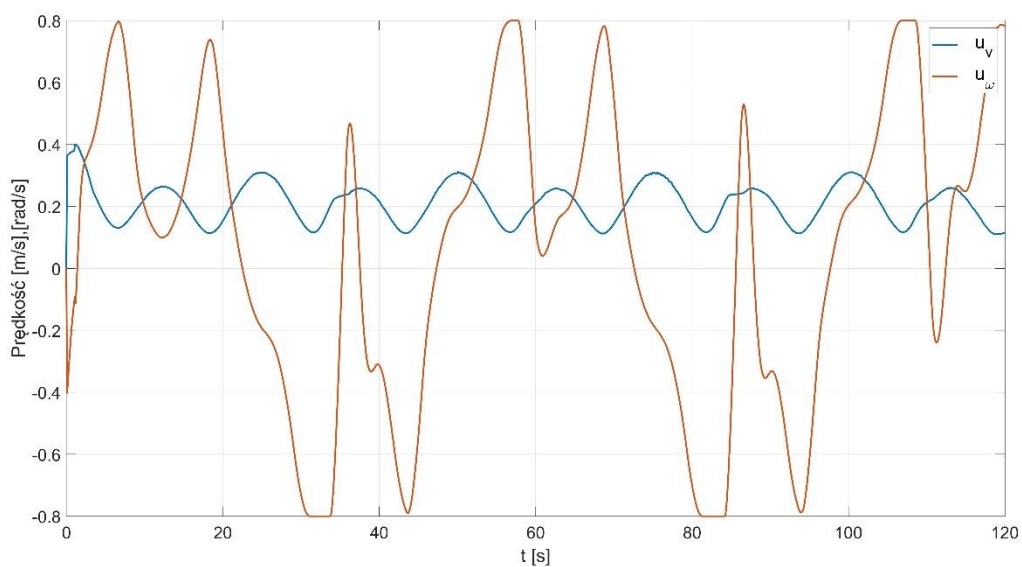
- w kierunku osi x: $e_x = 0.031$ m;
- w kierunku osi y: $e_y = 0.028$ m;
- orientacji: $e_\theta = 0.051$ rad;
- prędkości liniowej $e_v = 0.008 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.014 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 7.4$ s.

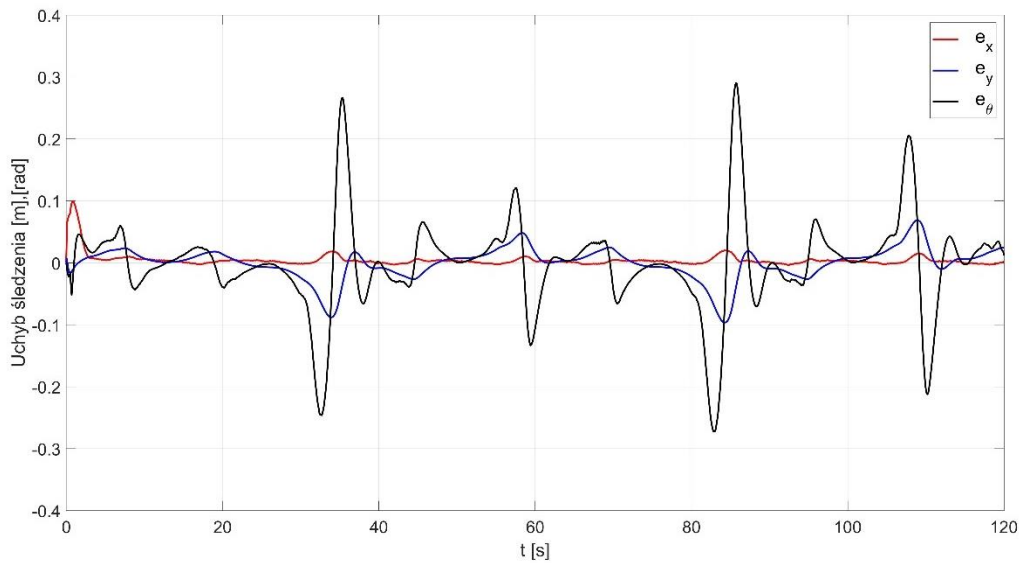
- Śledzenie trajektorii „ósemka”



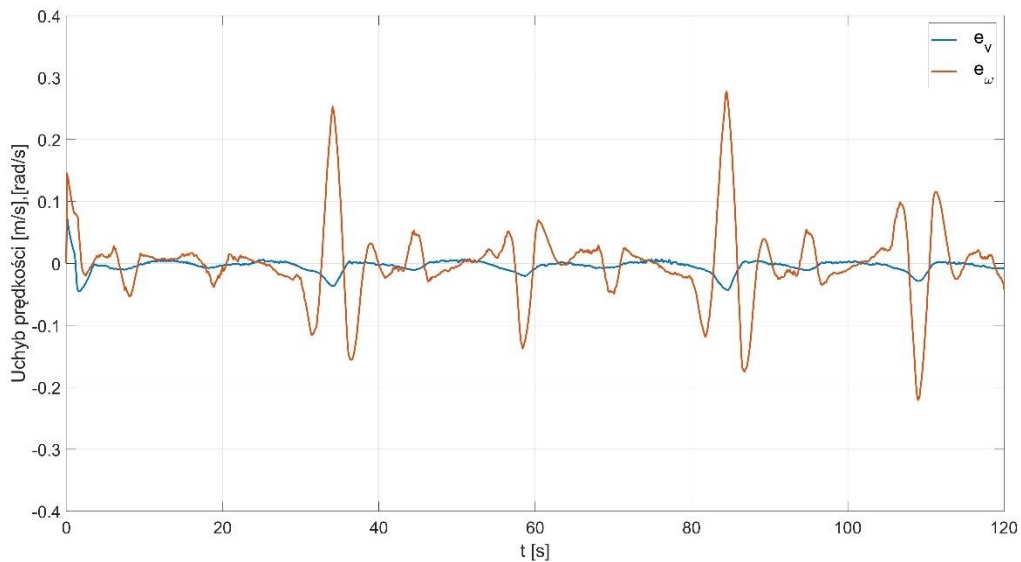
Rys. 6.13. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 6.14. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 6.15. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego



Rys. 6.16. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego

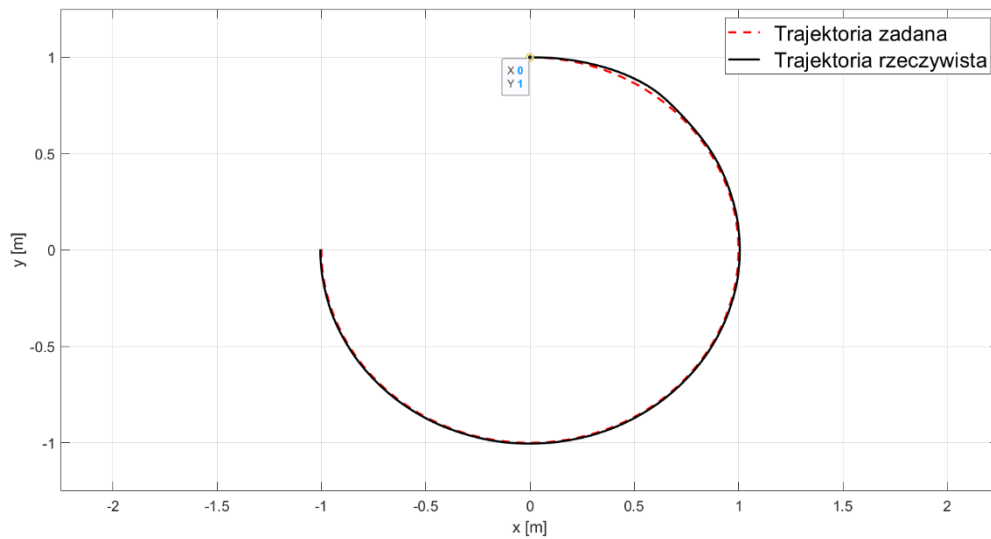
Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 6.15) oraz prędkości (Rys. 6.16) nie większymi niż:

- w kierunku osi x: $e_x = 0.021$ m;
- w kierunku osi y: $e_y = 0.095$ m;
- orientacji: $e_\theta = 0.291$ rad;
- prędkości liniowej $e_v = 0.043 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.277 \frac{\text{rad}}{\text{s}}$.

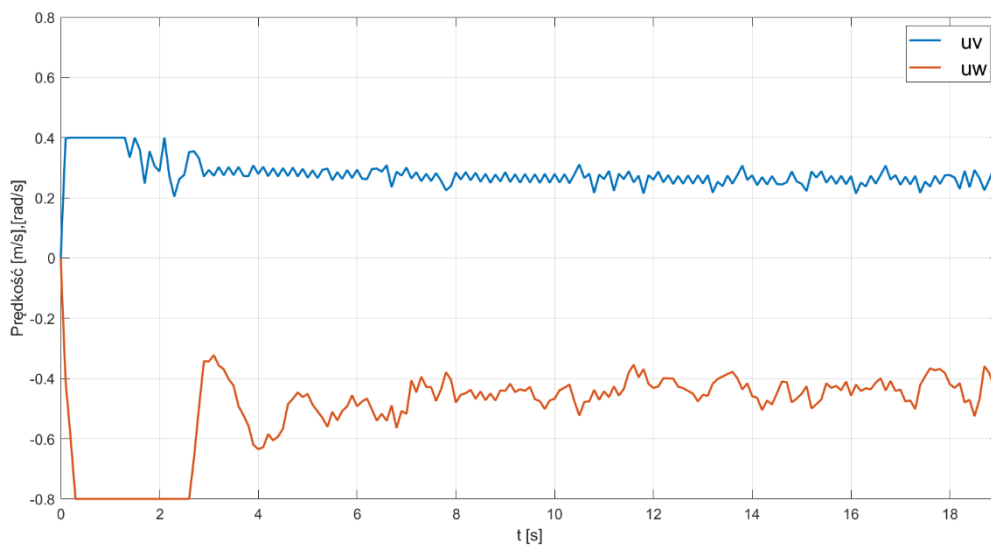
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 3.3$ s.

6.2.1.2. Kryterium ruchu po skończonej drodze dla kontrolera kinematycznego

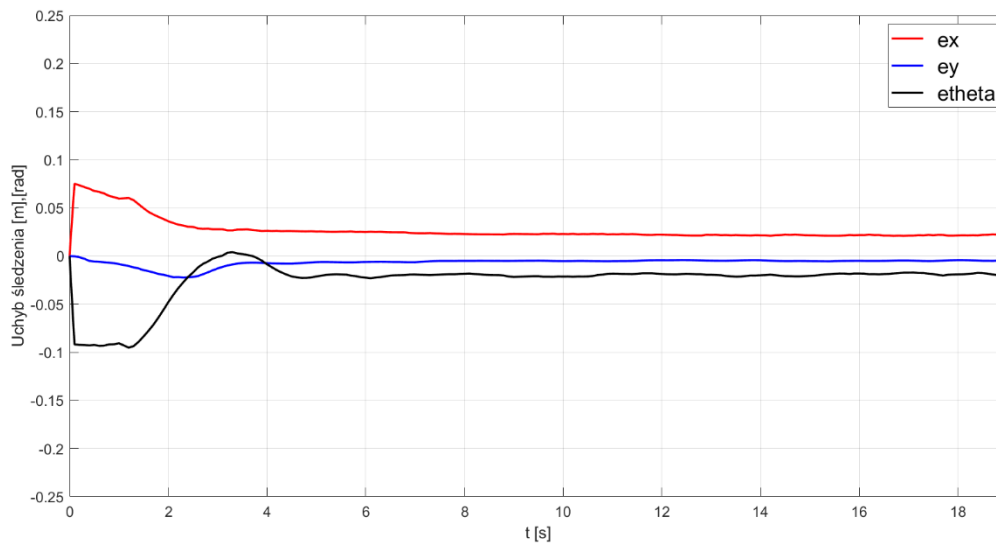
- Śledzenie trajektorii „okrąg”



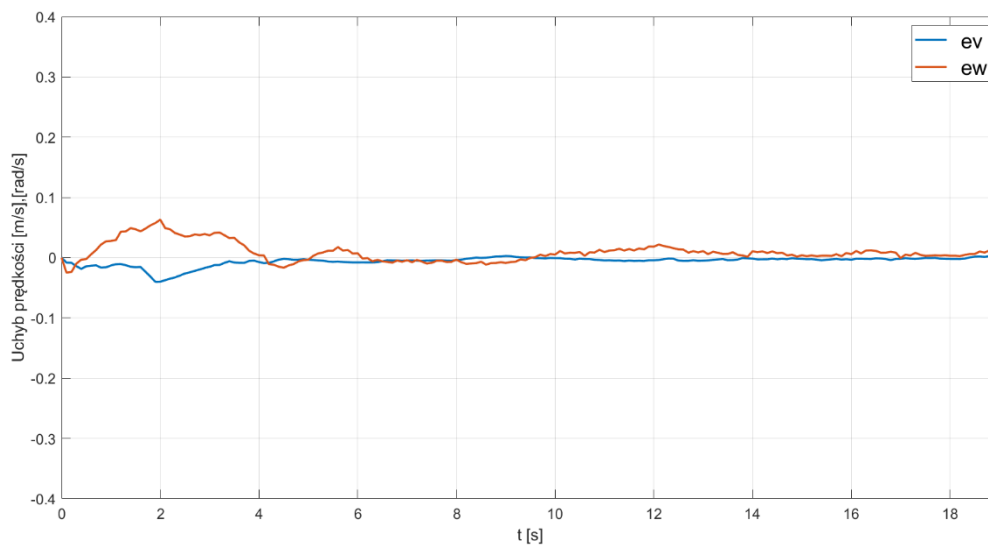
Rys. 6.17. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 6.18. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w badaniu laboratoryjnym śledzenia trajektorii „okrąg” za pomocą kontrolera kinematycznego



Rys. 6.19. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego



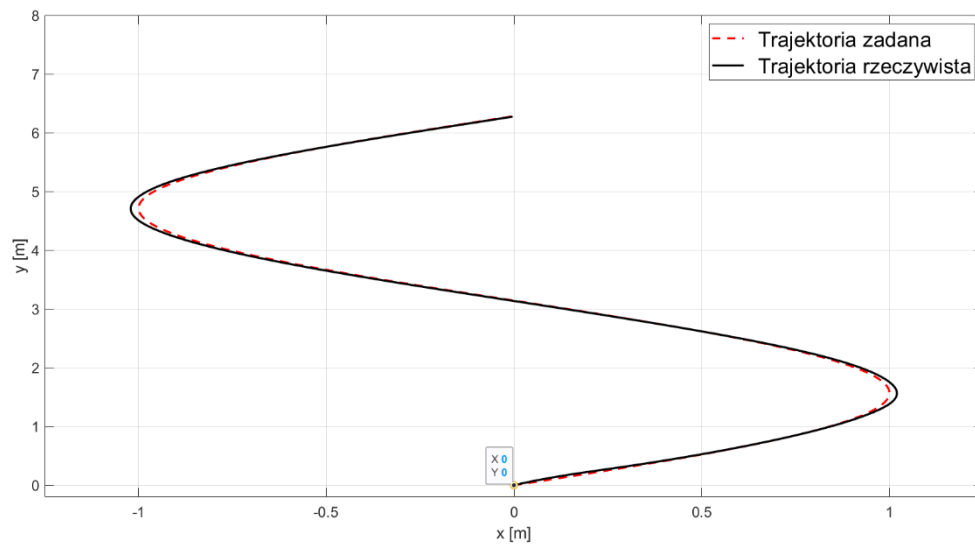
Rys. 6.20. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 6.19) oraz prędkości (Rys. 6.20) nie większymi niż:

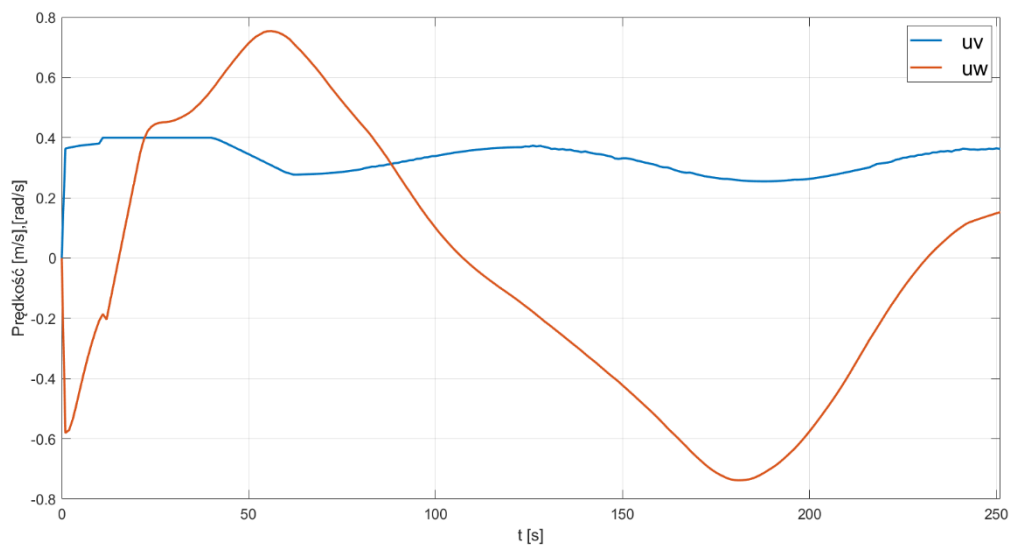
- w kierunku osi x: $e_x = 0.023 \text{ m}$;
- w kierunku osi y: $e_y = 0.005 \text{ m}$;
- orientacji: $e_\theta = 0.021 \text{ rad}$;
- prędkości liniowej $e_v = 0.003 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.012 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 3.6 \text{ s}$.

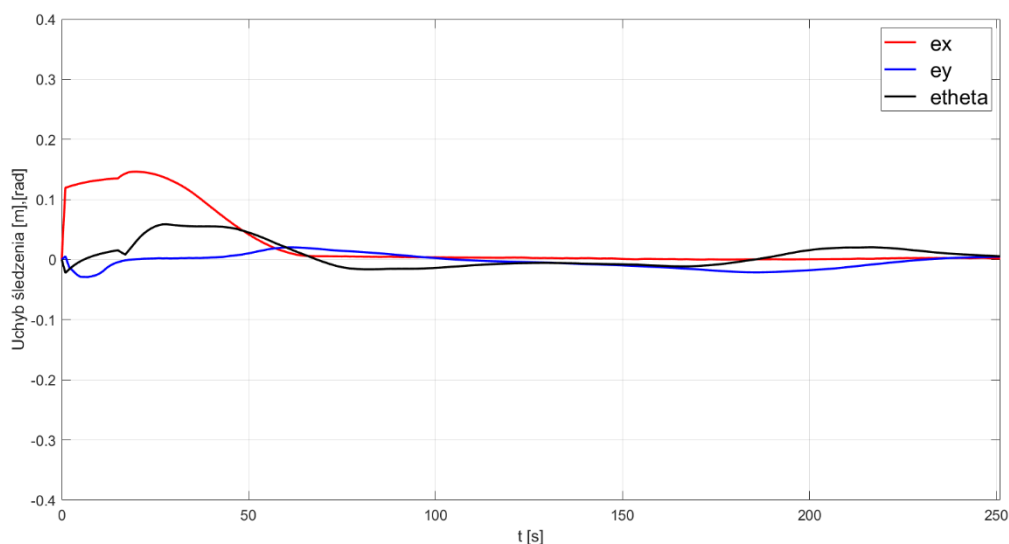
- Śledzenie trajektorii „sinus”



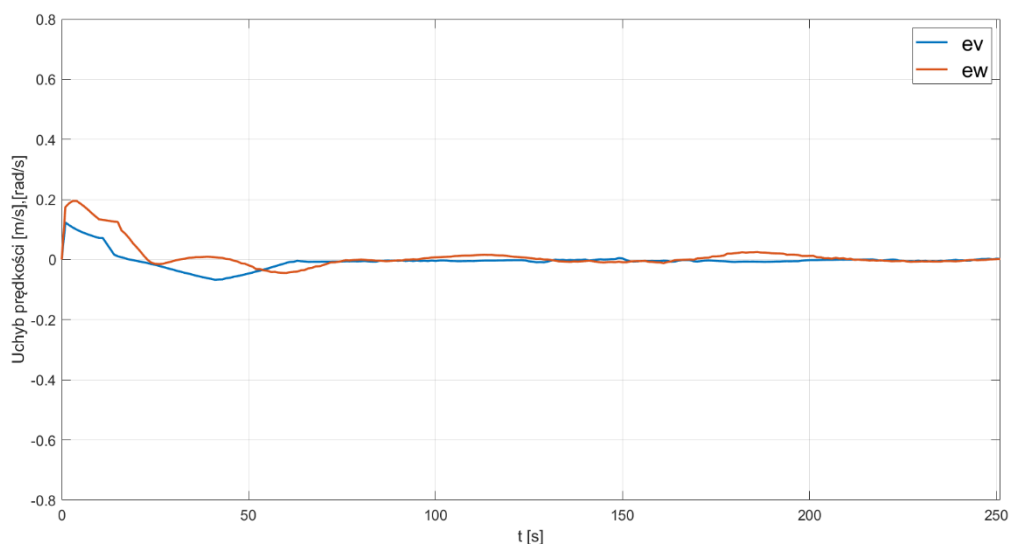
Rys. 6.21. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 6.22. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „sinus” za pomocą kontrolera kinematycznego



Rys. 6.23. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego



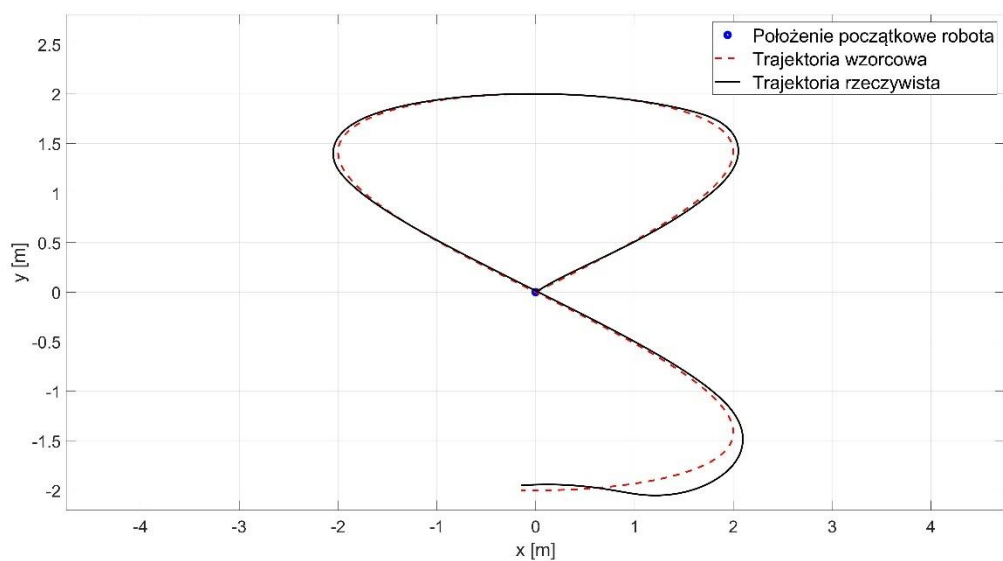
Rys. 6.24. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 6.23) oraz prędkości (Rys. 6.24) nie większymi niż:

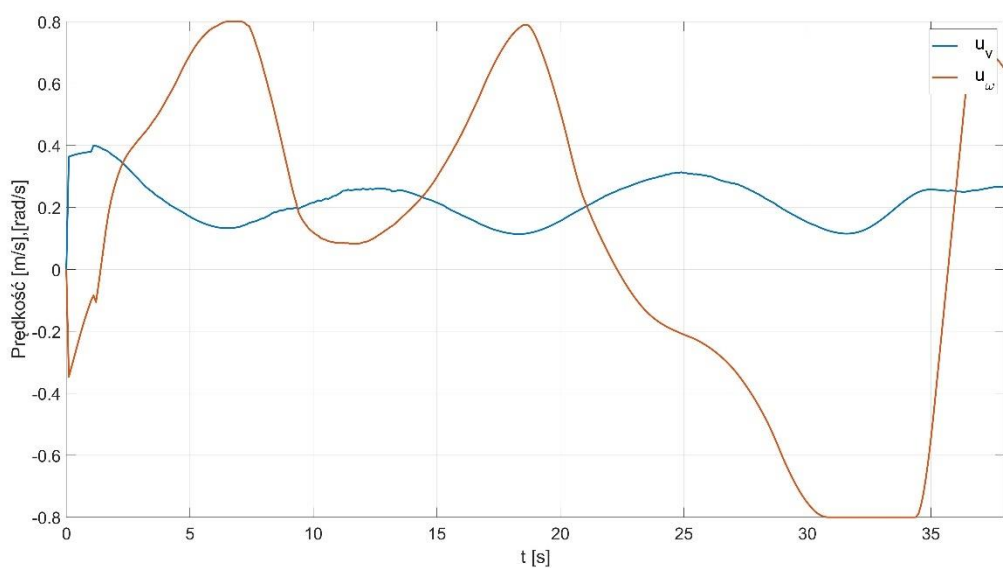
- w kierunku osi x: $e_x = 0.006$ m;
- w kierunku osi y: $e_y = 0.021$ m;
- orientacji: $e_\theta = 0.021$ rad;
- prędkości liniowej $e_v = 0.005 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.025 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 5.9$ s.

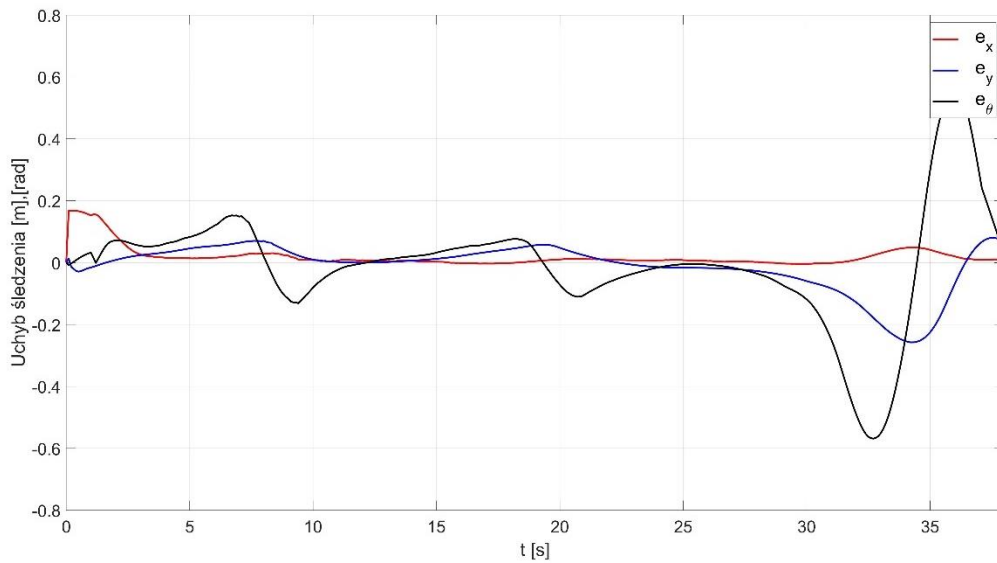
- Śledzenie trajektorii „ósemka”



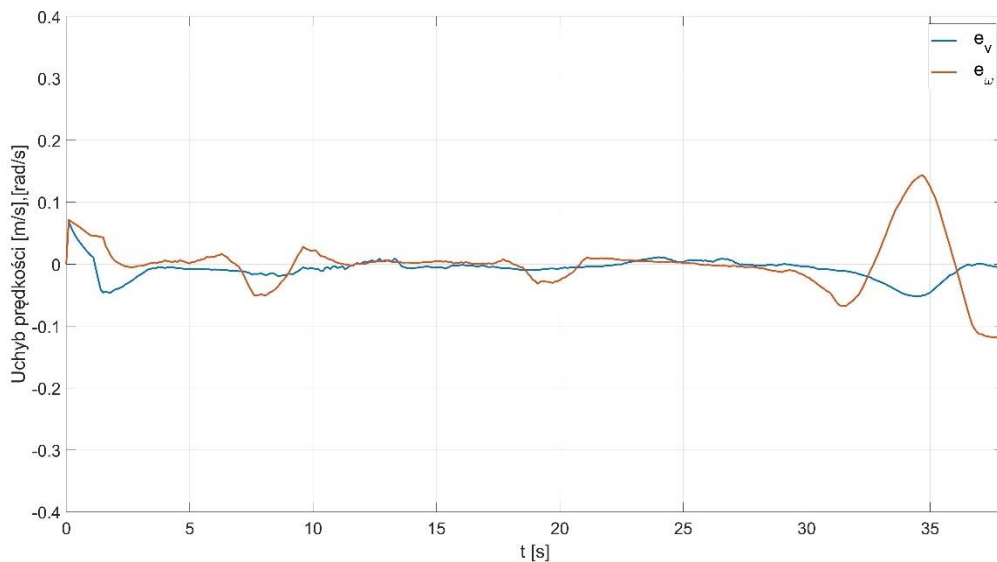
Rys. 6.25. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 6.26. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego



Rys. 6.27. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego



Rys. 6.28. Uchyby prędkości liniowej e_v i kątowej e_ω w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematycznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności, można stwierdzić, że kontroler kinematyczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 6.27) oraz prędkości (Rys. 6.28) nie większymi niż:

- w kierunku osi x: $e_x = 0.005 \text{ m}$;
- w kierunku osi y: $e_y = 0.030 \text{ m}$;
- orientacji: $e_\theta = 0.013 \text{ rad}$;
- prędkości liniowej $e_v = 0.003 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.238 \frac{\text{rad}}{\text{s}}$.

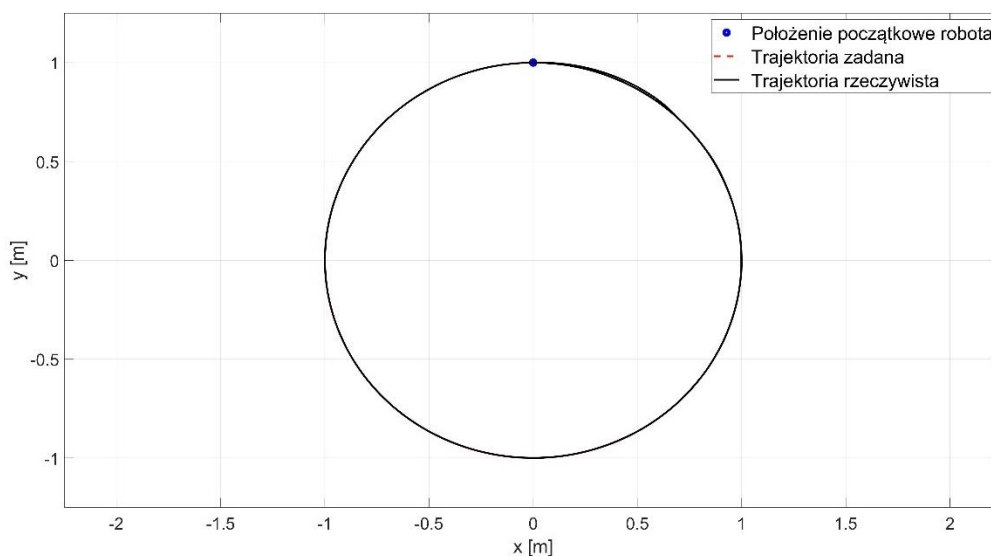
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 3.2 \text{ s}$.

6.2.2. Badania kontrolera kinematyczno-dynamicznego

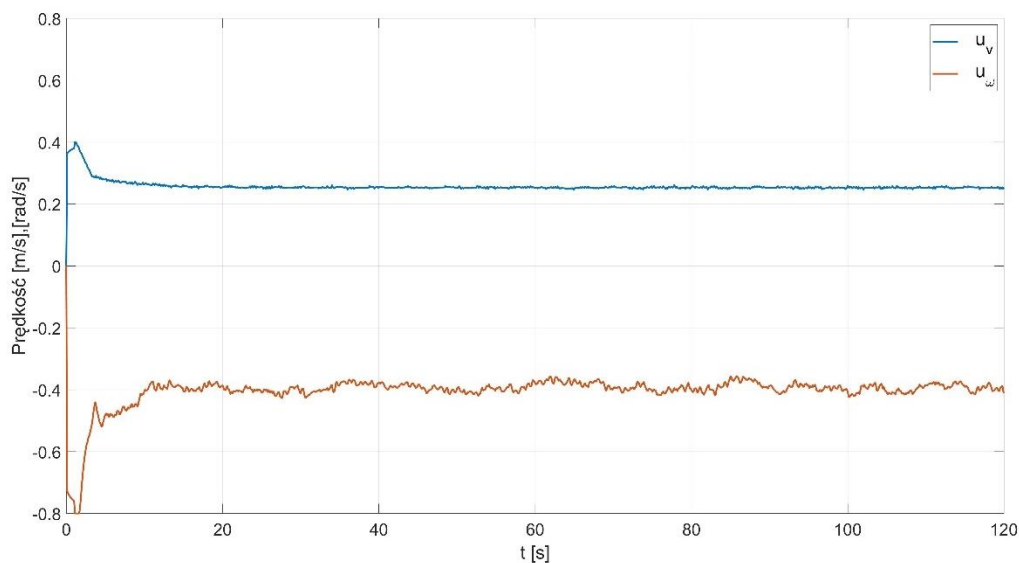
Wyniki badań doświadczalnych śledzenia trajektorii wzorcowej za pomocą kontrolera dynamicznego przedstawiono na Rys. 6.29 - Rys. 6.52. Badania przeprowadzono zgodnie ze schematem przedstawionym na Rys. 6.2.

6.2.2.1. Kryterium ruchu przez określony czas dla kontrolera kinematyczno-dynamicznego

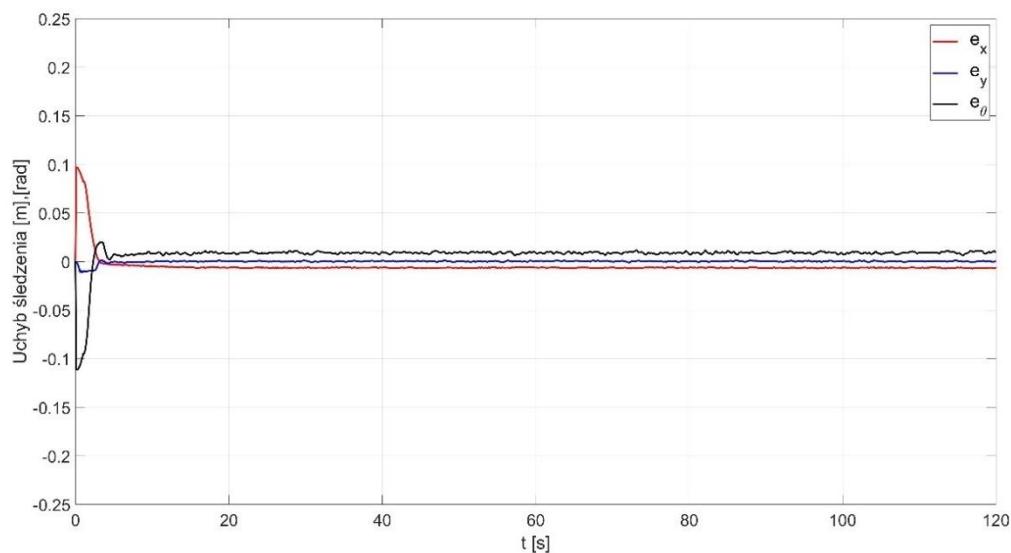
- Śledzenie trajektorii „okrąg”



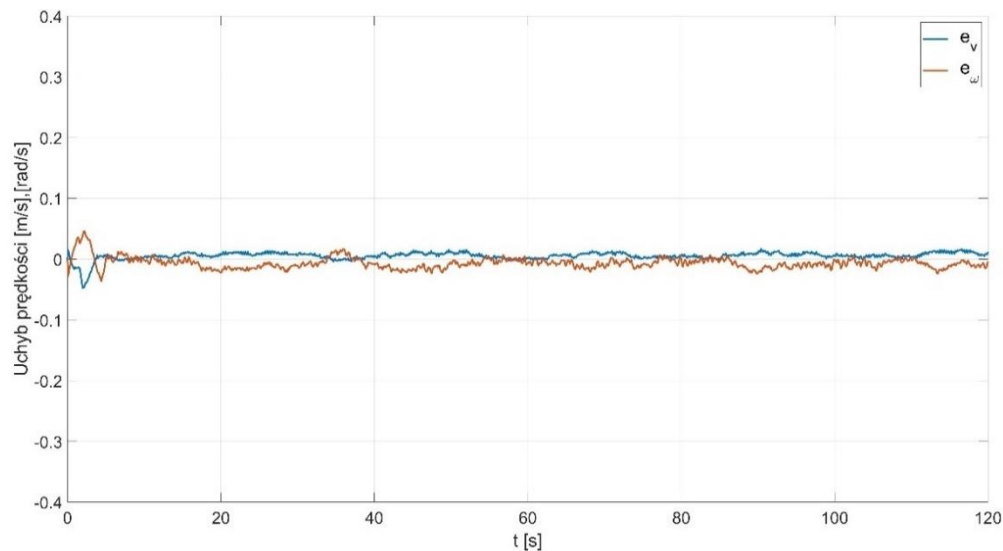
Rys. 6.29. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.30. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w badaniu laboratoryjnym śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.31. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



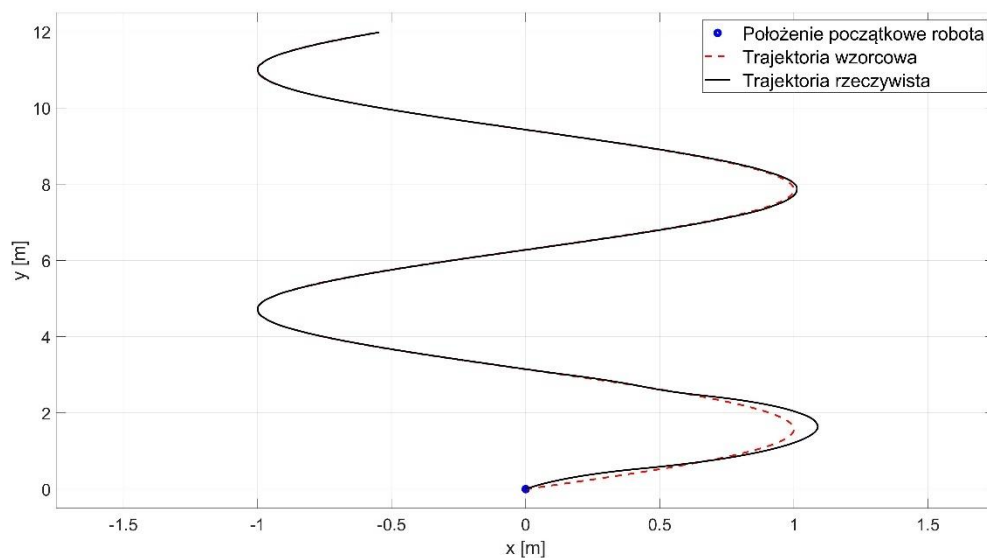
Rys. 6.32. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 6.31) oraz prędkości (Rys. 6.32) nie większymi niż:

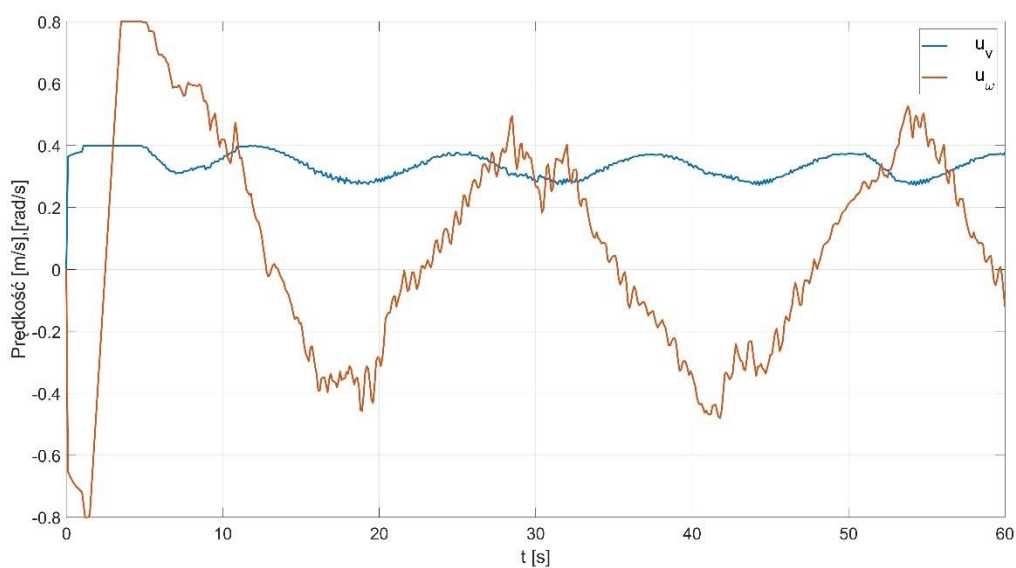
- w kierunku osi x: $e_x = 0.006$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.011$ rad;
- prędkości liniowej $e_v = 0.017 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.036 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 3.3$ s.

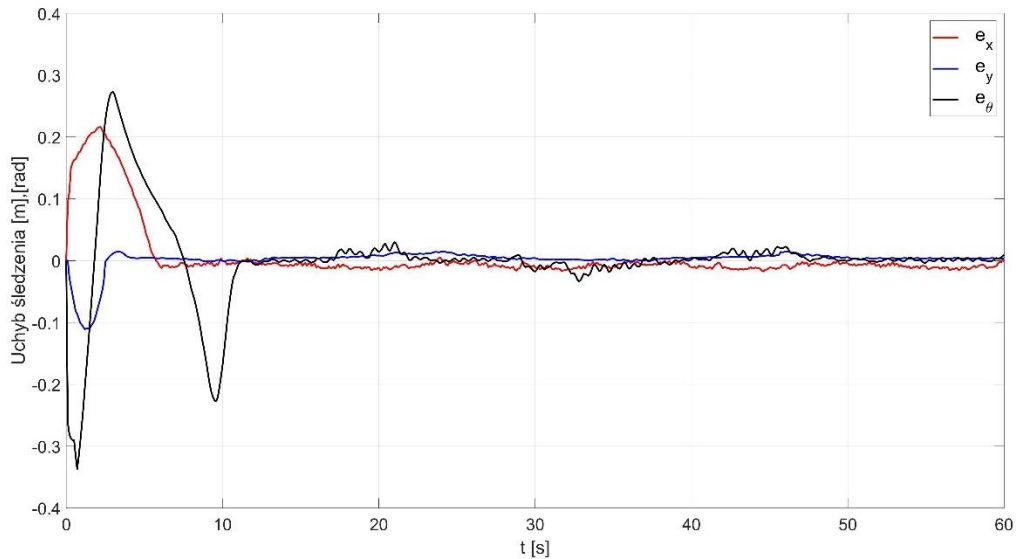
- Śledzenie trajektorii „sinus”



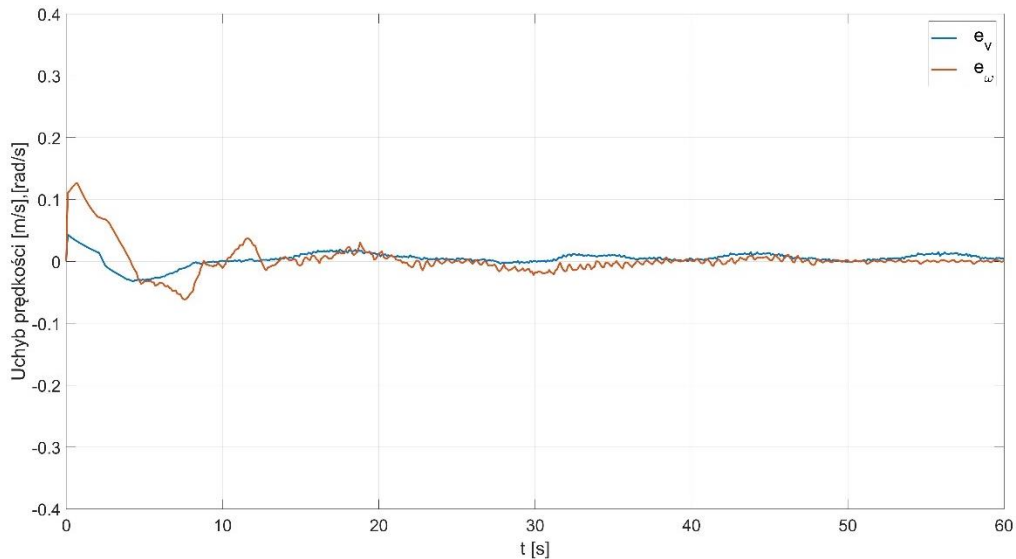
Rys. 6.33. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.34. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.35. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego



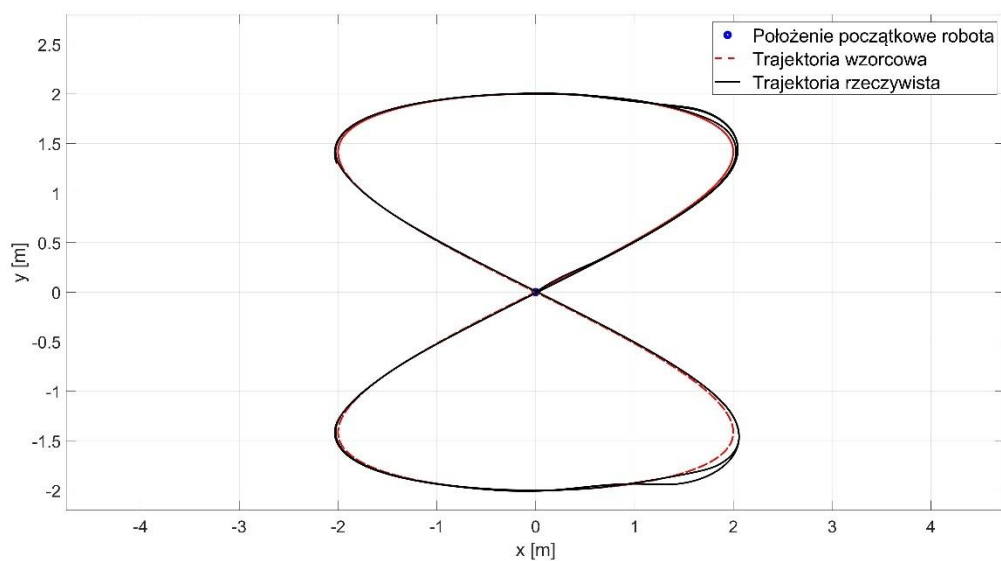
Rys. 6.36. Uchyby prędkości liniowej e_v i kątowej e_ω w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 6.35) oraz prędkości (Rys. 6.36) nie większymi niż:

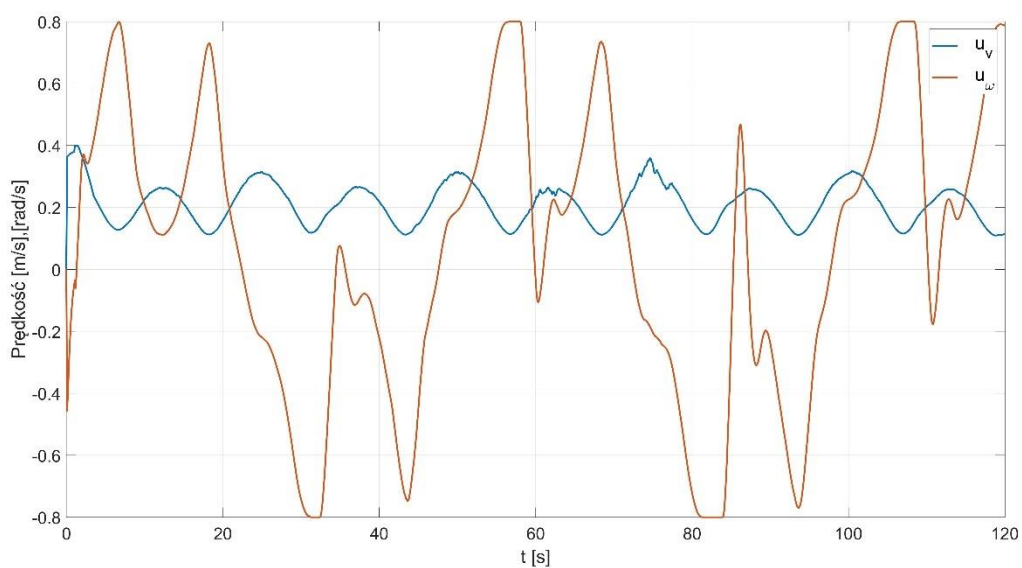
- w kierunku osi x: $e_x = 0.017$ m;
- w kierunku osi y: $e_y = 0.015$ m;
- orientacji: $e_\theta = 0.033$ rad;
- prędkości liniowej $e_v = 0.030 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.015 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 5.5$ s.

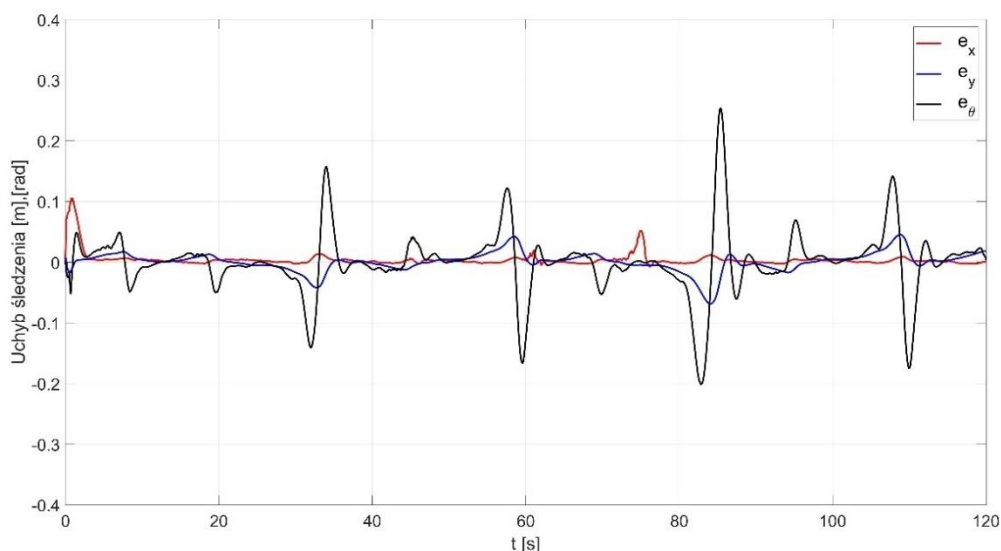
- Śledzenie trajektorii „ósemka”



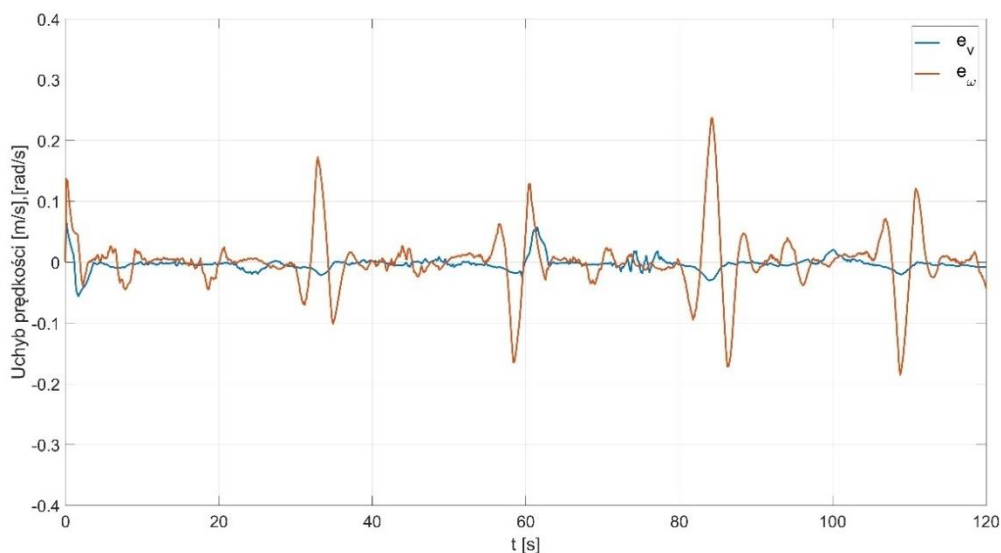
Rys. 6.37. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.38. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.39. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego.



Rys. 6.40. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego

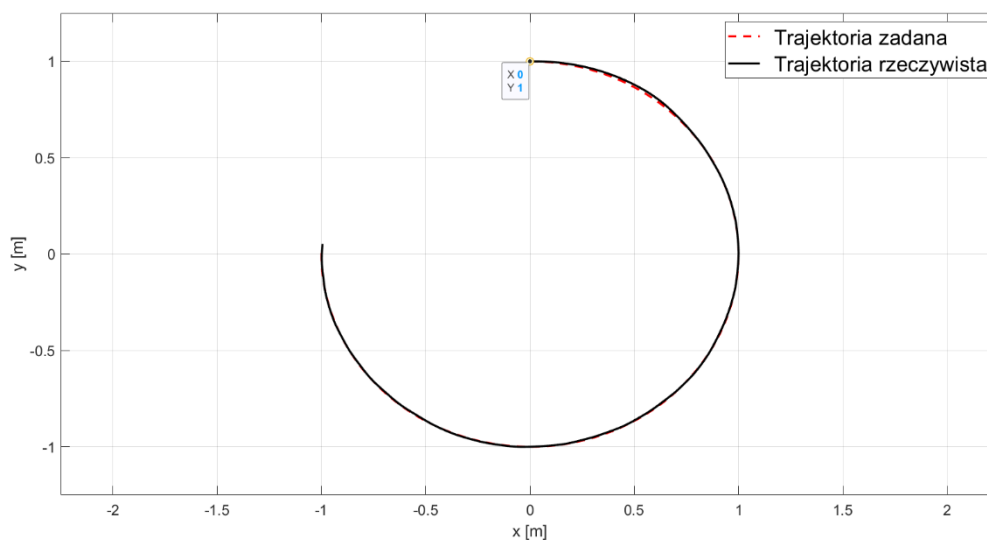
Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 6.39) oraz prędkości (Rys. 6.40) nie większymi niż:

- w kierunku osi x: $e_x = 0.016$ m;
- w kierunku osi y: $e_y = 0.043$ m;
- orientacji: $e_\theta = 0.025$ rad;
- prędkości liniowej $e_v = 0.029 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.237 \frac{\text{rad}}{\text{s}}$.

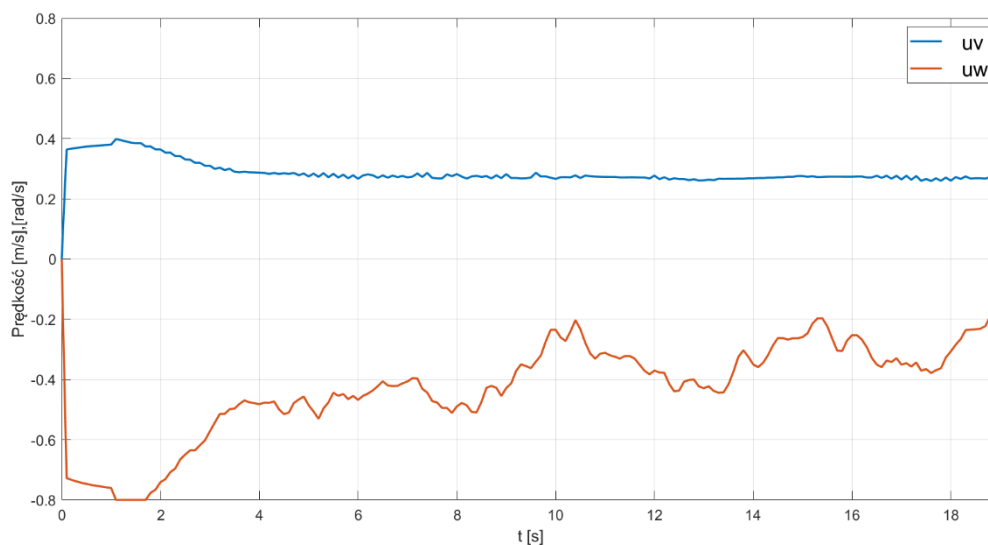
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 2.2$ s.

6.2.2.2. Kryterium ruchu po skończonej drodze dla kontrolera kinematyczno-dynamicznego

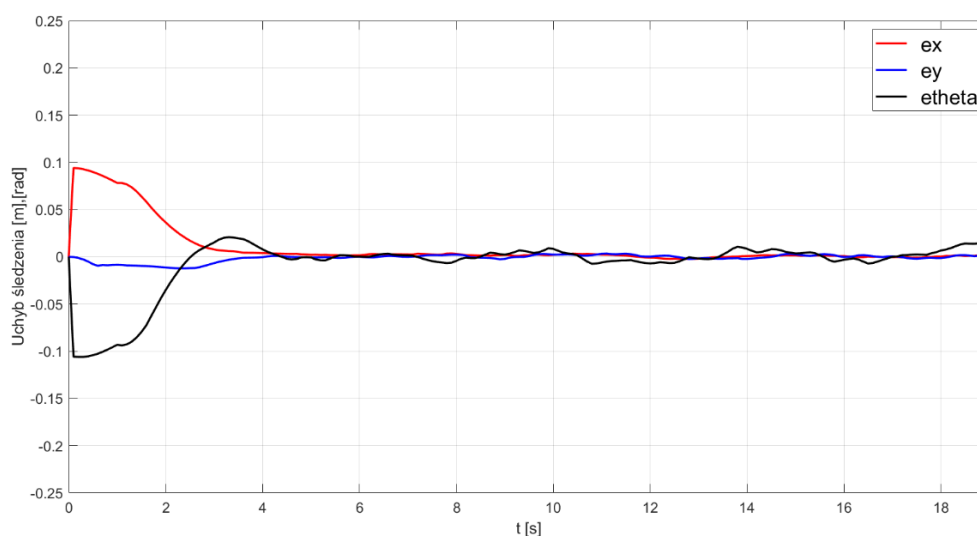
- Śledzenie trajektorii „okrąg”



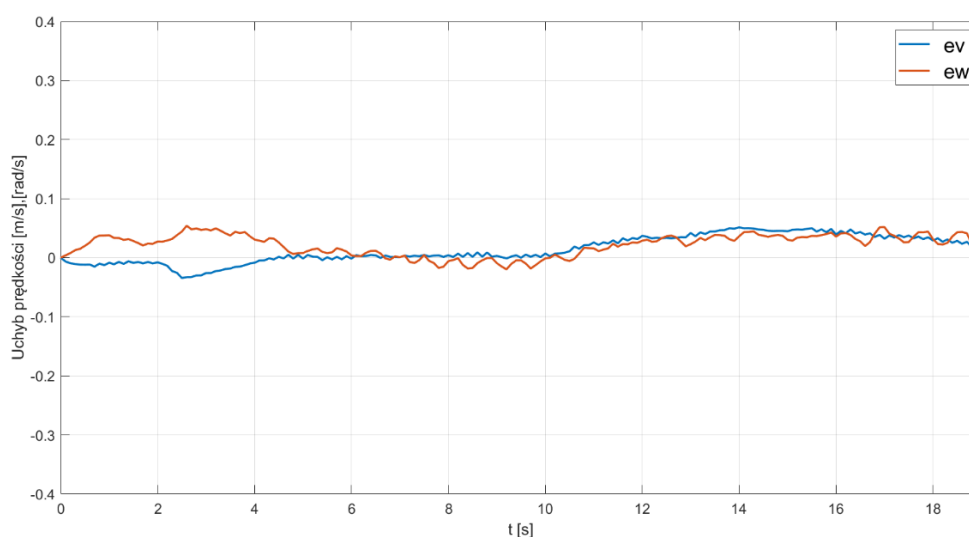
Rys. 6.41. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.42. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w badaniu laboratoryjnym śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.43. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego



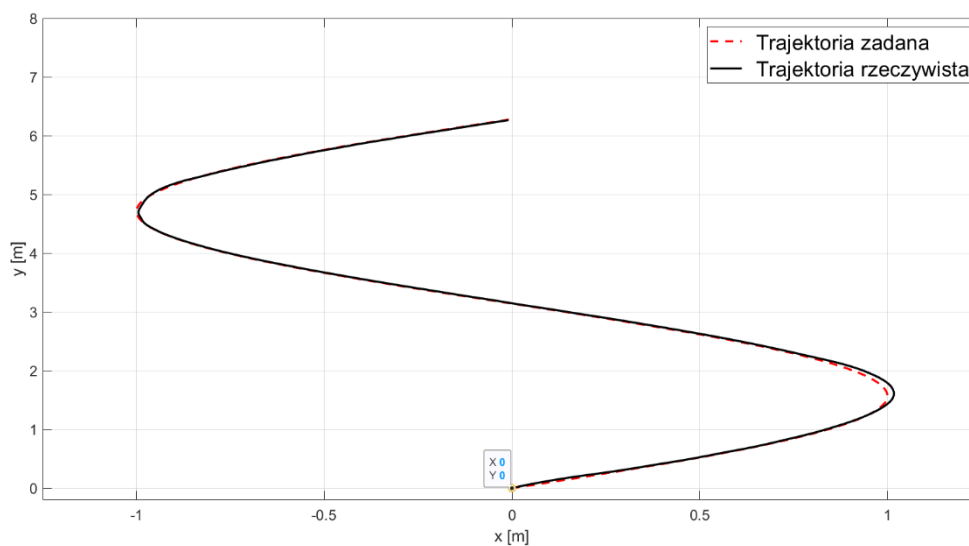
Rys. 6.44. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności śledzenia, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 6.43) oraz prędkości (Rys. 6.44) nie większymi niż:

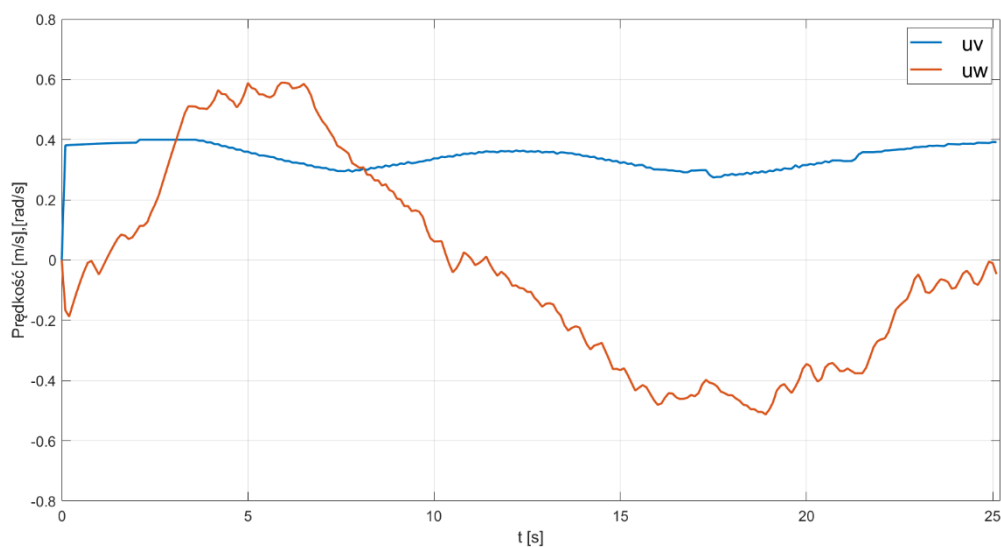
- w kierunku osi x: $e_x = 0.021$ m;
- w kierunku osi y: $e_y = 0.004$ m;
- orientacji: $e_\theta = 0.016$ rad;
- prędkości liniowej $e_v = 0.013 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.029 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 2.5$ s.

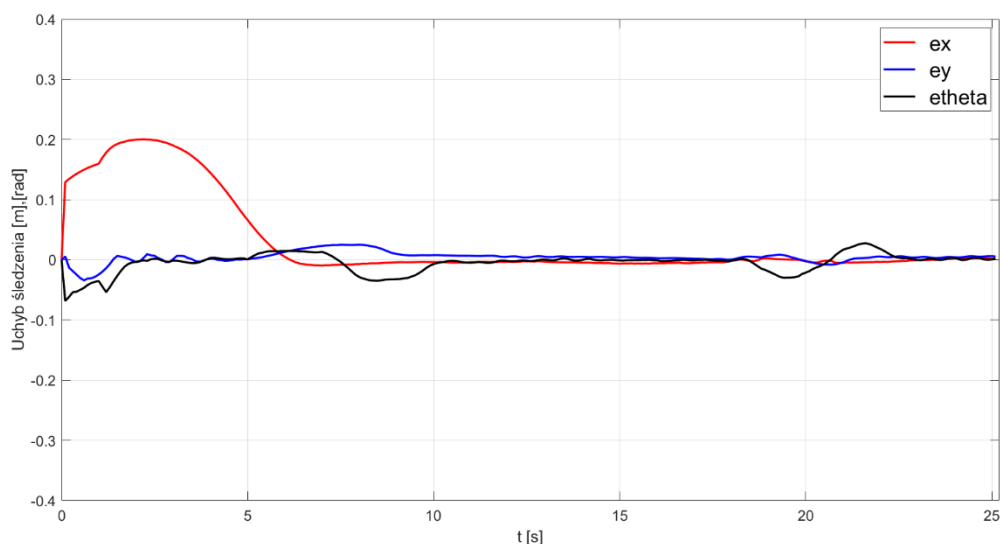
- Śledzenie trajektorii „sinus”



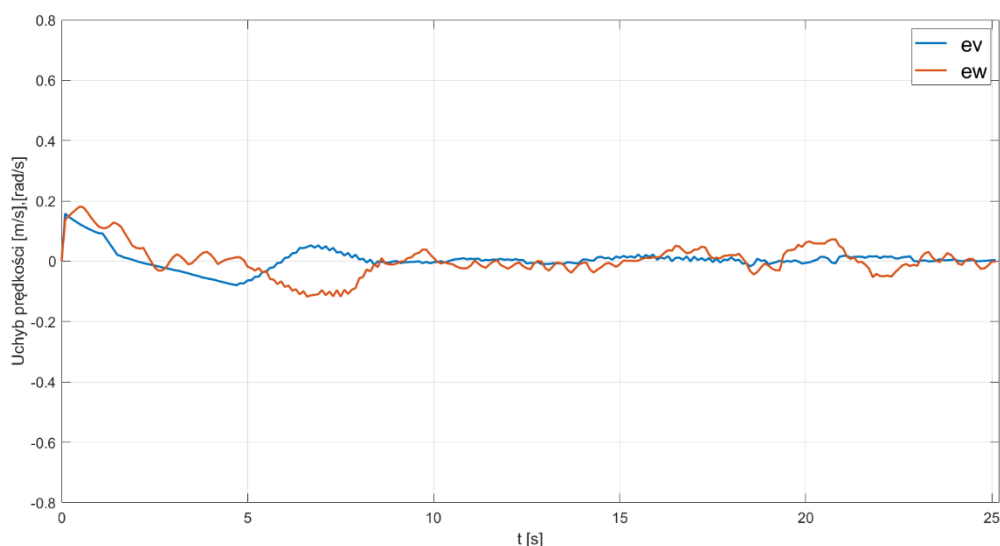
Rys. 6.45. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.46. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.47. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego



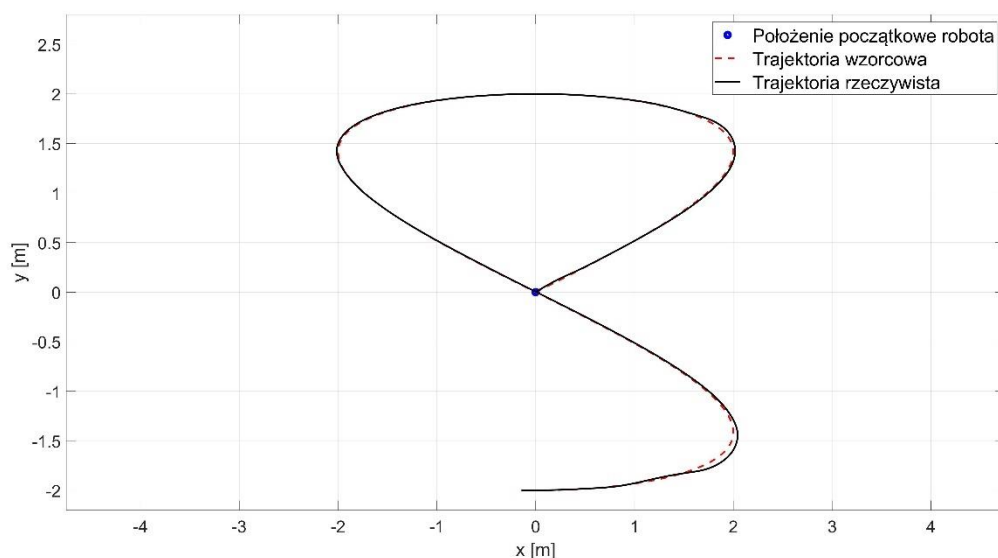
Rys. 6.48. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności śledzenia, można stwierdzić, że kontroler kinematyczno-dynamiczny realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 6.47) oraz prędkości (Rys. 6.48) nie większymi niż:

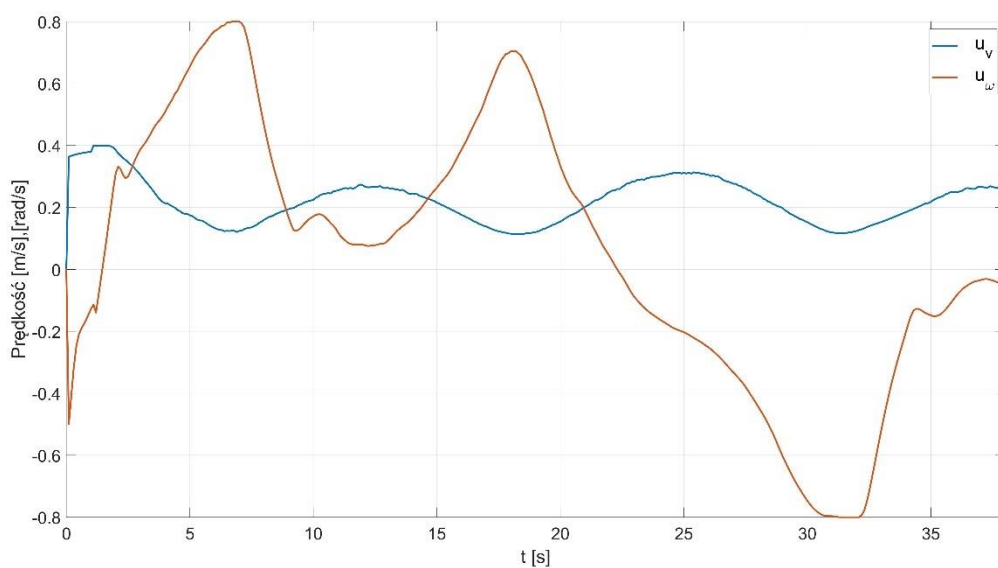
- w kierunku osi x : $e_x = 0.002$ m;
- w kierunku osi y : $e_y = 0.006$ m;
- orientacji: $e_\theta = 0.002$ rad;
- prędkości liniowej $e_v = 0.004 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.002 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 2.7$ s.

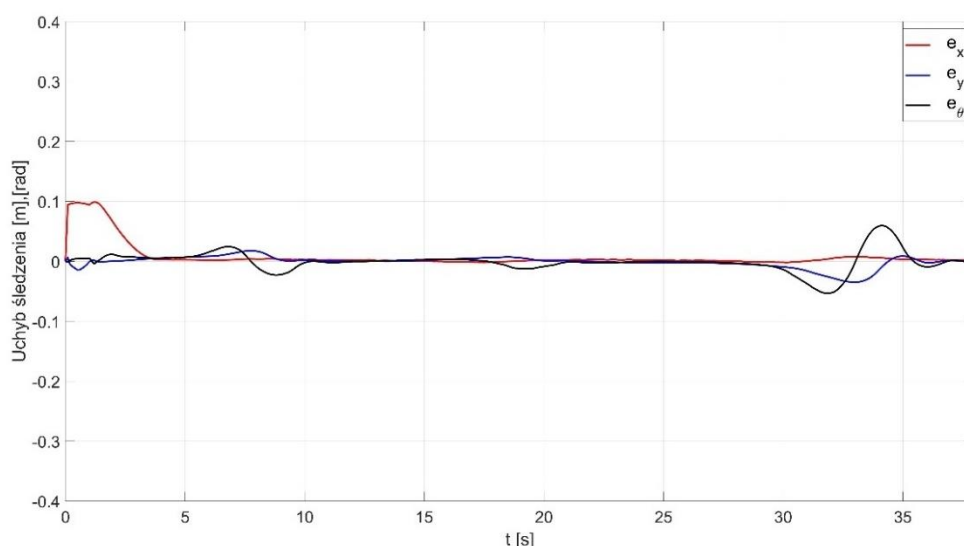
- Śledzenie trajektorii „ósemka”



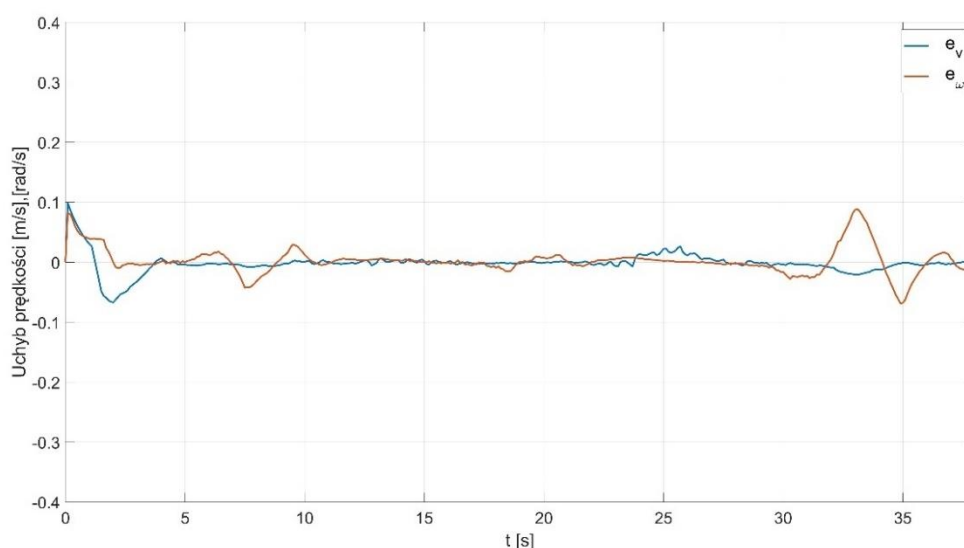
Rys. 6.49. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematycznego i dynamicznego



Rys. 6.50. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_w w badaniu laboratoryjnym śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.51. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego



Rys. 6.52. Uchyby prędkości liniowej e_v i kątowej e_ω w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności śledzenia, można stwierdzić, że kontroler kinematyczno-dynamicznego realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 6.51) oraz prędkości (Rys. 6.52) nie większymi niż:

- w kierunku osi x: $e_x = 0.004$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.006$ rad;
- prędkości liniowej $e_v = 0.046 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.013 \frac{\text{rad}}{\text{s}}$.

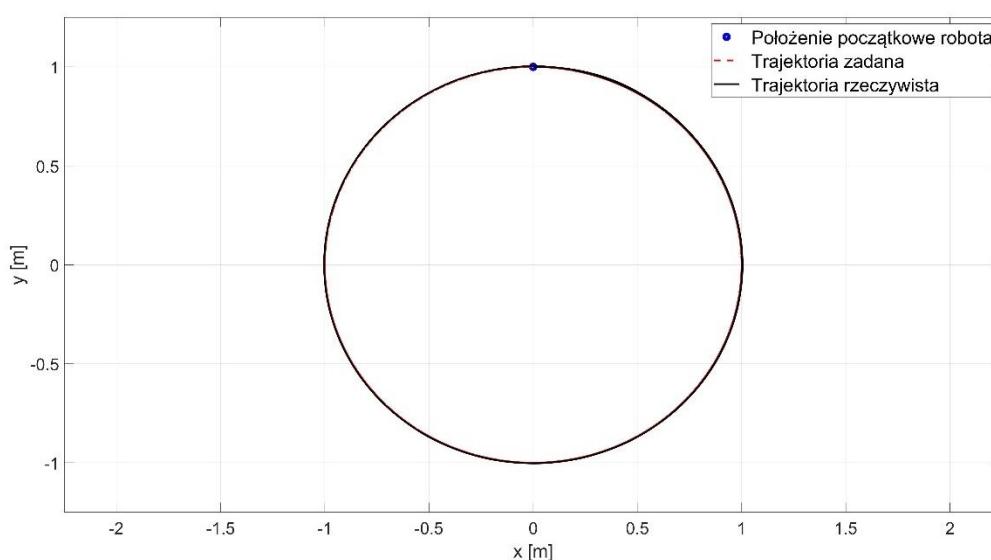
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 2.6$ s.

6.2.3. Badania kontrolera kinematyczno-dynamicznego z adaptacją

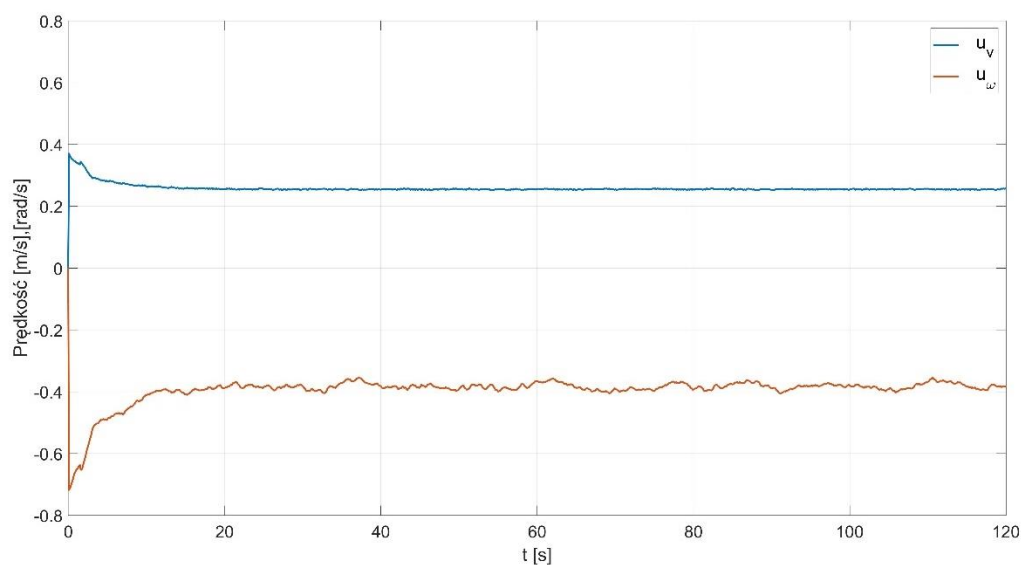
Wyniki badań doświadczalnych śledzenia trajektorii wzorcowej za pomocą kontrolera kinematyczno-dynamicznego z adaptacją przedstawiono na Rys. 6.53 - Rys. 6.76. Badania przeprowadzono zgodnie ze schematem przedstawionym na Rys. 6.3.

6.2.3.1. Kryterium ruchu przez określony czas dla kontrolera kinematyczno-dynamicznego z adaptacją

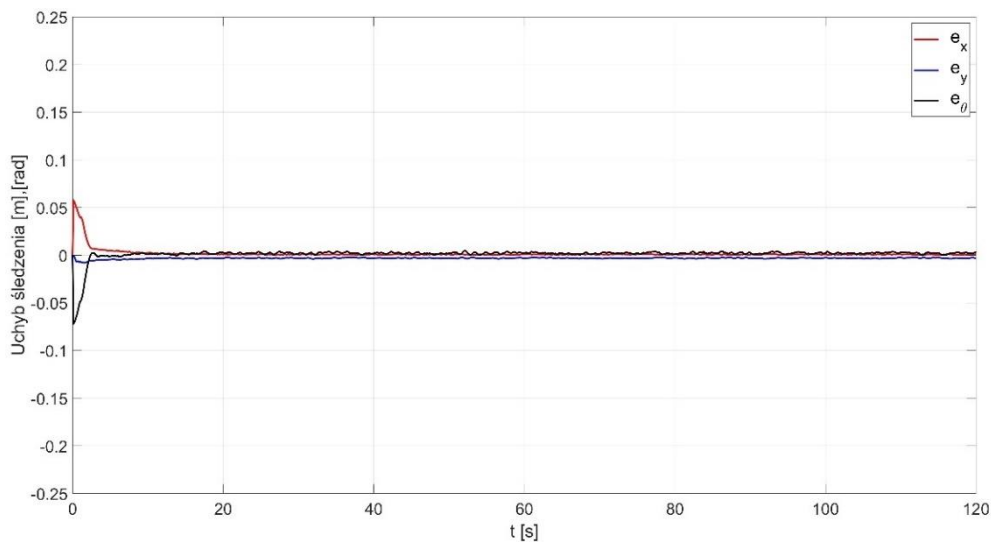
- Śledzenie trajektorii „okrąg”



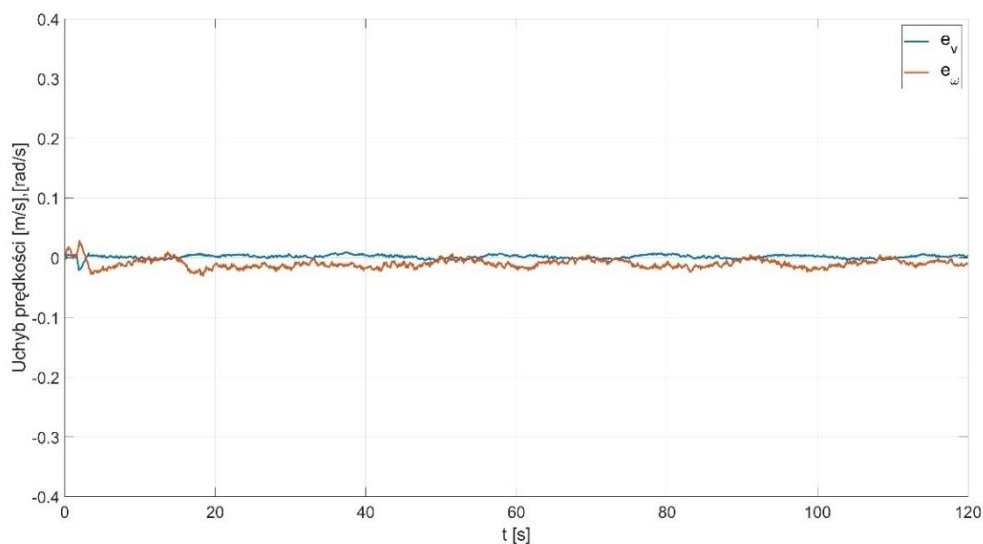
Rys. 6.53. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.54. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w badaniu laboratoryjnym śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.55. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



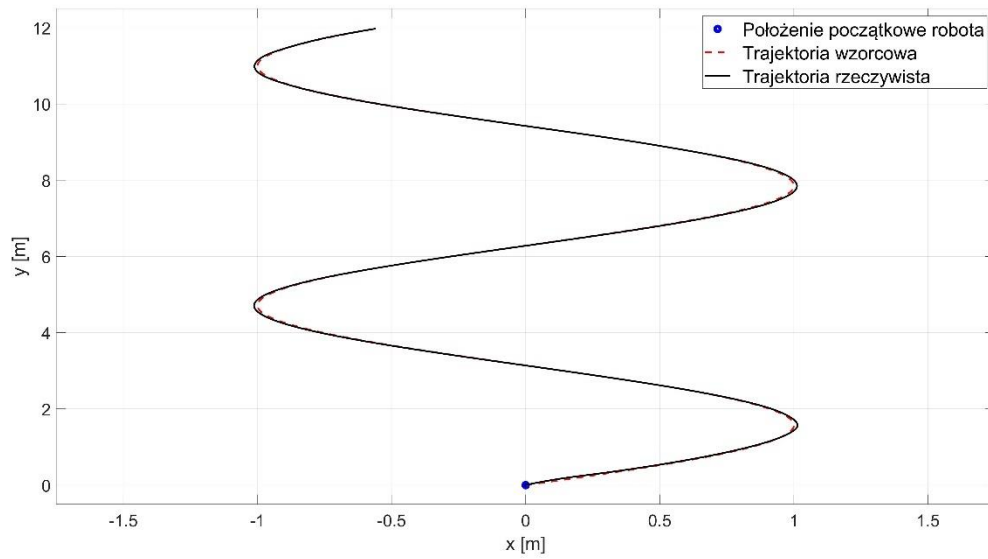
Rys. 6.56. Uchyby prędkości liniowej e_v i kątowej e_ω w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamicznego z adaptacją realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 6.55) oraz prędkości (Rys. 6.56) nie większymi niż:

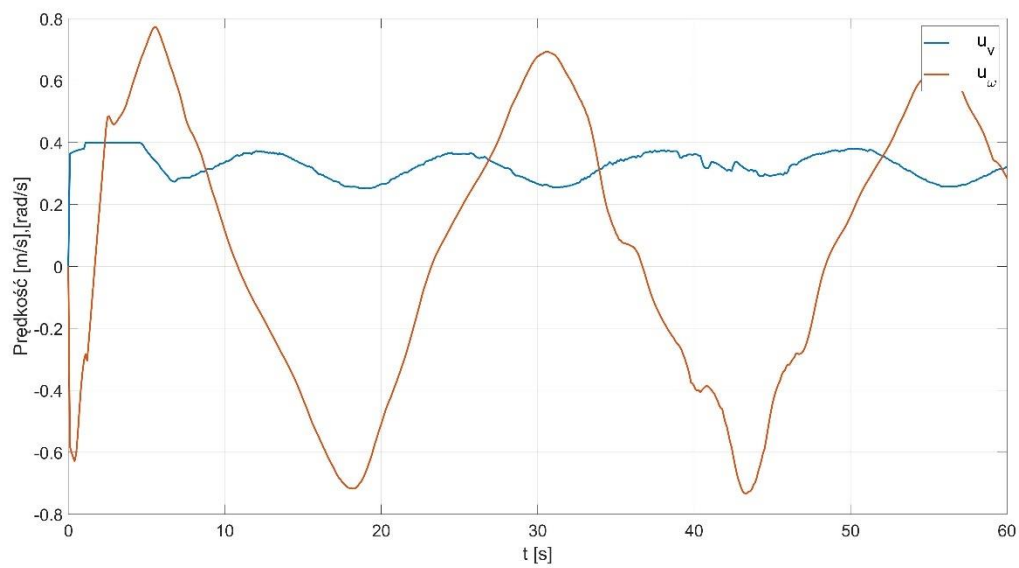
- w kierunku osi x: $e_x = 0.001$ m;
- w kierunku osi y: $e_y = 0.003$ m;
- orientacji: $e_\theta = 0.005$ rad;
- prędkości liniowej $e_v = 0.007 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.023 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 2.4$ s.

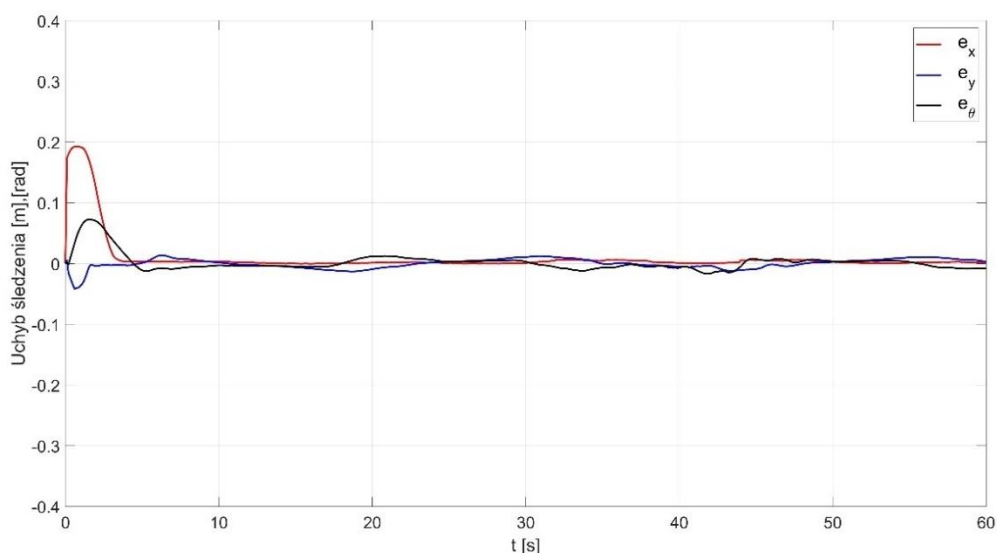
- Śledzenie trajektorii „sinus”



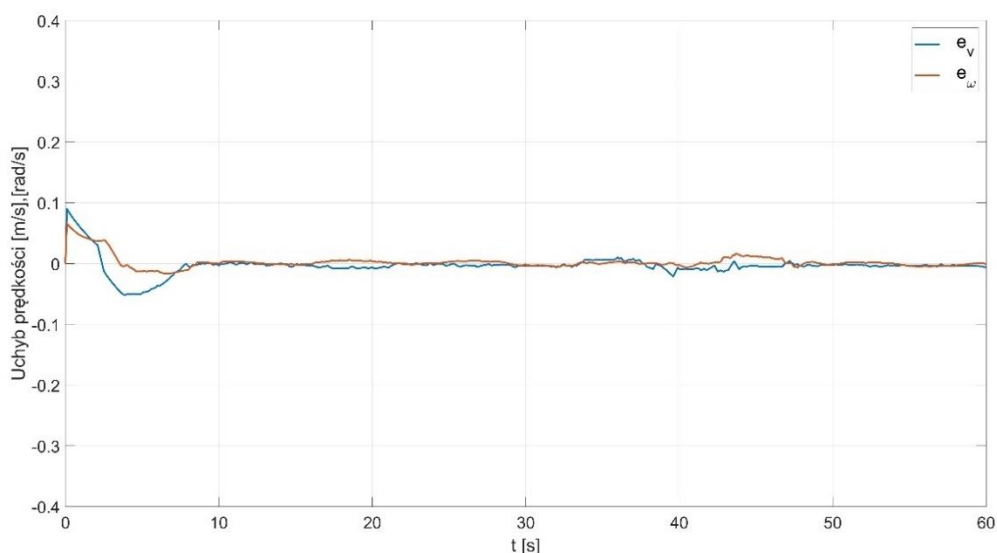
Rys. 6.57. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.58. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_{ω} w badaniu laboratoryjnym śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.59. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



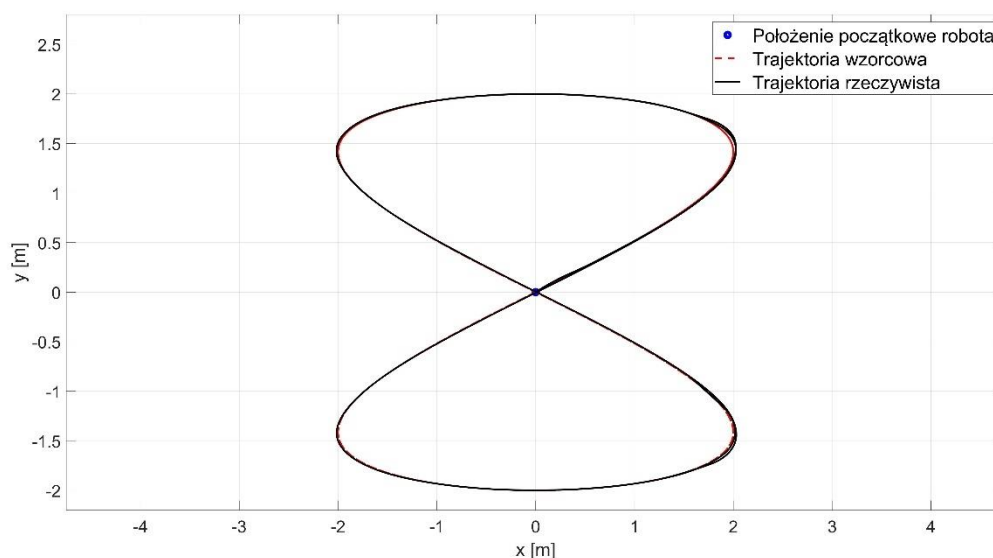
Rys. 6.60. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamiczny z adaptacją realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 6.59) oraz prędkości (Rys. 6.60) nie większymi niż:

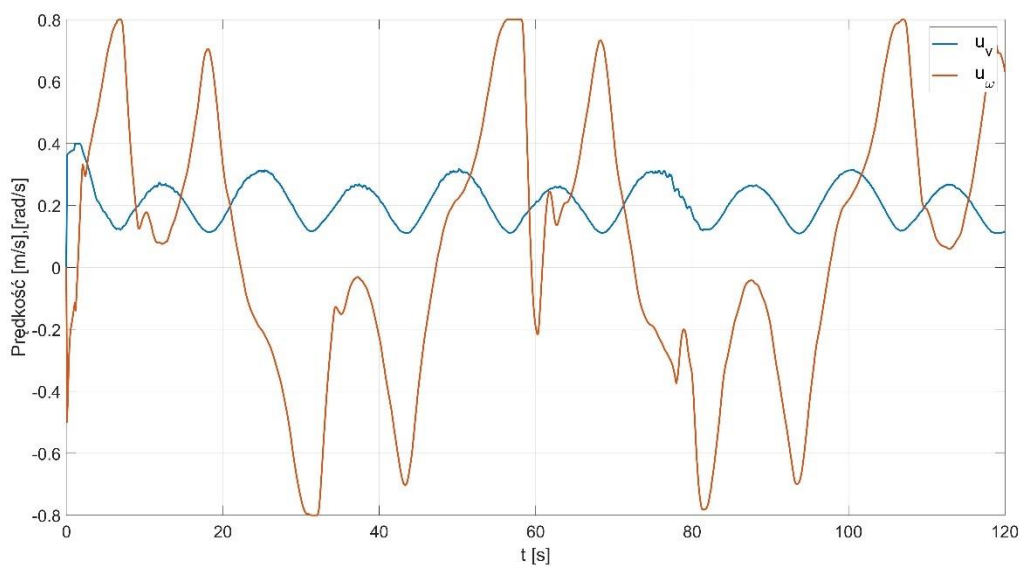
- w kierunku osi x: $e_x = 0.006$ m;
- w kierunku osi y: $e_y = 0.013$ m;
- orientacji: $e_\theta = 0.017$ rad;
- prędkości liniowej $e_v = 0.007 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.008 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 2.9$ s.

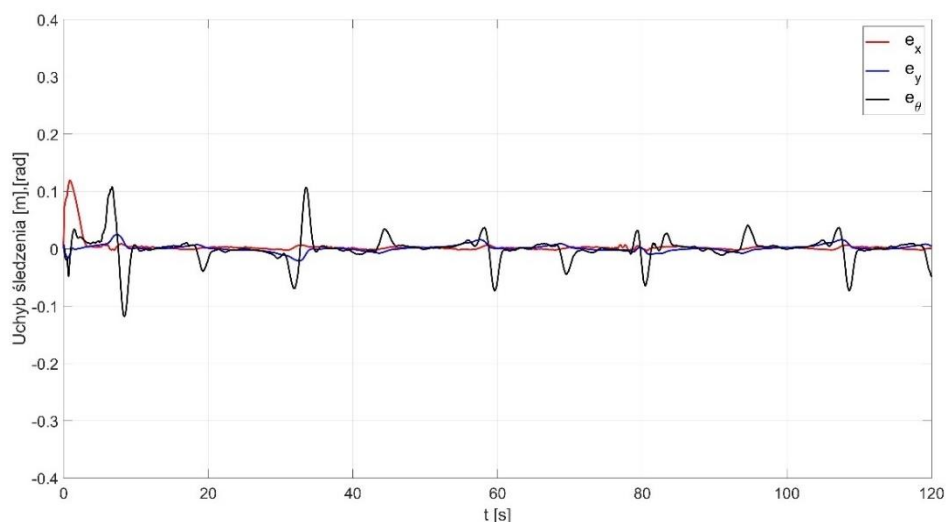
- Śledzenie trajektorii „ósemka”



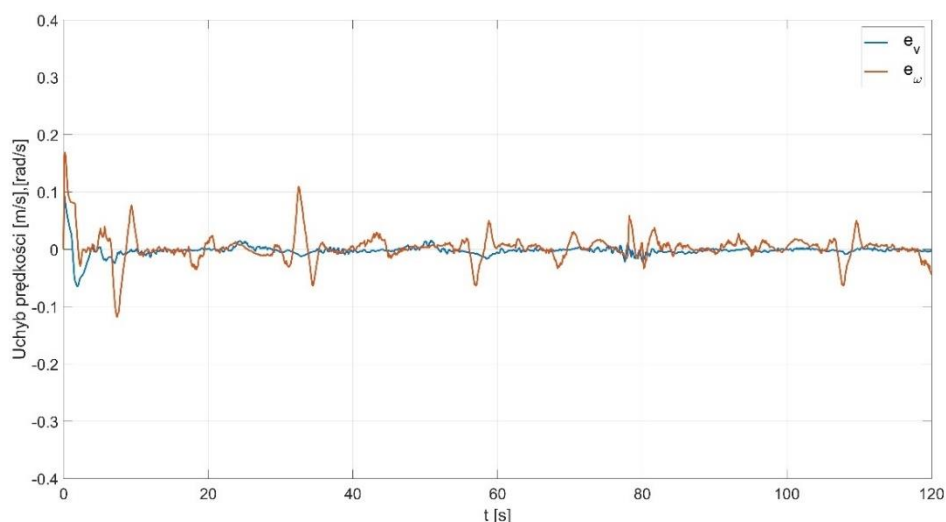
Rys. 6.61. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego stabilności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.62. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.63. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.64. Uchyby prędkości liniowej e_v i kątowej e_ω w badaniu laboratoryjnym stabilności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

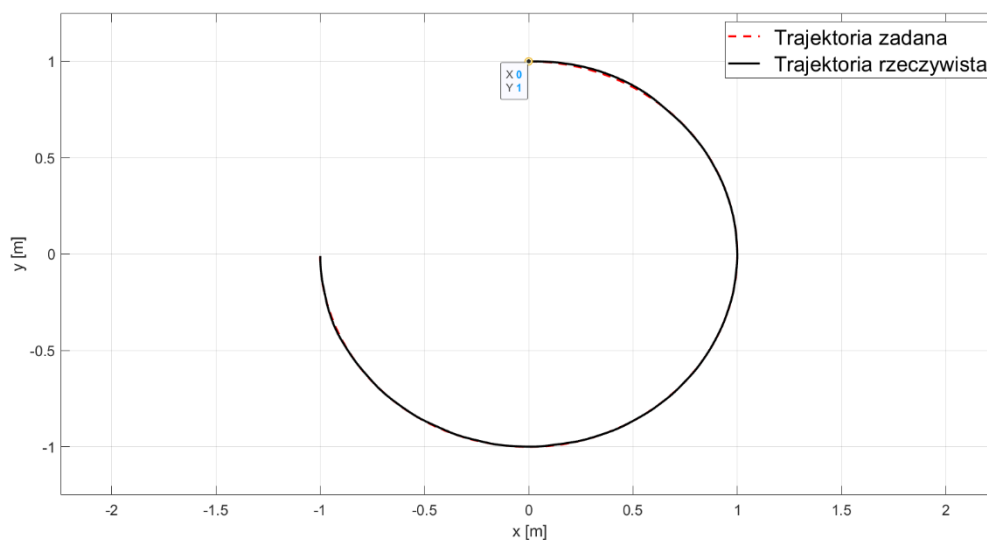
Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu stabilności, można stwierdzić, że kontroler kinematyczno-dynamicznego z adaptacją realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 6.63) oraz prędkości (Rys. 6.64) nie większymi niż:

- w kierunku osi x: $e_x = 0.006$ m;
- w kierunku osi y: $e_y = 0.016$ m;
- orientacji: $e_\theta = 0.107$ rad;
- prędkości liniowej $e_v = 0.012 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.101 \frac{\text{rad}}{\text{s}}$.

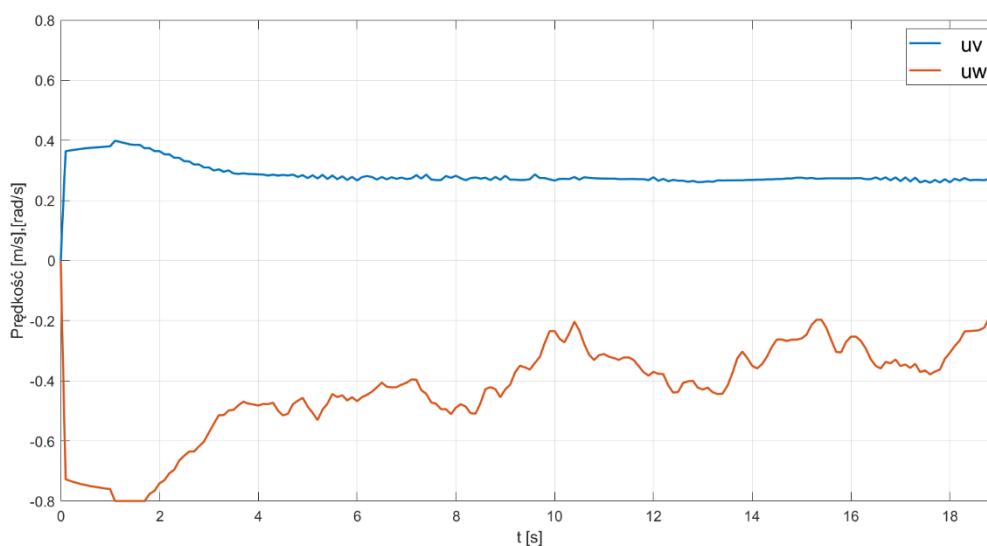
Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 1.5$ s.

6.2.3.2. Kryterium ruchu po skończonej drodze dla kontrolera kinematyczno-dynamicznego z adaptacją

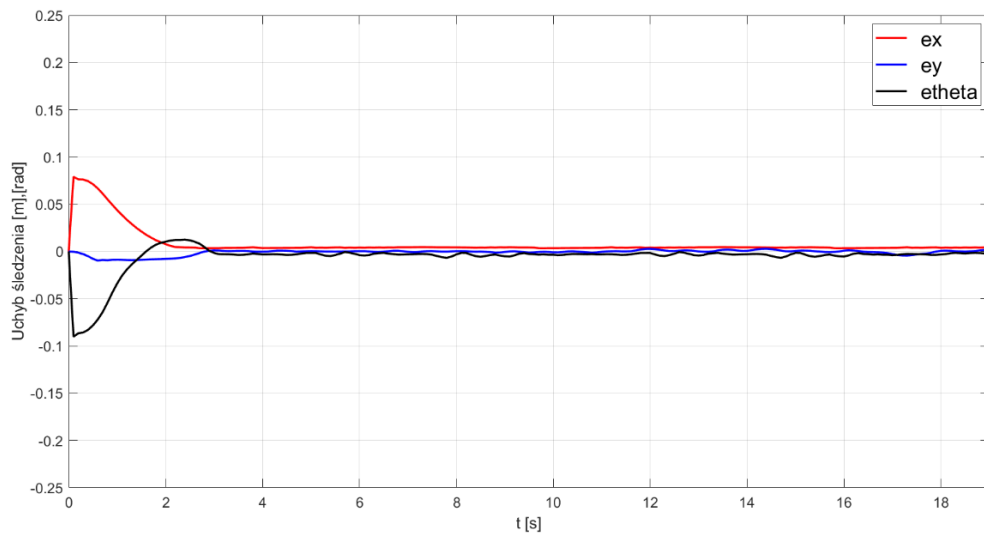
- Śledzenie trajektorii „okrąg”



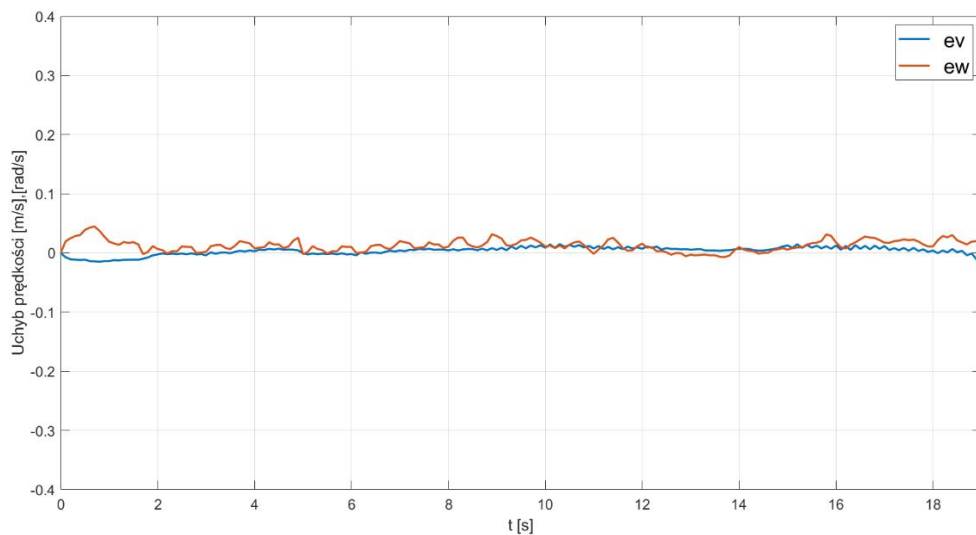
Rys. 6.65. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.66. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_w w badaniu laboratoryjnym śledzenia trajektorii „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.67. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



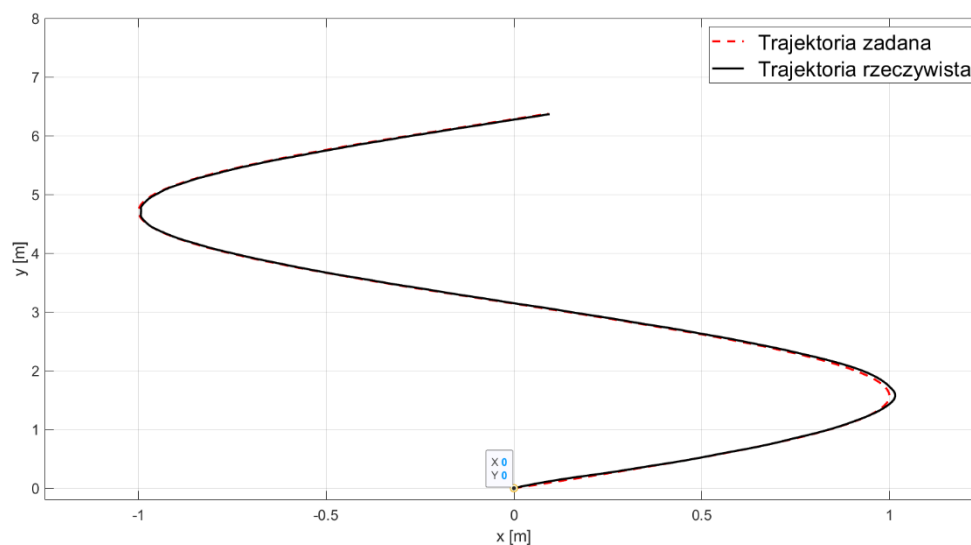
Rys. 6.68. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „okrąg” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności śledzenia, można stwierdzić, że kontroler kinematyczno-dynamicznego z adaptacją realizuje zadanie śledzenia trajektorii „okrąg” z uchybami pozycji i orientacji (Rys. 6.67) oraz prędkości (Rys. 6.68) nie większymi niż:

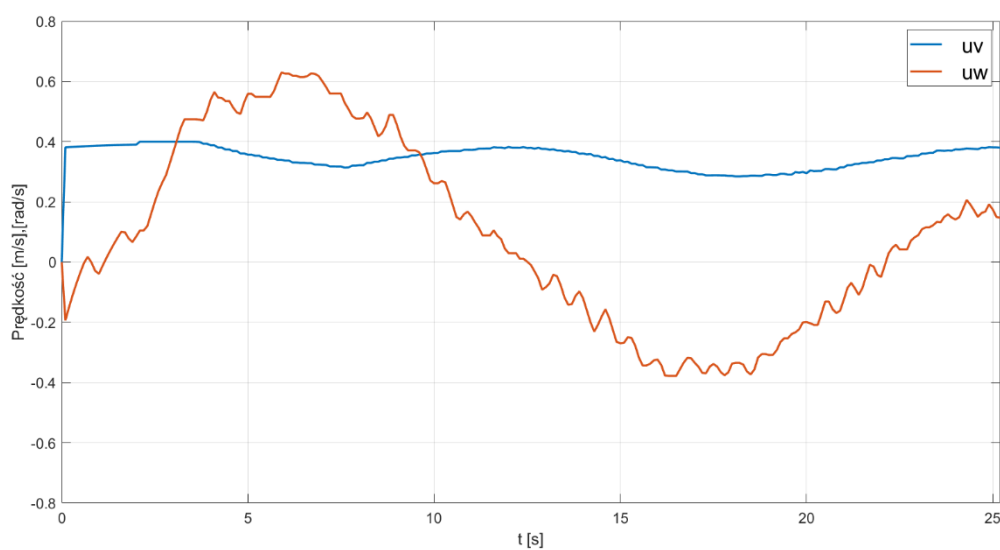
- w kierunku osi x: $e_x = 0.004$ m;
- w kierunku osi y: $e_y = 0.002$ m;
- orientacji: $e_\theta = 0.003$ rad;
- prędkości liniowej $e_v = 0.009 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.021 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „okrąg” jest równy $t_r \approx 1.9$ s.

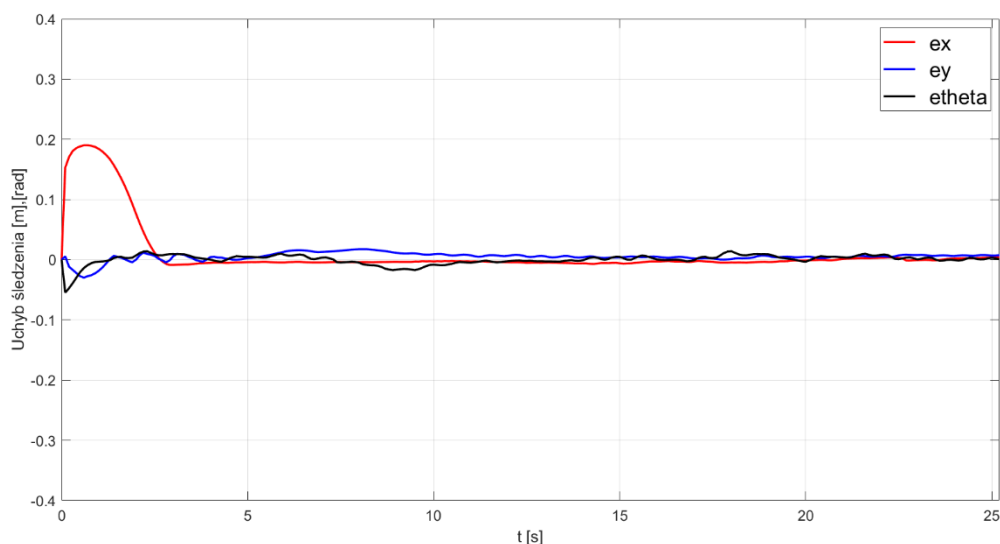
- Śledzenie trajektorii „sinus”



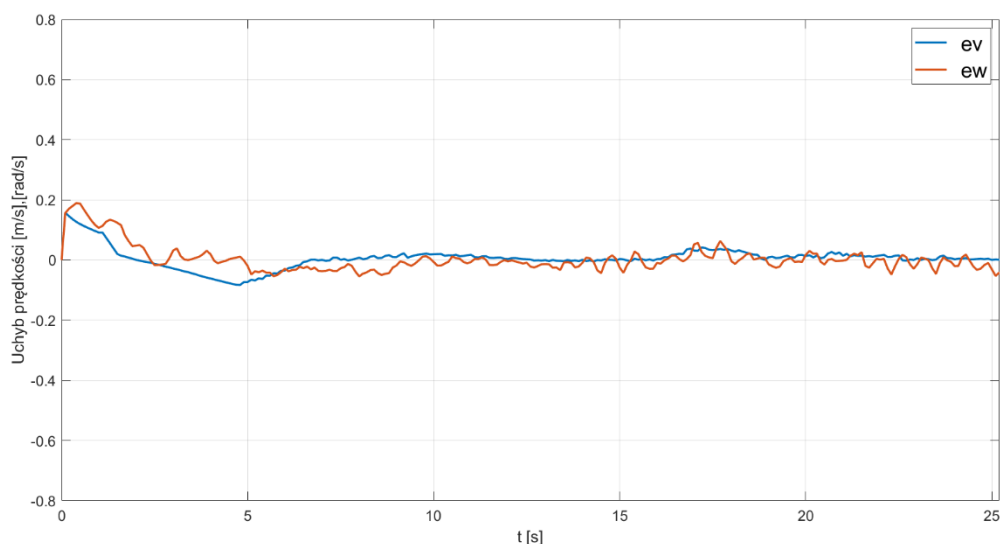
Rys. 6.69. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.70. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.71. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



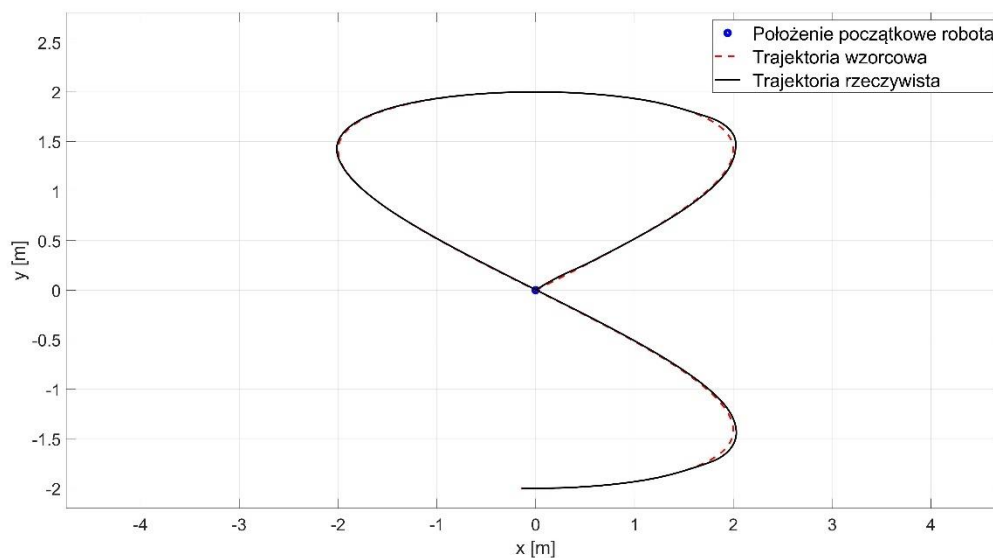
Rys. 6.72. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „sinus” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności śledzenia, można stwierdzić, że kontroler kinematyczno-dynamicznego z adaptacją realizuje zadanie śledzenia trajektorii „sinus” z uchybami pozycji i orientacji (Rys. 6.71) oraz prędkości (Rys. 6.72) nie większymi niż:

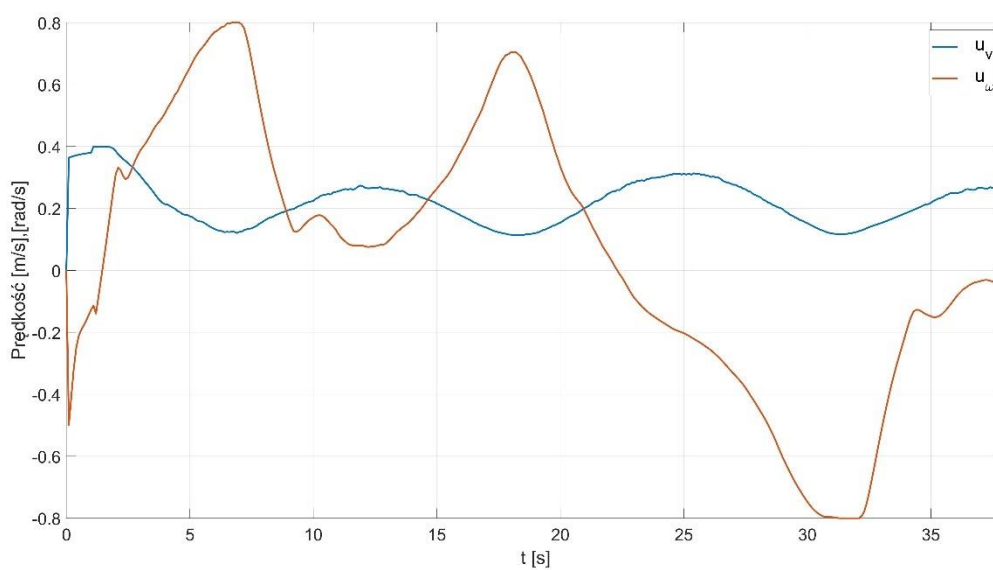
- w kierunku osi x: $e_x = 0.003$ m;
- w kierunku osi y: $e_y = 0.002$ m;
- orientacji: $e_\theta = 0.002$ rad;
- prędkości liniowej $e_v = 0.002 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_\omega = 0.011 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „sinus” jest równy $t_r \approx 2.3$ s.

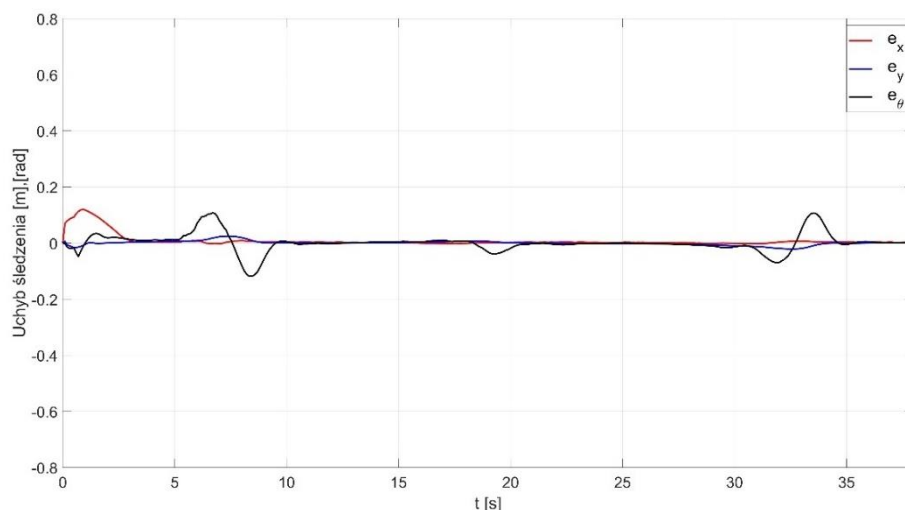
- Śledzenie trajektorii „ósemka”



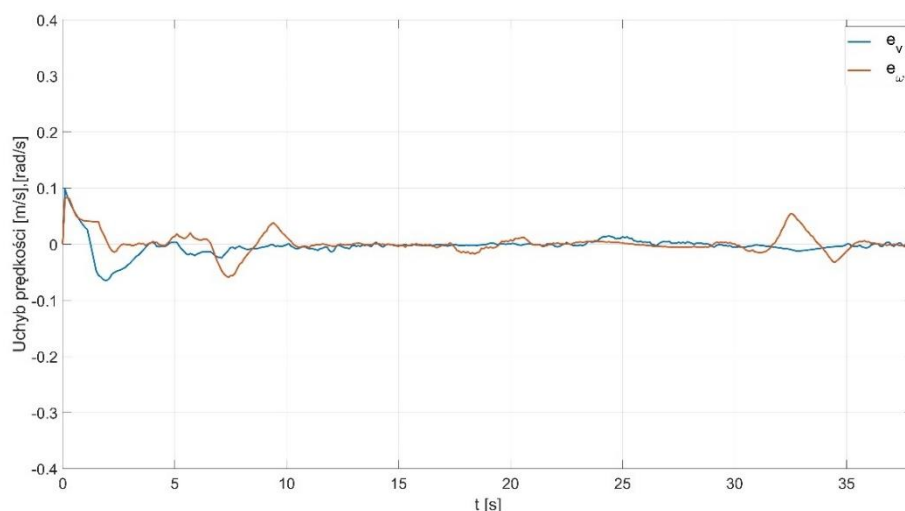
Rys. 6.73. Porównanie trajektorii wzorcowej i otrzymanej w wyniku badania laboratoryjnego dokładności śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.74. Przebieg sygnałów sterujących: prędkości liniowej u_v i prędkości kątowej u_ω w badaniu laboratoryjnym śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.75. Uchyby śledzenia e_x , e_y , e_θ w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.76. Uchyby prędkości liniowej e_v i kątowej e_w w badaniu laboratoryjnym dokładności śledzenia trajektorii wzorcowej „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie analizy uchybów wyznaczonych doświadczalnie w badaniu dokładności śledzenia, można stwierdzić, że kontroler kinematyczno-dynamicznego z adaptacją realizuje zadanie śledzenia trajektorii „ósemka” z uchybami pozycji i orientacji (Rys. 6.75) oraz prędkości (Rys. 6.76) nie większymi niż:

- w kierunku osi x: $e_x = 0.004$ m;
- w kierunku osi y: $e_y = 0.001$ m;
- orientacji: $e_\theta = 0.001$ rad;
- prędkości liniowej $e_v = 0.002 \frac{\text{m}}{\text{s}}$;
- prędkości kątowej $e_w = 0.008 \frac{\text{rad}}{\text{s}}$.

Czas regulacji dla przypadku zadania śledzenia trajektorii „ósemka” jest równy $t_r \approx 1.9$ s.

6.2.4. Podsumowanie badań doświadczalnych warstwy wewnętrznej W2 algorytmu sterowania

W badaniach doświadczalnych sprawdzono jakość sterowania warstwy wewnętrznej W2 struktury przedstawionej na Rys. 4.1 dla grupy trzech rzeczywistych robotów TURTLEBOT 2. W celu porównania wyników teoretycznych z uzyskanymi w badaniach doświadczalnych przyjęto takie same kryteria i warunki początkowe jak w badaniach teoretycznych. Na podstawie badań można stwierdzić, że zaproponowane kontrolery nakierowują robota na zadaną trajektorię, a jej kształt jest odwzorowany z bardzo dużą dokładnością. Wartości uchybów wyznaczonych w badaniu doświadczalnym śledzenia trajektorii zgodnie z kryterium określonego czasu ruchu zestawiono w Tabela 17. Wartości uchybów wyznaczonych w badaniu doświadczalnym śledzenia trajektorii zgodnie z kryterium ruchu po określonej drodze zestawiono w Tabela 18

Potwierdzona została także teza, że optymalną konfiguracją warstwy wewnętrznej W2 algorytmu sterowania jest zastosowanie kontrolera kinematyczno-dynamicznego z adaptacją. Dla tej konfiguracji układu sterowania w stosunku do pozostałych uzyskano:

- dla trajektorii okrąg, w stosunku do uchybów wyznaczonych dla kontrolera kinematycznego otrzymano wartości mniejsze o: 91% dla e_x , 67% dla e_y , 85% dla e_θ , 56% dla e_v , 26% dla e_ω , 78% dla t_r ; w stosunku do uchybów wyznaczonych dla kontrolera kinematyczno-dynamicznego otrzymano wartości mniejsze o: 83% dla e_x , 55% dla e_θ , 56% dla e_v , 36% dla e_ω , 27% dla t_r ; dla e_y uzyskano wartość większą o 300%;
- dla trajektorii sinus, w stosunku do uchybów wyznaczonych dla kontrolera kinematycznego otrzymano wartości mniejsze o: 81% dla e_x , 54% dla e_y , 67% dla e_θ , 13% dla e_v , 43% dla e_ω , 61% dla t_r ; w stosunku do uchybów wyznaczonych dla kontrolera kinematyczno-dynamicznego otrzymano wartości mniejsze o: 65% dla e_x , 13% dla e_y , 48% dla e_θ , 77% dla e_v , 47% dla e_ω , 47% dla t_r ;
- dla trajektorii ósemka, w stosunku do uchybów wyznaczonych dla kontrolera kinematycznego otrzymano wartości mniejsze o: 71% dla e_x , 83% dla e_y , 63% dla e_θ , 72% dla e_v , 64% dla e_ω , 55% dla t_r ; w stosunku do uchybów wyznaczonych dla kontrolera kinematyczno-dynamicznego otrzymano wartości mniejsze o: 63% dla e_x , 63% dla e_y , 328% dla e_θ , 59% dla e_v , 57% dla e_ω , 32% dla t_r ;

Tabela 17. Wartości uchybów wyznaczonych w badaniu doświadczalnym śledzenia trajektorii zgodnie z kryterium ruchu przez określony czas

		Kontroler kinematyczny	Kontroler kinematyczno-dynamiczny	Kontroler kinematyczno-dynamiczny z adaptacją
Trajektoria „okrąg”	e_x [m]	0.012	0.006	0.001
	e_y [m]	0.009	0.001	0.003
	e_θ [rad]	0.033	0.011	0.005
	e_v $\left[\frac{m}{s}\right]$	0.016	0.017	0.007
	e_ω $\left[\frac{rad}{s}\right]$	0.031	0.036	0.023
	t_r [s]	11	3.3	2.4
Trajektoria „sinus”	e_x [m]	0.031	0.017	0.006
	e_y [m]	0.028	0.015	0.013
	e_θ [rad]	0.051	0.033	0.017
	e_v $\left[\frac{m}{s}\right]$	0.008	0.030	0.007
	e_ω $\left[\frac{rad}{s}\right]$	0.014	0.015	0.008
	t_r [s]	7.4	5.5	2.9
Trajektoria „ósemka”	e_x [m]	0.021	0.016	0.006
	e_y [m]	0.095	0.043	0.016
	e_θ [rad]	0.291	0.025	0.107
	e_v $\left[\frac{m}{s}\right]$	0.043	0.029	0.012
	e_ω $\left[\frac{rad}{s}\right]$	0.277	0.237	0.101
	t_r [s]	3.3	2.2	1.5

Tabela 18. Wartości uchybów wyznaczonych w badaniu doświadczalnym śledzenia trajektorii zgodnie z kryterium ruchu po skończonej drodze

		Kontroler kinematyczny	Kontroler kinematyczno-dynamiczny	Kontroler kinematyczno-dynamiczny z adaptacją
Trajektoria „okrag”	e_x [m]	0.023	0.021	0.004
	e_y [m]	0.005	0.004	0.002
	e_θ [rad]	0.021	0.016	0.003
	e_v $\left[\frac{m}{s}\right]$	0.003	0.013	0.009
	e_ω $\left[\frac{rad}{s}\right]$	0.012	0.029	0.021
	t_r [s]	3.6	2.5	1.9
Trajektoria „sinus”	e_x [m]	0.006	0.002	0.003
	e_y [m]	0.021	0.006	0.002
	e_θ [rad]	0.021	0.002	0.002
	e_v $\left[\frac{m}{s}\right]$	0.005	0.004	0.002
	e_ω $\left[\frac{rad}{s}\right]$	0.025	0.002	0.011
	t_r [s]	5.9	2.7	2.3
Trajektoria „ósemka”	e_x [m]	0.005	0.004	0.004
	e_y [m]	0.030	0.001	0.001
	e_θ [rad]	0.013	0.006	0.001
	e_v $\left[\frac{m}{s}\right]$	0.003	0.046	0.002
	e_ω $\left[\frac{rad}{s}\right]$	0.238	0.013	0.008
	t_r [s]	3.2	2.6	1.9

6.3. Zakres i metodyka badań doświadczalnych warstwy zewnętrznej W1 algorytmu sterowania

Badania warstwy zewnętrznej W1 systemu sterowania przedstawionego na Rys. 4.1 przeprowadzono w celu weryfikacji wyników otrzymanych w badaniach teoretycznych. Podczas badań doświadczalnych warstwy zewnętrznej W1 zastosowano takie same warunki początkowe i nastawy kontrolerów jak w badaniach teoretycznych. Również rozpatrzono dwa przypadki (warianty) badań:

- synchronizacja rozpoczęcia ruchu przez grupę,
- synchroniczny ruch grupy po zadanej trajektorii.

W badaniach warstwy zewnętrznej W1 zastosowano konfigurację układu sterowania warstwy W2 przedstawioną na Rys. 6.3, obejmującą połączenie kontrolera kinematycznego (4.18) i dynamicznego (4.28) z adaptacją (2.22), która wykazała najlepsze rezultaty sterowania w badaniach przedstawionych w punkcie 6.2 oraz trzy roboty TURTLEBOT 2 tworzące grupę (Rys. 2.4). W celu oceny jakości działania kontrolera grupy opracowanego zgodnie ze schematem przedstawionym na Rys. 4.4 i opisanym w rozdziale 4.3, badania przeprowadzono w dwóch wariantach. W pierwszej kolejności przeprowadzono badanie bez uwzględnienia warstwy W2, czyli kontrolera grupy. Następnie to samo badanie powtórzono dla kompletnej konfiguracji algorytmu sterowania obejmującej wszystkie elementy przedstawione na Rys. 4.1.

Komunikację pomiędzy robotami zrealizowano za pomocą Wi-Fi w trybie Ad-Hoc, który posiada zdecentralizowaną strukturę, a przyłączane do niego urządzenia mogą pełnić zarówno rolę klienta jak i punktu dostępowego. Komunikacja pomiędzy klientami podłączonymi do sieci następuje w sposób bezpośredni tzn. pakiety dostarczane są do klientów bez konieczności istnienia węzłów zarządzających komunikacją jak przedstawiono na Rys. 3.2. Do realizacji programowej komunikacji wykorzystano bibliotekę *ad_hoc_communication* platformy ROS.

Odporność algorytmu sterowania na opóźnienia czasowe zbadano na podstawie synchronizacji rozpoczęcia ruchu grupy trzech robotów po linii prostej. W badaniu tym, grupa robotów miała przejechać drogę równą 1m z prędkością liniową równą $v_d = 0.1 \frac{m}{s}$. W pierwszej kolejności przeprowadzono badanie bez uwzględnienia warstwy W2, czyli kontrolera grupy opisanego w rozdziale 4.3. Następnie to samo badanie powtórzono dla kompletnej konfiguracji algorytmu sterowania obejmującej wszystkie elementy przedstawione na Rys. 4.1.

6.4. Badania doświadczalne warstwy zewnętrznej W1 algorytmu sterowania

Badania doświadczalne algorytmu synchronicznego sterowania przeprowadzono w laboratorium w Instytucie Techniki Raketowej i Mechatroniki na specjalnie przygotowanym placu manewrowym przedstawionym na Rys. 6.4. Formację grupy ustalono jako linię prostą. Roboty oddalono od siebie o odległość 0.5 m, a ich położenia początkowe opisano wektorami współrzędnych konfiguracyjnych w postaci:

- Robot 1 - $h_1(t_0) = [0, 0, 0, 0, 0]$;
- Robot 2 - $h_2(t_0) = [0, 0.5, 0, 0, 0]$;
- Robot 3 - $h_3(t_0) = [0, -0.5, 0, 0, 0]$;

Położenie środka geometrycznego wirtualnej struktury C_{vc} ustalono w środku geometrycznym Robota 1. Orientację θ^{vc} układu współrzędnych wirtualnej struktury również ustalono zgodną z orientacją Robota 1.

Wyniki badań przedstawiono w formie wykresów:

- porównania trajektorii wzorcowej i rzeczywistej,
- uchybów śledzenia trajektorii wyznaczonych z zależności (6.5),
- uchybów odległości pomiędzy robotami wyznaczonych z zależności (6.6),
- wykresów prędkości rzeczywistej.

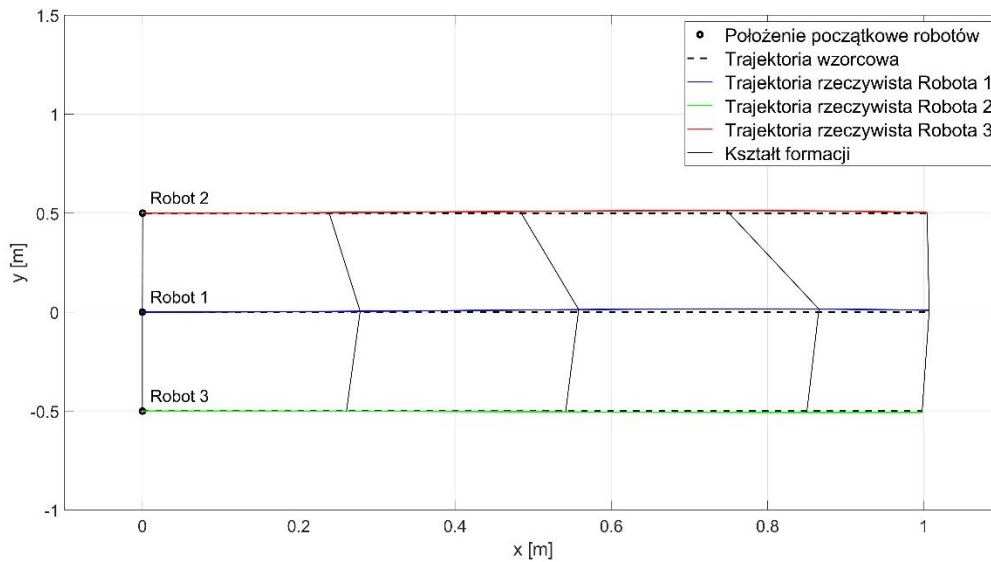
Jakość zaproponowanych układów regulacji oceniono na podstawie wartości uchybów opisanych wzorami (6.5) i (6.6).

6.4.1. Badania doświadczalne synchronizacji rozpoczęcia ruchu przez grupę robotów

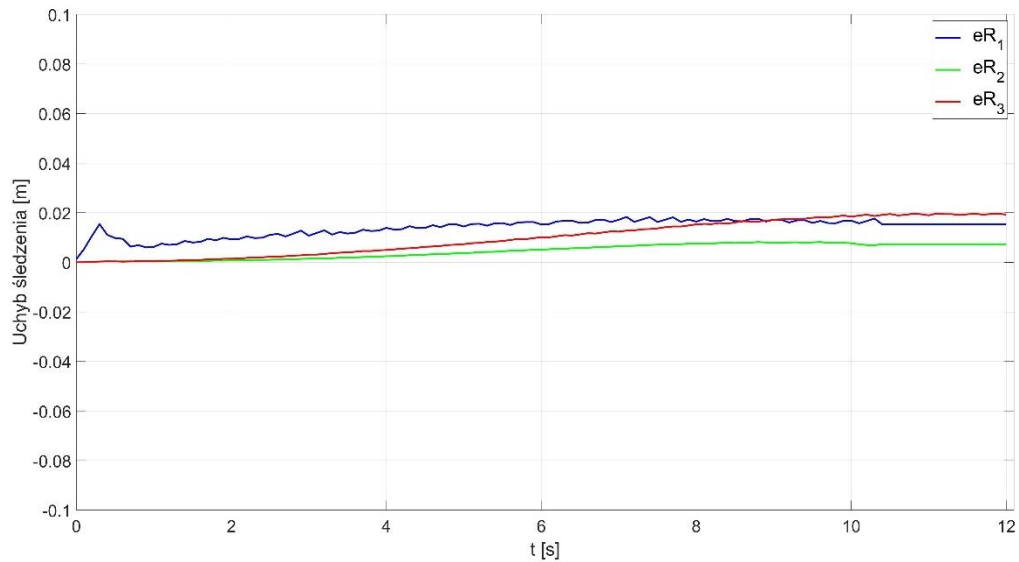
Wyniki badań symulacyjnych rozpoczęcia ruchu przez grupę robotów przedstawiono na Rys. 6.77 - Rys. 6.84. Badania przeprowadzono zgodnie z metodyką opisaną w rozdziale 6.3.

6.4.1.1. Badanie doświadczalne synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy

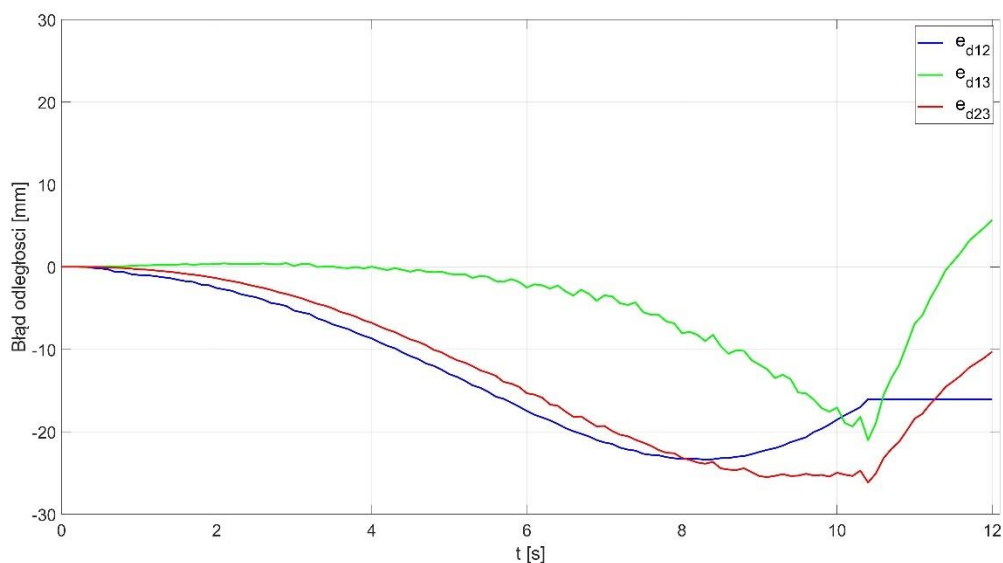
Badanie przeprowadzono dla grupy trzech robotów TURTLEBOT 2. W algorytmie sterowania nie uwzględniono kontrolera konfiguracji grupy przedstawionego w rozdziale 4.3. Wyniki badania przedstawiono na Rys. 6.77 - Rys. 6.80.



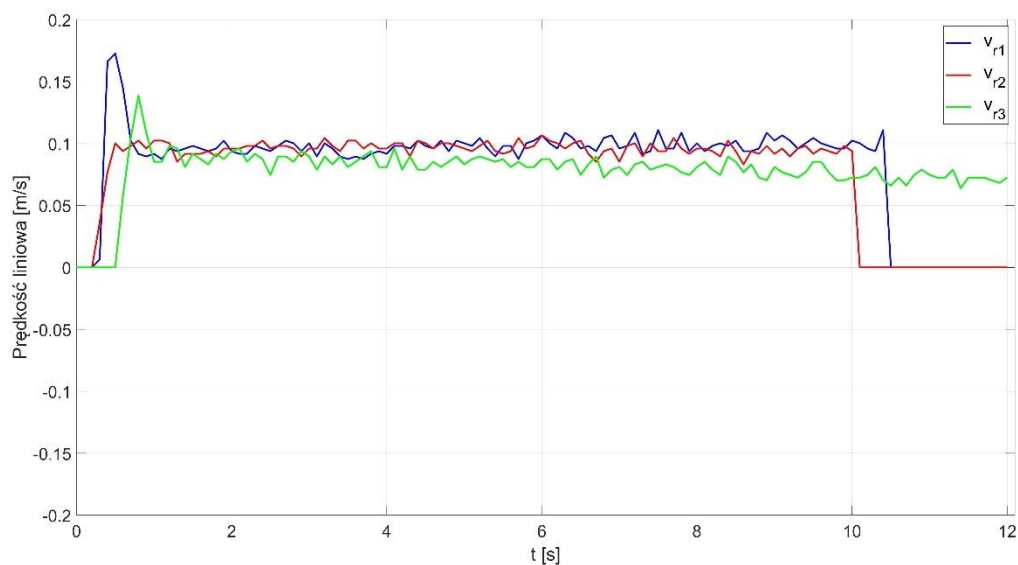
Rys. 6.77. Trajektorie rzeczywiste robotów w badaniu laboratoryjnym synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy



Rys. 6.78. Uchyby śledzenia trajektorii wzorcowych w badaniu laboratoryjnym synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy



Rys. 6.79. Błędy odległości pomiędzy robotami w badaniu laboratoryjnym synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy



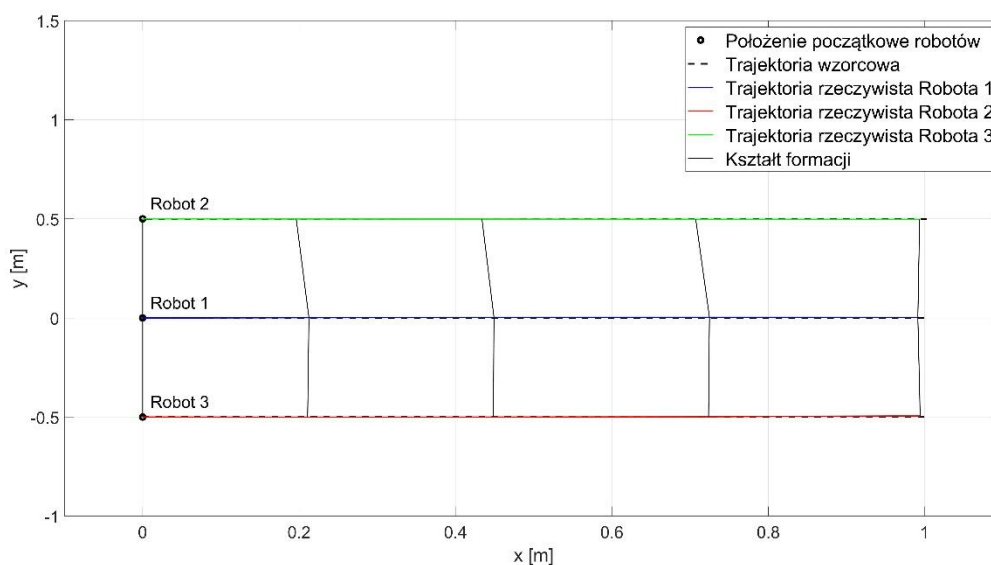
Rys. 6.80. Prędkości liniowe robotów wyznaczone doświadczalnie w badaniu synchronizacji rozpoczęcia ruchu bez uwzględnienia kontrolera konfiguracji grupy

W badaniu doświadczalnym synchronizacji rozpoczęcia ruchu przez grupę robotów bez uwzględnienia warstwy sterowania W1 roboty przejechały zadaną drogę równą 1m jednak konfiguracja grupy podczas ruchu nie była utrzymywana co jest widoczne na Rys. 6.77. Analizując wykres prędkości (Rys. 6.80) zauważono, że roboty poruszały się z zadeklarowaną prędkością, jednak moment startu był różny dla każdego robota, a w związku z tym również moment zatrzymania. Analizując wykresy uchybów śledzenia (Rys. 6.78) i uchybów odległości (Rys. 6.79) można zauważyć, że w tym przypadku priorytetem jest minimalizacja uchybów

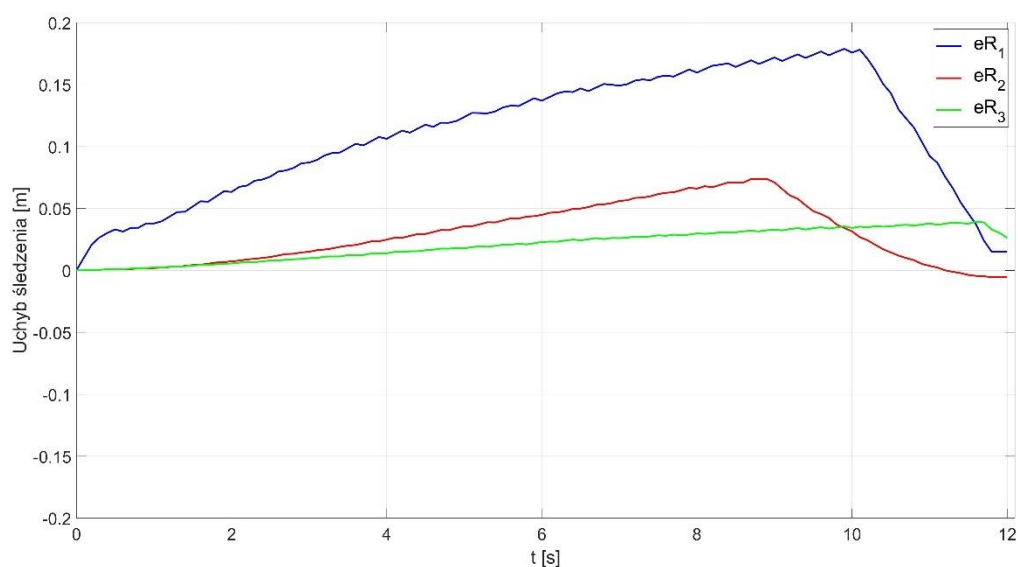
śledzenia, których wartość była nie większa niż $eR_i = 0.02$ m, kosztem dużych uchybów odległości o wartościach nie większych niż: $e_{d12} = 23$ mm, $e_{d13} = 21$ mm, $e_{d23} = 26$ mm, które nie były minimalizowane.

6.4.1.2. Badanie doświadczalne synchronizacji rozpoczęcia ruchu z uwzględnieniem kontrolera konfiguracji grupy

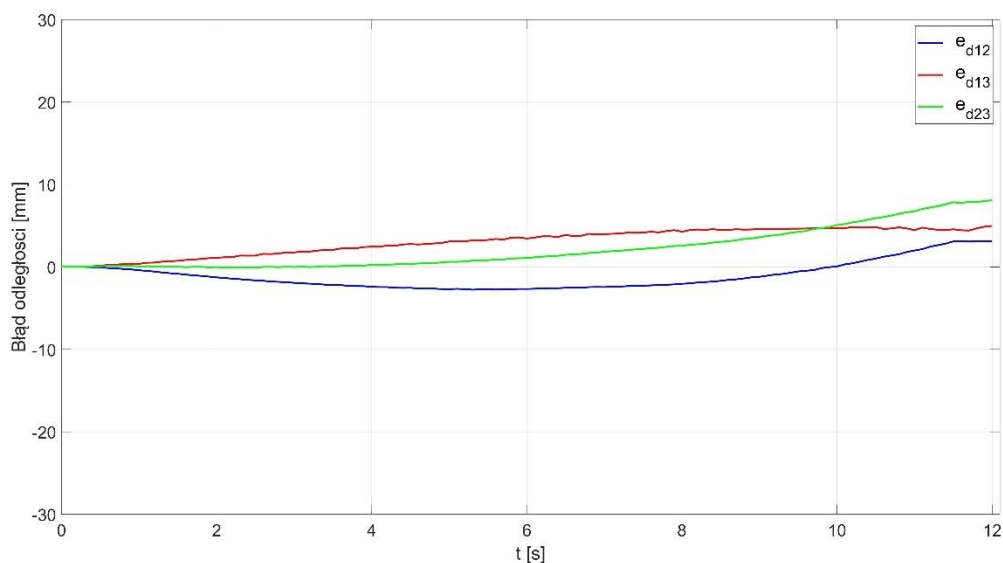
Badanie przeprowadzono dla grupy trzech robotów TURTLEBOT 2. W algorytmie sterowania uwzględniono kontroler konfiguracji grupy przedstawionego w rozdziale 4.3. Wyniki badania przedstawiono na Rys. 6.81 - Rys. 6.84.



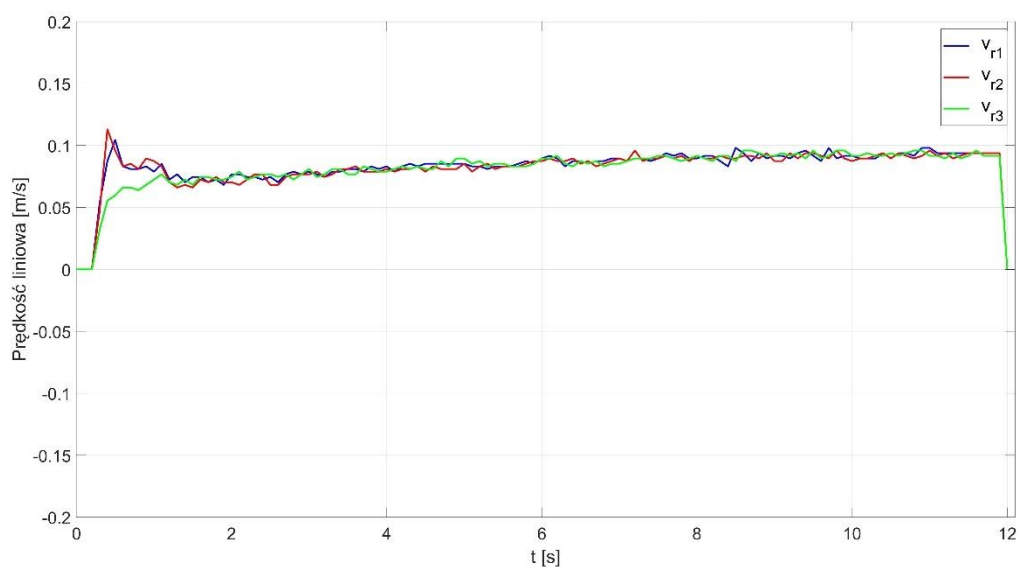
Rys. 6.81. Trajektorie rzeczywiste robotów w badaniu laboratoryjnym synchronicznego ruchu z uwzględnieniem kontrolera konfiguracji grupy



Rys. 6.82. Uchyby śledzenia trajektorii wzorcowych w badaniu laboratoryjnym synchronicznego ruchu z uwzględnieniem kontrolera konfiguracji grupy



Rys. 6.83. Błędy odległości pomiędzy robotami w badaniu laboratoryjnym synchronicznego ruchu z uwzględnieniem kontrolera konfiguracji grupy



Rys. 6.84. Prędkości liniowe robotów wyznaczone doświadczalnie w badaniu synchronicznego ruchu z uwzględnieniem kontrolera konfiguracji grupy

W badaniu doświadczalnym synchronizacji rozpoczęcia ruchu przez grupę robotów z uwzględnieniem warstwy sterowania W1 roboty przejechały zadaną drogę równą 1 m. Pomimo początkowego zaburzenia konfiguracji grupy warstwa sterowania W1 minimalizowała błędy odległości pomiędzy robotami i podczas ruchu zadana konfiguracja była utrzymywana, co jest widoczne na Rys. 6.81. Analizując wykres prędkości (Rys. 6.84) zauważono, że roboty poruszały się z zadeklarowaną prędkością oraz że nastąpił synchroniczny start i zatrzymanie się grupy. Dodatkowo widoczne fluktuacje przebiegów prędkości wskazują na działanie

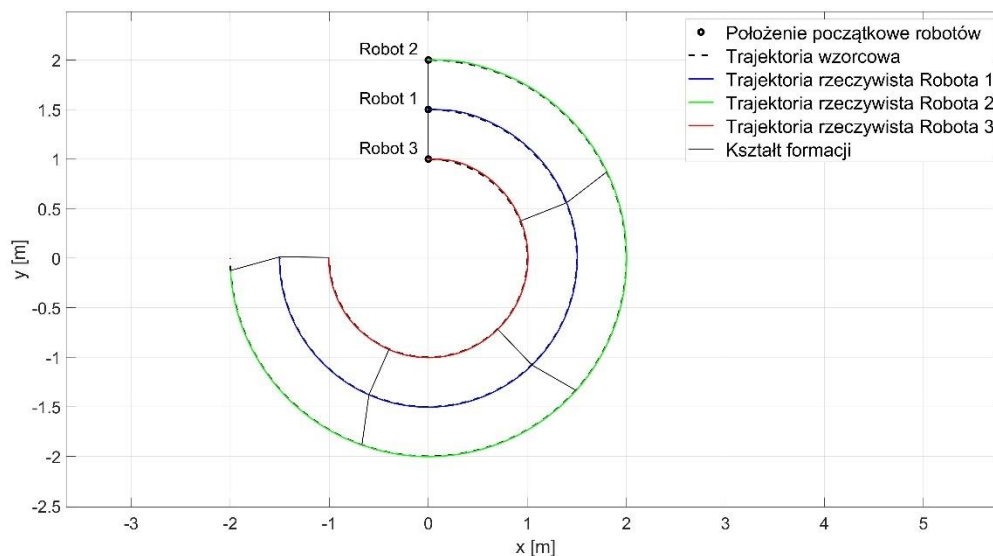
kontrolera grupy minimalizujące uchyb odległości pomiędzy robotami. Analizując wykresy uchybów śledzenia (Rys. 6.82) i uchybów odległości (Rys. 6.83) można zauważyć, że w tym przypadku priorytetem jest minimalizacja uchybów odległości pomiędzy robotami, których wartości były nie większe niż: $e_{d12} = 3.1 \text{ mm}$, $e_{d13} = 4.6 \text{ mm}$, $e_{d23} = 7.8 \text{ mm}$ kosztem dużych uchybów odległości o wartościach nie większych niż: $eR_1 = 0.18 \text{ m}$, $eR_2 = 0.07 \text{ m}$, $eR_3 = 0.04 \text{ mm}$, które nie były minimalizowane.

6.4.2. Badania doświadczalne synchronicznego ruchu grupy po zadanej trajektorii

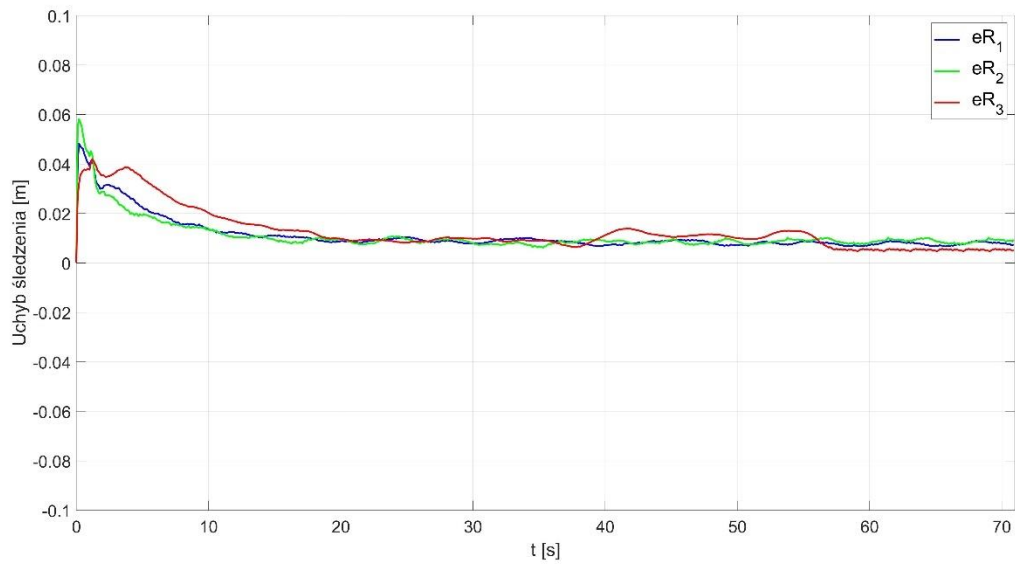
Wyniki badań doświadczalnych synchronicznego ruchu grupy po zadanej trajektorii przedstawiono na Rys. 6.85 - Rys. 6.92. Badania przeprowadzono zgodnie z metodyką opisaną w rozdziale 6.3.

6.4.2.1. Badanie doświadczalne synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy

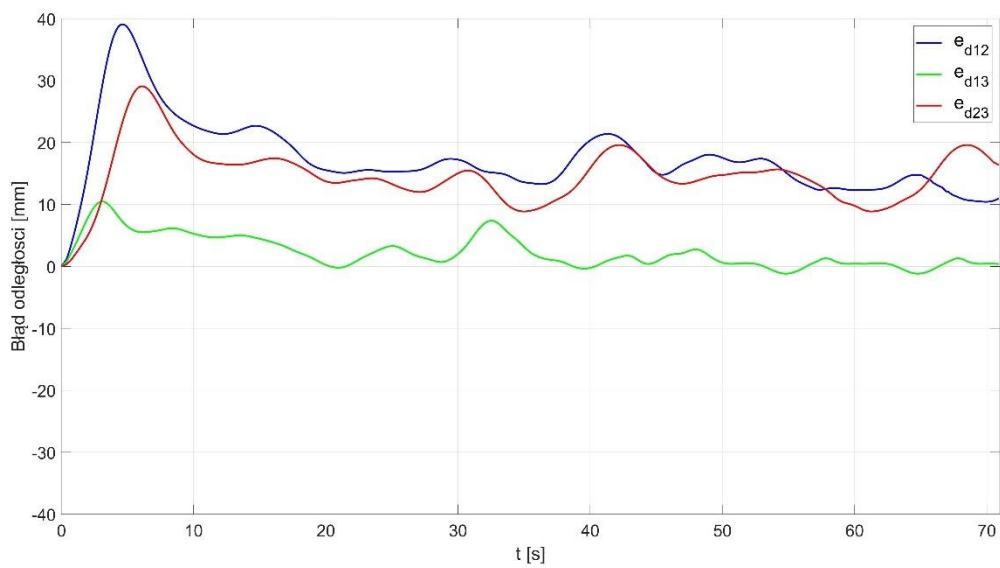
Badanie przeprowadzono dla grupy trzech robotów, której zadaniem był ruch po okręgu o promieniu 2 m i przejazd drogi równej $\frac{3}{4}$ obwodu. W algorytmie sterowania nie uwzględniono kontrolera konfiguracji grupy przedstawionego w rozdziale 4.3. Wyniki badania przedstawiono na Rys. 6.85 - Rys. 6.88.



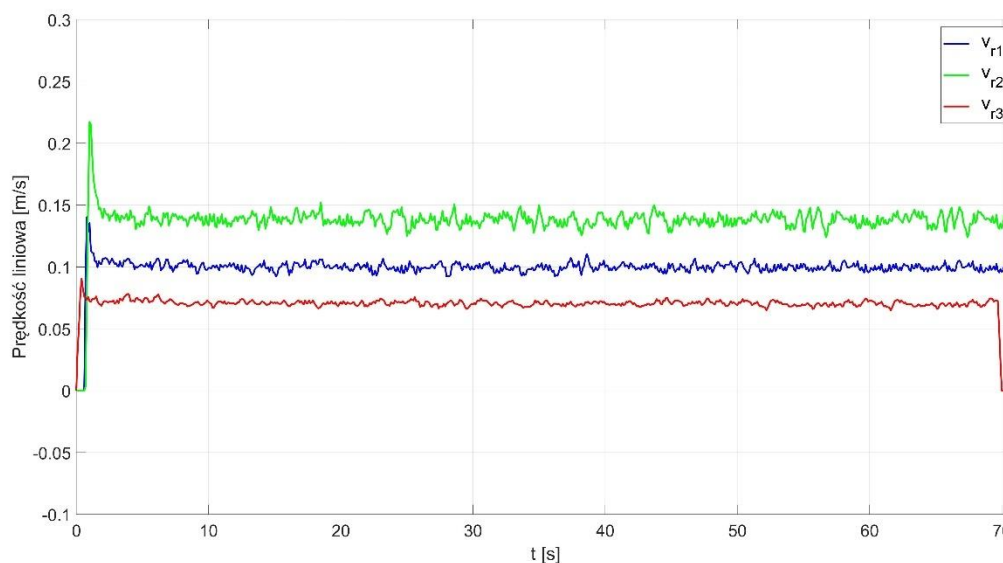
Rys. 6.85. Trajektorie rzeczywiste robotów w badaniu laboratoryjnym synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy



Rys. 6.86. Uchyby śledzenia trajektorii wzorcowych w badaniu laboratoryjnym synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy



Rys. 6.87. Błędy odległości pomiędzy robotami w badaniu laboratoryjnym synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy

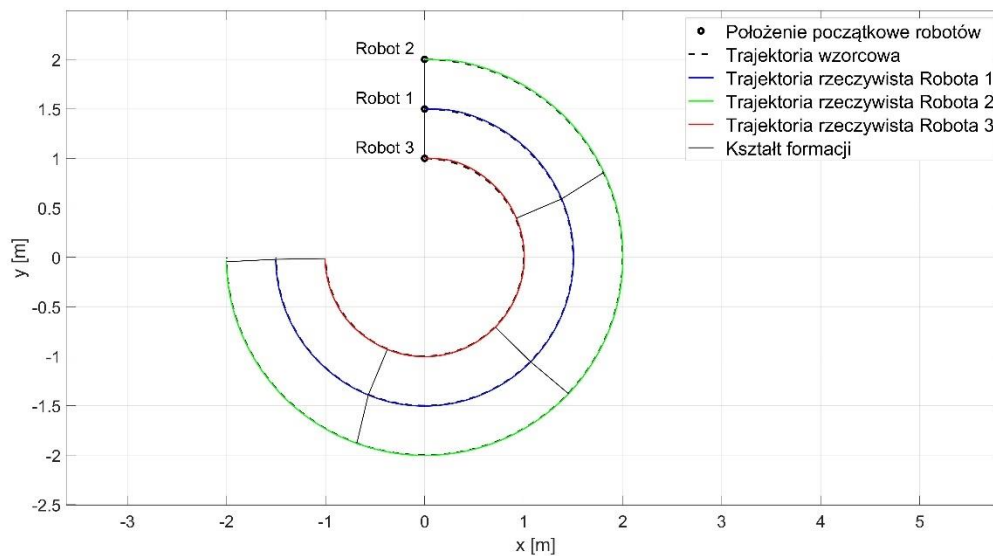


Rys. 6.88. Prędkości liniowe robotów wyznaczone doświadczalnie w badaniu synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia kontrolera konfiguracji grupy

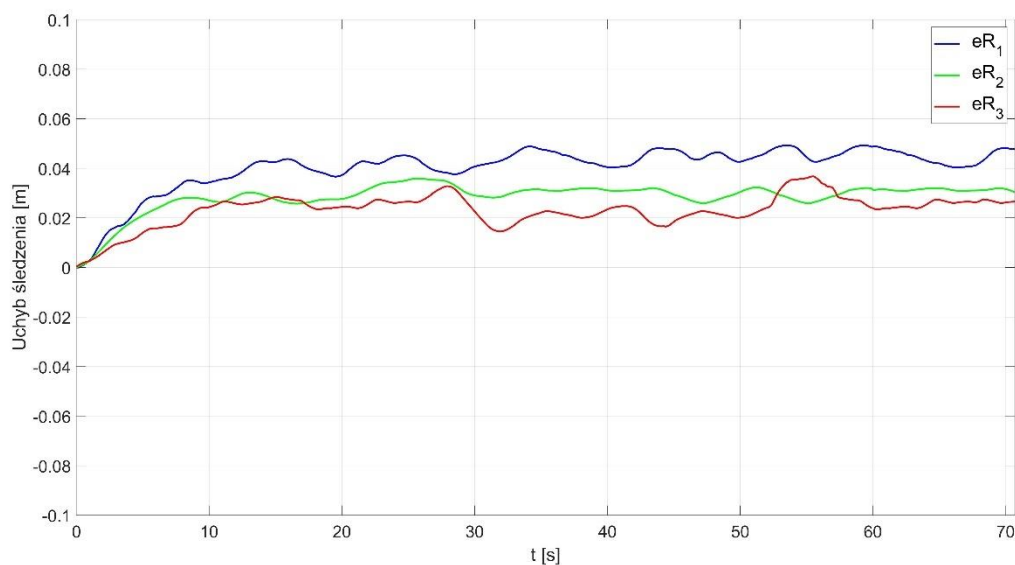
W badaniu doświadczalnym synchronicznego ruchu grupy po zadanej trajektorii bez uwzględnienia warstwy sterowania W1, roboty poruszały się po zadanej trajektorii, jednak w wyniku występowania opóźnień sygnałów sterujących, konfiguracja grupy podczas ruchu nie była utrzymywana co jest widoczne na Rys. 6.85. Analizując wykres prędkości (Rys. 6.88) zauważono, że roboty poruszały się z zadeklarowaną prędkością, jednak start Roboty 1 i Roboty 2 był opóźniony w stosunku do Roboty 3. Analizując wykresy uchybów śledzenia (Rys. 6.86) i uchybów odległości (Rys. 6.87) można zauważyć, że pojawiły się duże uchyby odległości pomiędzy robotami, które osiągnęły maksymalne wartości: $e_{d12} = 39.6 \text{ mm}$, $e_{d13} = 10.9 \text{ mm}$, $e_{d23} = 27.8 \text{ mm}$. Wartości uchybów śledzenia trajektorii zadanych przez poszczególne roboty nie były większe niż: $e_{R1} = 0.05 \text{ m}$, $e_{R2} = 0.06 \text{ m}$, $e_{R3} = 0.04 \text{ m}$ i ostatecznie ustabilizowały się przy wartości około 0.01 m .

6.4.2.2. Badanie doświadczalne synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy

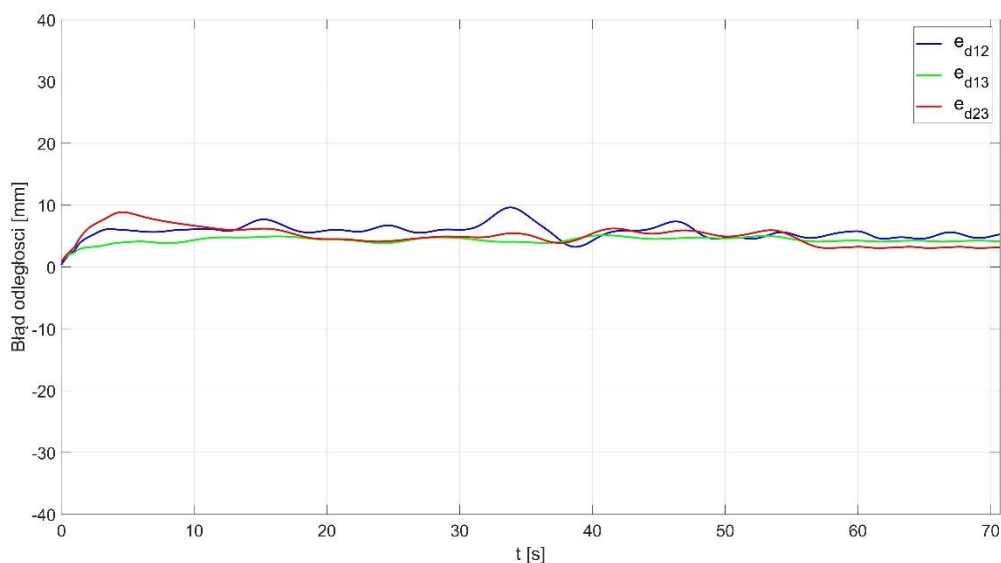
Badanie przeprowadzono dla grupy trzech robotów TURTLEBOT 2. W algorytmie sterowania uwzględniono kontroler konfiguracji grupy przedstawionego w rozdziale 4.3. Wyniki badania przedstawiono na Rys. 6.89 - Rys. 6.92.



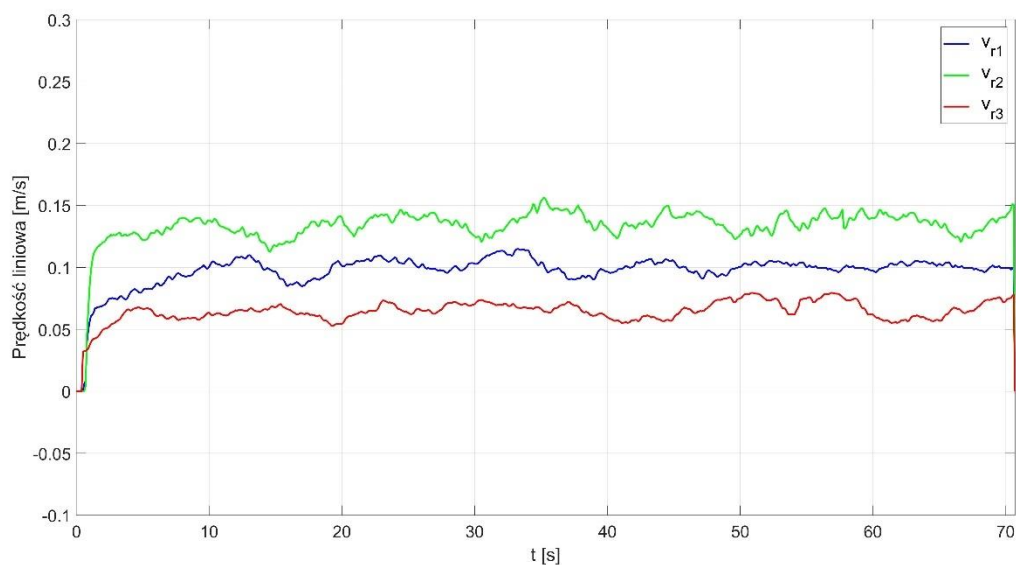
Rys. 6.89. Trajektorie rzeczywiste robotów w badaniu laboratoryjnym synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy



Rys. 6.90. Uchyby śledzenia trajektorii wzorcowych w badaniu laboratoryjnym synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy



Rys. 6.91. Błędy odległości pomiędzy robotami w badaniu laboratoryjnym synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy



Rys. 6.92. Prędkości liniowe robotów wyznaczone doświadczalnie w badaniu synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy

W badaniu doświadczalnym synchronicznego ruchu grupy po zadanej trajektorii z uwzględnieniem kontrolera konfiguracji grupy, roboty poruszały się po zadanej trajektorii z zachowaniem zadeklarowanej konfiguracji grupy co jest widoczne na Rys. 6.89. Analizując wykres prędkości (Rys. 6.92) zauważono, że roboty wystartowały synchronicznie i zakończyły ruch w tym samym czasie. Analizując wykresy uchybów śledzenia (Rys. 6.90) i uchybów odległości (Rys. 6.91) można zauważyć, że w trakcie całego ruchu uchyby odległości pomiędzy

robotami są minimalizowane i utrzymywane, a ich wartości są dopuszczalne i nie większe niż: $e_{d12} = 9.6 \text{ mm}$, $e_{d13} = 5.1 \text{ mm}$, $e_{d23} = 8.7 \text{ mm}$. Uchyby śledzenia trajektorii przez poszczególne roboty w tym badaniu osiągnęły wartości nie większe niż: $eR_1 = 0.049 \text{ m}$, $eR_2 = 0.036 \text{ m}$, $eR_3 = 0.037 \text{ m}$.

6.4.3. Podsumowanie badań doświadczalnych warstwy zewnętrznej W1 algorytmu sterowania

Analizując uchyby położenia robotów i odległości pomiędzy robotami, wyznaczone w badaniach doświadczalnych warstwy wewnętrznej W1 algorytmu sterowania, czyli badaniach synchronizacji rozpoczęcia ruchu i synchronicznego ruchu po zadanej trajektorii można stwierdzić, że zaproponowany autorski algorytm sterowania grupą robotów działa prawidłowo.

Wyznaczone w badaniach doświadczalnych z wykorzystaniem rzeczywistych robotów TURTLEBOT 2 wartości uchybów są akceptowalne z punktu widzenia zastosowania jako system transportowy. Wartości uchybów wyznaczonych w badaniach doświadczalnych zestawiono w tabeli poniżej.

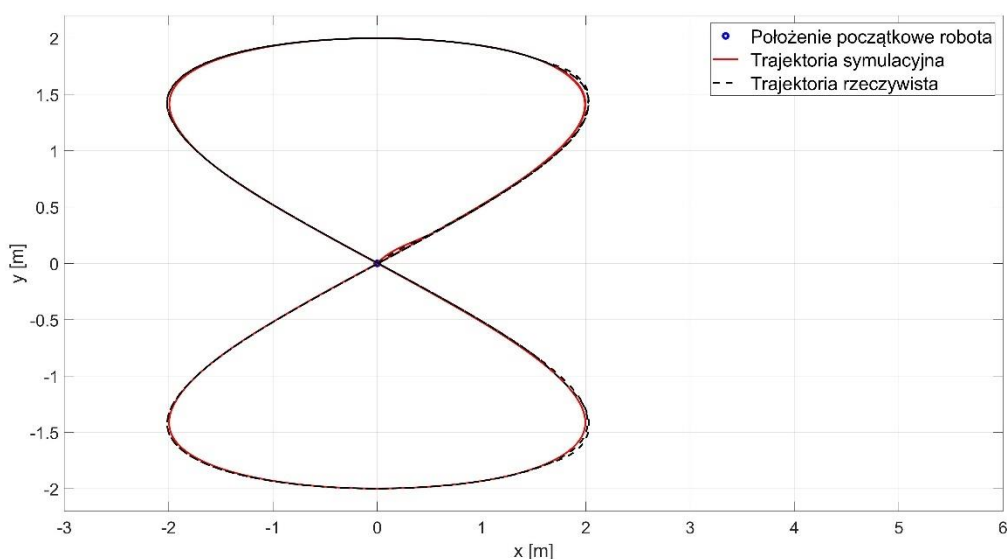
Tabela 19. Wartości uchybów wyznaczonych w badaniach doświadczalnych warstwy zewnętrznej W1 algorytmu sterowania

	Badanie doświadczalne synchronizacji rozpoczęcia ruchu		Badania doświadczalne synchronicznego ruchu grupy po zadanej trajektorii	
	Bez uwzględnienia kontrolera grupy	Z uwzględnieniem kontrolera grupy	Bez uwzględnienia kontrolera grupy	Z uwzględnieniem kontrolera grupy
$eR_1 \text{ [m]}$	0.02	0.18	0.05	0.05
$eR_2 \text{ [m]}$	0.02	0.07	0.06	0.04
$eR_3 \text{ [m]}$	0.02	0.04	0.04	0.04
$e_{d12} \text{ [mm]}$	23	3.1	39.6	9.6
$e_{d13} \text{ [mm]}$	21	4.6	10.9	5.1
$e_{d23} \text{ [mm]}$	26	7.8	27.8	8.7

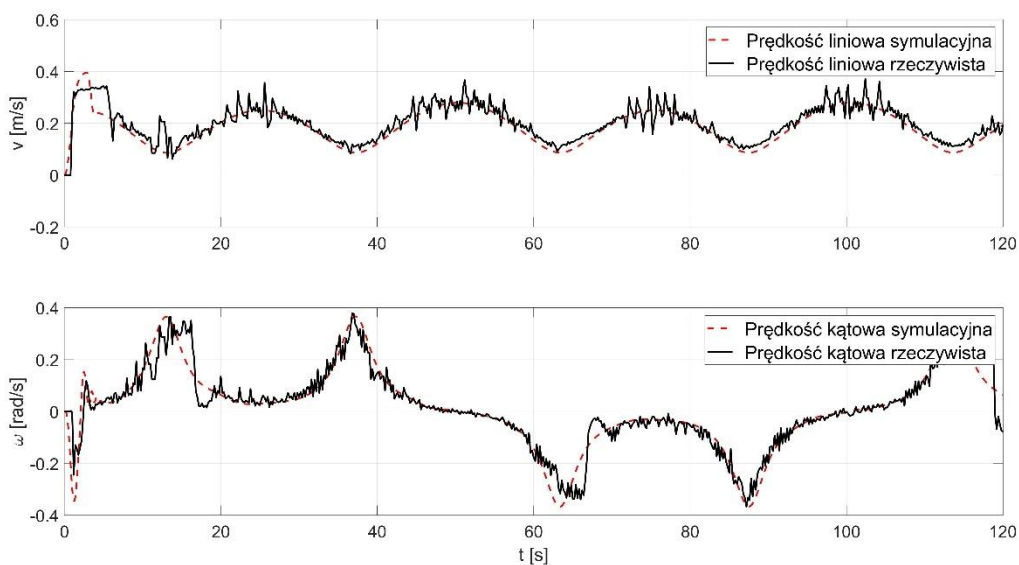
6.5. Podsumowanie badań doświadczalnych

Analizując wyznaczone w badaniach doświadczalnych warstwy wewnętrznej W2 uchyby położenia, orientacji i prędkości, które zestawiono w tabelach w rozdziale 6.4.3 można stwierdzić, że otrzymane wyniki są zgodne z badaniami teoretycznymi.

Potwierdza to porównanie trajektorii uzyskanej w wyniku symulacji i badań rzeczywistych przedstawione na Rys. 6.93 oraz porównanie symulacyjnych i rzeczywistych prędkości liniowej i kątowej przedstawione na Rys. 6.94. Na przedstawionych wykresach można zauważyć dokładne pokrycie się wyników obliczeń co świadczy o prawidłowym zaprojektowaniu kontrolerów i doborze ich parametrów. Uzyskane wyniki pozwalają na zastosowanie zaproponowanej konfiguracji układu sterowania jako warstwa wewnętrzna algorytmu sterowania grupą robotów.



Rys. 6.93. Porównanie trajektorii symulacyjnej i rzeczywistej uzyskanej w badaniach śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją



Rys. 6.94. Porównanie symulacyjnych i rzeczywistych prędkości: liniowej v i kątowej ω , uzyskanych w badaniach śledzenia trajektorii „ósemka” za pomocą kontrolera kinematyczno-dynamicznego z adaptacją

Na podstawie przeprowadzonych badań doświadczalnych warstwy zewnętrznej W1 algorytmu sterowania, mających na celu sprawdzenie jego odporności na opóźnienia czasowe sygnałów sterujących i zakłócenia wynikające z niesprawności robotów wchodzących w skład grupy, można stwierdzić, że zaproponowany autorski algorytm sterowania przy występowaniu opóźnień czasowych sygnałów sterujących realizuje synchroniczny start i ruch grupy po zadanej trajektorii z zachowaniem ustalonego kształtu formacji.

Zastosowanie synchronizacji czasowej sygnałów sterujących pozwoliło na zmniejszenie błędów odległości pomiędzy robotami wywołanych opóźnieniami czasowymi oraz zakłóceniami:

- w badaniu synchronizacji rozpoczęcia ruchu: o 98% dla e_{d12} , o 87% dla e_{d13} , oraz o 0% dla e_{d23} ;
- w badaniu synchronicznego ruchu grupy po zadanej trajektorii o 94% dla e_{d12} , 66% dla e_{d13} , 77% dla e_{d23} .

Badania doświadczalne potwierdziły prawidłowe opracowanie systemu sterowania grupą robotów uwzględniającego synchronizację czasową. Zastosowanie synchronizacji czasowej sygnałów sterujących pozwoliło na znaczne zmniejszenie błędów odległości pomiędzy robotami w czasie startu i ruchu, a w efekcie dokładniejsze zachowanie zadanej formacji grupy.

PODSUMOWANIE I WNIOSKI KOŃCOWE

Zastosowanie robotów mobilnych jako systemów transportowych jest wyraźnym trendem, obserwowanym w obszarze automatyki przemysłowej i robotyzacji procesów produkcyjnych. Roboty mobilne wykorzystywane są na liniach produkcyjnych do realizacji zadań takich jak: dostawa części w procesie montażu, odbiór gotowych produktów, odbiór pełnych palet z zapakowanymi produktami lub dostawa palet pustych, transport towarów w magazynach. Coraz częściej roboty mobilne wyposażane są w manipulatory, co znacznie zwiększa ich funkcjonalność – taki system jest w stanie samodzielnie pobrać ładunek, przewieźć we wskazane miejsce i odłożyć – przekazać do dalszej obróbki. W obszarze zastosowań militarnych, roboty mobilne zwykle rozpatrywane są w aspekcie prowadzenia rozpoznania terenu, dostawy zaopatrzenia, wskazywania celów oraz jako odciążenie żołnierzy w marszu (poprzez transport ich wyposażenia).

Przeprowadzony przegląd literatury pokazuje jednak, że coraz częściej, w celu wykonania skomplikowanego zadania roboty integrowane są w grupy. Ogromny potencjał tego zagadnienia potwierdza bardzo duża ilość badań dotyczących metod sterowania grupami robotów mobilnych. Na ich podstawie, w aspekcie sterowania, można wyróżnić dwie struktury grup robotów – centralną (hierarchiczną) i zdecentralizowaną (rozproszoną), a ich wybór jest ściśle powiązany ze specyfiką realizowanego zadania. Rozpatrując obszar transportu wewnętrznego, który jest obszarem zainteresowań w niniejszej pracy, uwagę należało skupić na strukturze zdecentralizowanej.

Analizując głębiej algorytmy sterowania rozproszonymi grupami robotów mobilnych i zadania realizowane przez grupy zauważono, że w ostateczności każde zadanie sprowadza się do śledzenia przez robota trajektorii wzorcowej (zadanej). W związku z powyższym dużą część przeglądu literatury stanowi przegląd algorytmów śledzenia trajektorii przez jednego robota.

Nieodłączną częścią systemu sterowania pojedynczym robotem czy grupą robotów są błędy wpływające na dokładność śledzenia trajektorii wzorcowej i dokładność pozycjonowania robotów w grupie. W przeprowadzonej analizie stanu zagadnienia zestawiono źródła błędów wpływające na dokładność pozycjonowania robotów w grupie. Ze względu na bezprzewodową komunikację pomiędzy robotami oraz fakt, że jest ona wysoce podatna na zakłócenia, szczególną uwagę należy poświęcić na wyeliminowanie błędów właśnie w tym obszarze. Skupiając się na rozproszonej strukturze grupy robotów stwierdzono, że największym problemem jest dryft czasu, który może doprowadzić nawet do przerwania komunikacji

między robotami. W tym miejscu zauważono możliwość poprawy jakości sterowania grupą rozproszoną wprowadzając w algorytmie synchronizację czasową jednostek w grupie.

Przeprowadzona w rozdziale 1 analiza stanu zagadnienia umożliwiła sformułowanie celu naukowego pracy, którym było opracowanie algorytmu synchronicznego sterowania grupą robotów mobilnych o strukturze rozproszonej. Realizacja tak postawionego celu wymagała zrealizowania kilku zadań cząstkowych związanych z wyborem i analizą obiektu badań, opracowaniem modeli teoretycznych wybranego obiektu badań, opracowaniem algorytmu sterowania grupą rozproszoną wybranych robotów, przeprowadzeniem badań symulacyjnych i doświadczalnych.

W niniejszej pracy wykorzystano założenia sterowania z wykorzystaniem algorytmu Wirtualnej Struktury opisane w pracach [21] i [37] i algorytmu Konsensusu. Zakres badań rozwinęto o wprowadzenie synchronizacji czasowej robotów w grupie, w celu minimalizacji błędów pozycji robotów. Dodatkowo wykluczono błędy śledzenia trajektorii poprzez Wirtualną Strukturę zastępując dynamiczne śledzenie statycznym generatorem trajektorii wzorcowej dla pojedynczych robotów. Minimalizację błędów śledzenia trajektorii wzorcowej każdego robota zrealizowano uwzględniając ich dynamikę i wprowadzając adaptacyjny kontroler dynamiczny, czego nie obejmują prace [21] i [37]. Przedstawione w [41] i [42] rozwiązania były podłożem do sformułowania założeń i opracowania rozwiązań dotyczących dwuwarstwowej architektury systemu sterowania opisanej w niniejszej pracy.

Wybór obiektu badań (rozdział 2) był prosty. Wykorzystano roboty TURTLEBOT 2, które były dostępne w Laboratorium Mechatroniki Instytutu Techniki Rakietowej i Mechatroniki WML WAT. Wybrane roboty dedykowane są do współpracy z systemem operacyjnym Linux i platformą Robot Operating System – w oparciu o którą opracowano przedmiotowy system sterowania. Wstępne badania wybranego robota wykazały, że prowadzenie dokładnych i szczegółowych badań wymaga wprowadzenia modyfikacji w jego konstrukcji polegających na demontażu zbędnego osprzętu, obniżeniu i wysunięciu w kierunku czoła robota środka ciężkości. Wprowadzone modyfikacje pozwoliły zniwelować drgania i wahania robota. Po zapoznaniu się z budową robota TURTLEBOT 2 opracowano jego model matematyczny złożony z modelu kinematycznego (sprowadzonego do postaci, gdzie sygnałami sterującymi są prędkości liniowa i kątowa, a sygnałami wyjściowymi - pozycja i orientacja robota) oraz modelu dynamicznego (w postaci, gdzie sygnałami sterującymi i wyjściowymi są prędkości liniowa i kątowa). Model dynamiczny robota zawierał cztery nieznane parametry, których wartości uzyskano w drodze identyfikacji offline i online. W tym zagadnieniu zastosowano autorskie podejście, pokazujące połączenie tych dwóch metod

w celu minimalizacji błędu identyfikacji. Następnie przeprowadzono analizę wrażliwości, która wykazała wpływ poszczególnych parametrów modelu dynamicznego na uchyby śledzenia trajektorii i prędkości wzorcowej. W kroku ostatnim przeprowadzono walidację modelu matematycznego polegającą na porównaniu trajektorii otrzymanej w wyniku symulacji i zrealizowanej przez rzeczywistego robota w odpowiedzi na zadane wymuszenie. Walidacja dała pozytywny wynik, pokazujący bardzo dobre odzwierciedlenie rzeczywistego robota za pomocą opracowanego modelu teoretycznego. Pozwoliło to na opracowanie modelu teoretycznego dla rozproszonej grupy robotów złożonej z robotów TURTLEBOT 2.

Po dokładnym przebadaniu wybranego robota przeprowadzono badania rozpoznawcze dokładności pozycjonowania robotów w grupie o strukturze rozproszonej, które przedstawiono w rozdziale 3. Miały one na celu rozpoznanie charakteru i wielkości błędów wywołanych opóźnieniami transmisji danych sterujących. W tym przypadku przeprowadzono dwa badania symulacyjne w środowisku MATLAB/Simulink i MSC Adams, oraz dwa badania laboratoryjne z wykorzystaniem rzeczywistych robotów. Podczas badań zauważono, że nawet niewielkie opóźnienia czasowe (rzędu 300 -500 ms) w transmisji sygnałów sterujących mają znaczący wpływ na osiągnięte przez roboty położenie (błąd rzędu 0.12 m). Podczas analizy wykazano, że w przypadku ruchu grupy robotów opóźnienia sygnałów sterujących powodowały deformację szyku, co może działać niszcząco na układy jezdne robotów (w aspekcie zastosowania jako systemy transportowe), a ostatecznie mogło skutkować brakiem możliwości wykonania zadania. Przeprowadzone badania laboratoryjne wykazały wpływ ilości robotów w grupie i innych urządzeń podłączonych do tej samej sieci na wartość opóźnień czasowych komunikacji pomiędzy robotami. Wykazano również, że pojawienie się przeszkód środowiskowych na drodze transmisji pomiędzy robotami a punktem dostępowym nie jest czynnikiem opóźniającym transmisję.

Po uzyskaniu wiedzy na temat źródeł pochodzenia, charakteru i wartości opóźnień czasowych sygnałów sterujących przystąpiono do opracowania autorskiego algorytmu synchronicznego sterowania grupą robotów o strukturze rozproszonej (rozdział 4). W opracowaniu przyjęto dwuwarstwową ideę sterowania. Warstwa zewnętrzna realizowała ruch grupy robotów i utrzymanie zadeklarowanej formacji. W tym celu zastosowano założenia algorytmu Wirtualnej Struktury i Consensus Tracking, który rozbudowano o synchronizację czasową sygnałów. Warstwa wewnętrzna algorytmu realizowała ruch nadążny pojedynczego robota polegający na śledzeniu trajektorii wzorcowej. Do tego zadania wykorzystano kontroler kinematyczno-dynamiczny z adaptacją parametrów modelu.

W celu potwierdzenia poprawności opracowania autorskiego algorytmu sterowania przeprowadzono badania symulacyjne (rozdział 5) w środowisku MATLAB/Simulink. Badania zrealizowano w dwóch etapach. W pierwszej kolejności przeprowadzono symulacje śledzenia trajektorii wzorcowych przez jednego robota co realizowane było przez warstwę wewnętrzną algorytmu sterowania. Następnie przeprowadzono badania symulacyjne warstwy zewnętrznej, która odpowiadała za synchronizację i utrzymanie zadanej formacji grupy robotów.

Badania symulacyjne warstwy wewnętrznej W1 przeprowadzono dla trzech rodzajów trajektorii testowych: kołowej, sinusoidalnej i w kształcie cyfry osiem. Badania przeprowadzono według dwóch postawionych kryteriów: ruchu przez określony czas i ruchu po skończonej drodze, które wykazały stabilność i dokładność systemu sterowania. Badania warstwy wewnętrznej zrealizowano dla trzech konfiguracji systemu sterowania: śledzenie trajektorii za pomocą kontrolera kinematycznego, kinematyczno-dynamicznego oraz kinematyczno-dynamicznego z adaptacją. Zastosowanie trzeciej konfiguracji systemu sterowania pozwoliło na zredukowanie błędów śledzenia trajektorii i prędkości wzorcowej w stosunku do konfiguracji pierwszej o ok. 50% w przypadku trajektorii „koło” i ok. 75% w przypadku trajektorii sinus i ósemka oraz skrócenie czasu regulacji o ok. 50% dla wszystkich przypadków. Otrzymane wyniki potwierdziły fakt, że aby uzyskać dużą dokładność śledzenia w przypadku bardziej skomplikowanych trajektorii konieczne jest uwzględnienie kontrolera kinematyczno-dynamicznego.

Badania symulacyjne warstwy zewnętrznej systemu sterowania przeprowadzono uwzględniając wyłącznie konfigurację systemu sterowania pojedynczego robota obejmującą kontroler kinematyczny i dynamiczny z adaptacją. Badania te przeprowadzono w celu sprawdzenia odporności algorytmu sterowania na opóźnienia czasowe sygnałów sterujących i zakłócenia wynikające z niesprawności robotów wchodzących w skład grupy.

W kolejnym etapie (rozdział 6) algorytm sterowania zaimplementowano na rzeczywistych robotach TURTLEBOT 2 i przebadano w warunkach laboratoryjnych. System sterowania opracowano z wykorzystaniem platformy Robot Operating System (ROS), z którą współpracują roboty TURTLEBOT 2. W celu potwierdzenia wyników uzyskanych w symulacjach, zakres, metodyki, nastawy kontrolerów i warunki początkowe w badaniach doświadczalnych przyjęto takie same jak w symulacjach. Analizując wyznaczone w badaniach doświadczalnych uchyby położenia, orientacji, prędkości i odległości pomiędzy robotami, stwierdzono zgodność z wynikami uzyskanymi drogą symulacji i prawidłowe opracowanie autorskiego algorytmu sterowania.

Na podstawie przeprowadzonych badań stwierdzono, że zaproponowany autorski algorytm sterowania z wykorzystaniem zmodyfikowanej metody Consensus Tracking realizuje synchroniczny ruch grupy z zachowaniem zadanego kształtu formacji nawet w przypadku występowania błędów w komunikacji pomiędzy robotami. Wyznaczone w badaniach symulacyjnych wartości uchybów nie są większe niż 1.3 mm w przypadku ruchu po linii prostej i 5.5 mm w przypadku ruchu po okręgu (w stosunku do 14.2 mm dla ruchu po linii prostej i 55.8 mm dla ruchu po okręgu bez uwzględnienia w algorytmie sterowania kontrolera grupy). Wyznaczone w badaniach doświadczalnych wartości uchybów (w przypadku zastosowania autorskiego algorytmu sterowania) są dopuszczalne z punktu widzenia zastosowania grupy robotów do zadań transportowych.

Przeprowadzone w ramach pracy badania symulacyjne (z wykorzystaniem modeli matematycznych i środowiska MATLAB/Simulink) i doświadczalne (z wykorzystaniem rzeczywistych robotów) potwierdziły prawidłowe opracowanie algorytmu synchronicznego sterowania grupą robotów o strukturze rozproszonej co było celem naukowym dysertacji. Opracowana metoda sterowania pozwala na łatwą adaptację do dostępnych na rynku konfiguracji robotów AGV przedstawionych na Rys. 1.1.

Przeprowadzona analiza zagadnienia oraz wyniki i wnioski z przeprowadzonych badań symulacyjnych i doświadczalnych pozwoliły określić kierunki dalszego rozwoju, które dotyczą:

- opracowania globalnego systemu nawigacji grupy robotów w obrębie hali produkcyjnej, magazynu lub przedsiębiorstwa,
- integracji z robotami przemysłowymi,
- poruszania się w konwoju pojazdów z uzbrojeniem (rozpatrując zastosowania militarne).

Załącznik A

W przedstawionym na Rys. 2.6 przypadku, punkt obrotu robota G położony jest na osi łączącej koła napędowe, zatem siły działające w kierunku osi y nie powodują żadnego momentu obrotowego. W związku z tym układ równań (3.11) można przekształcić do postaci:

$$\begin{aligned}\dot{v}_x &= \frac{F_{rlx} + F_{rrx}}{m} + v_y \omega \\ \dot{\omega} &= \frac{d(F_{rrx} - F_{rlx})}{2I_z}\end{aligned}\quad \text{A.1}$$

Prędkość wzdluzna v_x i poprzeczna v_y oraz katowa ω bez uwzględnienia poślizgu są równe:

$$\begin{aligned}v_x &= \frac{1}{2}r(\omega_p + \omega_l) \\ v_x &= 0 \\ \omega &= \frac{1}{2d}r(\omega_p - \omega_l)\end{aligned}\quad \text{A.2}$$

gdzie r - promień kół napędowych, ω_p, ω_l - prędkość katowa kół (prawego i lewego).

Prędkość katową kół można wyznaczyć przekształcając wyrażenia na momenty napędowe silnika prawego τ_p i lewego τ_l [24]:

$$\begin{aligned}\tau_p &= \frac{k_a(V_p - k_b\omega_p)}{R_a} \\ \tau_l &= \frac{k_a(V_l - k_b\omega_l)}{R_a}\end{aligned}\quad \text{A.3}$$

gdzie V_p, V_l - napięcia zasilania silników (prawego i lewego), k_b - stała elektryczna silników, k_a - stała mechaniczna silników, R_a - rezystancja uzwojeń silników.

Równania dynamiki kół napędowych [24] przyjmują postać:

$$\begin{aligned}I_e\dot{\omega}_r + B_e\omega_r &= \tau_p - F_{rrx}r \\ I_e\dot{\omega}_l + B_e\omega_l &= \tau_l - F_{rlx}r\end{aligned}\quad \text{A.4}$$

gdzie I_e - moment bezwładności koła względem osi obrotu, B_e - współczynnik tarcia lepkiego zredukowany do wału silnika.

Równania kontrolera PD zaimplementowanego do regulacji napięcia zasilania silników [24] przyjmują postać:

$$\begin{aligned}u_v &= k_{PT}(v_d - v_x) - k_{DT}\dot{v}_x \\ u_\omega &= k_{PR}(\omega_d - \omega) - k_{DR}\dot{\omega}\end{aligned}\quad \text{A.5}$$

gdzie $u_v = \frac{V_p + V_l}{2}$, $u_\omega = \frac{V_p - V_l}{2}$, $k_{PT}, k_{DT}, k_{PR}, k_{DR}$ - wzmacnienia kontrolera PD, v_d - referencyjna prędkość liniowa, ω_d - referencyjna prędkość katowa.

Podstawiając równanie (A.3) do równania (A.4) i przekształcając do postaci na wyznaczenie sił napędowych wzdłużnych otrzymano układ równań:

$$\begin{aligned} F_{rrx} &= \frac{1}{r} \left(\frac{k_a}{R_a} (V_p - k_b \omega_p) - I_e \dot{\omega}_p + B_e \omega_p \right) \\ F_{rlx} &= \frac{1}{r} \left(\frac{k_a}{R_a} (V_l - k_b \omega_l) - I_e \dot{\omega}_l + B_e \omega_l \right) \end{aligned} \quad A.6$$

Następnie podstawiając równania (A.6) do układu równań (A.1) otrzymano:

$$\begin{aligned} \dot{v}_x &= \frac{1}{rm} \left(\frac{k_a}{R_a} (V_l + V_p) + (\omega_l + \omega_p) \left(-\frac{k_a}{R_a} + B_e \right) - I_e (\omega_l + \omega_p) \right) + v_y \omega \\ \dot{\omega} &= \frac{d}{I_z^2 r} \left(\frac{k_a}{R_a} (V_p - V_l) + (\omega_p + \omega_l) \left(-\frac{k_a}{R_a} k_b + B_e \right) - I_e (\omega_p + \omega_l) \right) \end{aligned} \quad A.7$$

Łącząc powyższe równania z równaniami (A.2) i (A.3) oraz dokonując przekształceń algebraicznych otrzymano wyrażenia opisujące model dynamiczny robota TURTLEBOT 2:

$$\begin{aligned} \dot{v}_x \left(\frac{1}{k_{PT}} \left(\frac{R_a}{k_a} \left(\frac{rm}{2} + \frac{I_e}{r} \right) + k_{DT} \right) \right) &= -v_x \left(1 - \frac{1}{rk_{PT}} \left(1 + B_e \frac{R_a}{k_a} \right) \right) + v_d \\ \dot{\omega}_x \left(\frac{1}{k_{PR}} \left(\frac{R_a}{2k_a} \left(\frac{I_z^2 r}{d} + \frac{dI_e}{r} \right) + k_{DR} \right) \right) &= -\omega \left(\frac{d}{rk_{PR}} \left(1 - \frac{R_a}{k_a} \right) + 1 \right) + \omega_d \end{aligned} \quad A.8$$

lub w skróconej formie:

$$\begin{aligned} \dot{v}_x \sigma_1 &= -v_x \sigma_3 + v_d \\ \dot{\omega}_x \sigma_2 &= -\omega \sigma_4 + \omega_d \end{aligned}$$

$$\begin{aligned} \sigma_1 &= \frac{1}{k_{PT}} \left(\frac{R_a}{k_a} \left(\frac{rm}{2} + \frac{I_e}{r} \right) + k_{DT} \right) \\ \sigma_2 &= \frac{1}{k_{PR}} \left(\frac{R_a}{2k_a} \left(\frac{I_z^2 r}{d} + \frac{dI_e}{r} \right) + k_{DR} \right) \\ \sigma_3 &= 1 - \frac{1}{rk_{PT}} \left(1 + B_e \frac{R_a}{k_a} \right) \\ \sigma_4 &= \frac{d}{rk_{PR}} \left(1 - \frac{R_a}{k_a} \right) + 1 \end{aligned} \quad A.9$$

Załącznik B

Wymnażając równanie (4.17) obustronnie przez $\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$ otrzymano postać:

$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\sigma_3 v_x \\ -\sigma_4 \omega \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_d \\ \omega_d \end{bmatrix} \quad \text{B.1}$$

Dokonując przekształceń, powyższe wyrażenie można zapisać w postaci:

$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \sigma_3 & 0 \\ \sigma_4 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_d \\ \omega_d \end{bmatrix} \quad \text{B.2}$$

a następnie doprowadzić do postaci parametrycznej:

$$\begin{bmatrix} v_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0 & 0 & v_x & 0 \\ 0 & 0 & 0 & \omega \end{bmatrix} [\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4]^T \quad \text{B.3}$$

lub w skróconej formie:

$$\begin{bmatrix} v_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} \dot{v}_x & 0 & v_x & 0 \\ 0 & 0 & \dot{v}_x & \omega \end{bmatrix} [\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4]^T \quad \text{B.4}$$

$$\begin{bmatrix} v_d \\ \omega_d \end{bmatrix} = A\sigma \quad \text{B.5}$$

Bazując na zadaniu odwrotnym dynamiki [24], [76] wyrażenie (B.3) można sprowadzić do postaci kontrolera dynamicznego (4.28).

LITERATURA

- [1] Ailon A., Arogeti S.: *Trajectory tracking control for a kinematic bicycle model*, 28th Mediterranean Conference on Control and Automation, MED, 2020, nr.art. 9183161, s. 526-531.
- [2] Alismail H., Baker L.D., Browning B.: *Continuous Trajectory estimation for 3D SLAM from actuated lidar*, IEEE International Conference on Robotics and Automation (materiały konferencyjne), 2014, nr art. 6907757, s. 6096-6101, doi: 10.1109/ICRA.2014.6907757
- [3] Arkin R. C., *Behavior-Based Robotics*, MIT, Cambridge, 1998.
- [4] Arnold R., Carey K., Abruzzo B., Korpela Ch.: *What is A Robot Swarm: A Definition for Swarming Robotics*, Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, s. 74-81
- [5] Autefade V., Chaumette S., Magoni D.: *Comparison of time synchronization techniques in a distributed collaborative swarm system*, IEEE 24th European Conference on Networks and Communications, Materiały konferencyjne, 2015, 460-464.
- [6] Autefade V., Chaumette S., Magoni D.: *Comparison of time synchronization techniques in a distributed collaborative swarm system*, IEEE 24th European Conference on Networks and Communications (Materiały konferencyjne), 2015, s. 460-464.
- [7] Bai J., Sun Z., Chen Y.: *Trajectory tracking control for wheeled mobile robots with input saturation*, 2020 7th International Conference on Information, Cybernetics, and Computational Social Systems, ICCSS, 2020, nr. art. 9336810, s. 537-540.
- [8] Balch T., Arkin R. C.: *Behavior-based formation control for multirobot teams*, IEEE Trans. on Robotics and Automation, 1998, nr 14, s.926–939.
- [9] Baranowski L., Siwek M.: *Use of 3D Simulation to Design Theoretical and Real Pipe Inspection Mobile Robot Model*, Acta Mechanica et Automatica, 2018, nr 12, s. 232-236, doi: 10.2478/ama-2018-0036
- [10] Besseghieur K. L., Trębiński R., Kaczmarek, W., Panasiuk, J.: *From Trajectory Tracking Control to Leader-Follower Formation Control*, Cybernetics and Systems, 2020, 51(4), s. 339-356
- [11] Besseghieur K. L., Trębiński R., Kaczmarek, W., Panasiuk, J.: *Leader-follower formation control for a group of ROS-enabled mobile robots*, International Conference on Control, Decision and Information Technologies (materiały konferencyjne), 2019, s. 1556–1561, 8820460
- [12] Besseghieur K. L.: *Leader-Follower Formation Control with Dynamic Leader Changing of Nonholonomic Mobile Robots Under Robot Operating System*, praca doktorska, WAT, Warszawa 2019
- [13] Borgonovo E.: *Sensitivity Analysis*, Springer, 2017.
- [14] Boyden F. D., Velinsky S. A.: *Dynamic Modeling of Wheeled Mobile Robots for High Load Applications*, IEEE International Conference on Robotics and Automation, 4, 1994, s. 3071–3078.
- [15] Burghardt A., Hendzel Z., Giergiel J., Nawrocki M.: *Sterowanie formacją robotów metodą śledzenia lidera*, Modelowanie Inżynierskie, 2012, nr 45, t. 14, s.220-225.
- [16] Cai N., Diao Ch., Khan M. J.: *A novel clustering method based on quasi-consensus motions of dynamical multiagent systems*, Complexity, 2017, s. 1-8.
- [17] Cai N., Diao Ch., Khan M. J.: *A novel clustering method based on quasi-consensus motions of dynamical multiagent systems*, Complexity, 2017, s. 1-8.

- [18] Chaimowicz L., Kumar V., Campos M. F. M.: *A Paradigm for Dynamic Coordination of Multiple Robots*, Autonomous Robots, 2004, nr. 17, s. 7-21
- [19] Chen J., Sun D., Yang J., Chen H.: *A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme*, Int. J. Robot. Res., 2010, vol. 29, nr 6, s. 727–747.
- [20] Chen S., Liu J., Liang X., Zhang S., Hyypä J., Chen R.: *A Novel Calibration Method between a Camera and a 3D LiDAR with Infrared Images*, 2020, IEEE International Conference on Robotics and Automation (materiały konferencyjne), nr art. 9196512, s. 4963-4969, doi: 10.1109/ICRA40945.2020.9196512
- [21] Cheng C., Yu X., Ou L., Guo Y.: *Research on multi-robot collaborative transportation control system*, Proceedings of Control and Decision Conference (CCDC), 2016, s. 4886-4891.
- [22] Das K.: *A vision-based formation control framework*, IEEE Trans. On Robotics and Automation, 2002, nr 18, s.813–825.
- [23] Das T., Kar I. N.: *Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots*, IEEE Transaction on Control Systems Technology, 2006, nr 14(3), 501-510
- [24] De La Cruz C., Carelli R.: *Dynamic model based formation control and obstacle avoidance of multi-robot systems*, Robotica, 2008, vol 26, 345-356.
- [25] Domański Cz.: *Metody statystyczne teoria i zadania*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź, 2001
- [26] Dong W., Farrel J. A.: *Cooperative Control of Multiple Nonholonomic Mobile Agents*, IEEE Transaction on Automatic Control, 2008, nr 53(6), s. 414-424.
- [27] Dudka P.: *Czynniki wpływające na dokładność i powtarzalność pozycjonowania robota przemysłowego*, Pomiar Automatyka Robotyka, 2016, nr. 4, s. 59-66.
- [28] Fateh M.M., Khorashadizadeh S.: *Robust control of electrically driven robots by adaptive fuzzy estimation of uncertainty*, Nonlinear Dynamics, vol. 69, nr. 3, s. 1465-1477.
- [29] Fateh M.M.: *Robust control of flexible-joint robots using voltage control strategy*, Nonlinear Dynamics, 2012, vol. 67 nr. 2, s. 1525-1537.
- [30] Fax J. A., Murray R. M.: *Information Flow and Cooperative Control of Vehicle Formations*, IEEE Transaction on Automatic Control, 2004, vol. 49, nr 9, s. 1465-1476.
- [31] Fiengo G., Lui D. G., Petrillo A., i inni: *Distributed robust output consensus for linear multi-agent systems with input time-varying delays and parameter uncertainties*, IET Control Theory and Applications, 2018, nr 13, s. 203-212
- [32] Fukao T., Nakagawa H., Adachi N.: *Adaptive tracking control of a nonholonomic mobile robot*, IEEE Transaction on Robotics and Automation, nr 16(5), 609-615.
- [33] Giergiel M., Hendzel Z., Żylski W.: *Modelowanie i sterowanie mobilnych robotów kołowych*, Warszawa, 2002, PWN.
- [34] Giergiel M., Małka P.: *Modelowanie kinematyki i dynamiki mobilnego minirobota*, Modelowanie Inżynierskie, 2006, nr 32, s. 157-162.
- [35] Gratton S., Lawless A.S., Nichols N.K.: *Approximate Gauss-Newton methods for nonlinear least squares problems*, SIAM journal on optimization, 2007, vol.18(1), 106-132.
- [36] Guanghua W., Deyi L., Wenyan G., Peng J.: *Study on formation control of multi-robot systems*, Intelligent System Design and Engineering Applications (ISDEA), 2013, s. 1335-1339.
- [37] Hassan B., Ayoob A.: *On the New Quasi-Newton Equation for Unconstrained Optimization*, International Engineering Conference: Towards Engineering Innovations

- and Sustainability (materiały konferencyjne), 2022, s. 168-172, doi: 10.1109/IEC54822.2022.9807584
- [38] Hassan B., Moghrabi I.: *A modified secant equation quasi-Newton method for unconstrained optimization*, Journal of Applied Mathematics and Computing, 2023, vol. 69, s. 451-464, doi: 10.1007/s12190-022-01750-x
- [39] Holland J.: *Design Autonomous Mobile Robots*, Elsevier Inc., Oxford, 2004
- [40] http://sun-light.com.sg/index.php?route=product/product&product_id=879 (dostęp 12.12.2022)
- [41] <https://robosklep.com/pl/roboty-mobilne/142-robotnik-turtlebot-2.html> (dostęp 12.12.2022)
- [42] Hu Y., Wu X., Zhai J., Lou P., Qian X., Xiao H.: *Hybrid Task Allocation of an AGV System for Task Groups of an Assembly Line*, Applied Sciences (Switzerland), 2022, nr 12(21), nr art. 10956, doi: 10.3390/app122110956
- [43] Hua C., Sun X., You X., Guan X.: *Finite-time consensus control for second-order multi-agent systems without velocity measurements*, International Journal of Systems Science, 2016, 48 (2), s. 1-10
- [44] Huang J.: *Localization and follow-the-leader control of a heterogeneous group of mobile robots*, IEEE/ASME Trans. on Mechatronics, 2006, nr 11, t.2, s.205-215.
- [45] ISO 3691-4:2020 *Industrial Trucks - Safety Requirements And Verification - Part 4: Driverless Industrial Trucks And Their Systems*
- [46] Jiang Z.-P., Nijmeijer H.: *Tracking control of mobile robots: a case study in backstepping* Automatica, 1997, nr. 33(7), s. 1393–1399.
- [47] Kaczmarek W., Besseghieur K., Panasiuk J., Prusaczyk P., Siwek M.: *Industrial and ROS – enabled mobile robots cooperation*, 2018, Materiały konferencyjne ENGINEERING MECHANICS 2018, vol. 24, s. 349-352.
- [48] Kaczmarek W., Panasiuk J., Borys S. Dyczkowski R., Siwek M.: *Robotyzacja i Automatyzacja. Przemysł 4.0.*, PWN, Warszawa, 2022
- [49] Kanjanawanishkul K., Hofmeister M., Zell A.: *Path Following with an Optimal Forward Velocity for a Mobile Robot*, IFAC (materiały konferencyjne), 2010, nr 43(16), 19-24.
- [50] Kar-Han T., Lewis M. A.: *Virtual structures for high-precision cooperative mobile robotic control*, Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996, nr. 1, s. 132-139.
- [51] Khatib O.: *Real-time obstacle avoidance for manipulators and mobile robots. Autonomous robot vehicles*, Springer, 1986, s. 396-404
- [52] Khooban M., *Design an intelligent proportional-derivate (PD) feedback linearization control for nonholonomic-wheeled mobile robot*, Journal of Intelligent and Fuzzy Systems, 2014, nr 26(4), 1833-1843.
- [53] Klancar G., Matko D., Blazic S.: *Mobile Robot Control on a Reference Path*, IEEE International Symposium on Control and Automation Intelligent Control, 2005, 1343-1348.
- [54] Komareji M., Shang Y., Boiffanais R.: *Consensus in topologically interacting swarms inder communication constraints and time-delays*, Nonlinear Dynamics, 2018, nr 93, s. 1287-1300.
- [55] Komareji M., Shang Y., Boiffanais R.: *Consensus in topologically interacting swarms inder communication constraints and time-delays*, Nonlinear Dynamics, 2018, nr 93, s. 1287-1300.
- [56] Krzysztofik I., Koruba Z.: *Study on the sensitivity of a gyroscope system homing a quadcopter onto a moving ground targer under the action of external disturbance*, Energies, 2021, nr 14(6), nr art. 1696, doi: 10.3390/en14061696

- [57] Kube C. R., Bonabeau E.: *Cooperative transport by ants and robots*, Robotics and Autonomous Systems, 2000, nr 30, s. 85-100.
- [58] KUKA – Robots & Automation, <https://prom.ua/p558776844-robot-gajkovert-kuka.html?&primelead=MS4zNQ> dostęp 08.12.2022
- [59] Kula R., Pielot J.: Zastosowanie analizy wrażliwości do optymalnego sterowania procesów przeróbki węgla, Zeszyty naukowe Politechniki Śląskiej, 2000, z. 246/III, s. 303-314
- [60] Ładyżyńska-Kozdraś E.: *Robotnic Underwater Vehicle Steered by a Gyroscope – Model of Navigation and Dynamics*, Mechatronics 2013, 2014, s. 627-632
- [61] Leonard N. E., Fiorelli E.: *Virtual leaders, artificial potentials and coordinated control of groups*, Proc. of the 40th IEEE Conf. on Decision and Control, 2001, s. 2968–2973.
- [62] Lewis F., Zhang H., Hengster-Movric K., Das A.: *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches*, Springer, 2014
- [63] Lewis M. A., Kar-Han T.: *High Precision Formation Control of Mobile Robots Using Virtual Structures*, Autonomous Robots, 1997, nr. 4, s. 387-403
- [64] Liu L., Yu J., Ji J., Miao Z., Zhou J.: *Cooperative adaptive consensus tracking for multiple nonholonomic mobile robots*, International Journal of Systems Science, 2019, nr 8, s. 1556-1567
- [65] Liu S., Sun D., Zhu C., Shang W.: *A decentralized local constraint path planner for multiple mobile robots*, IEEE International Conference on Information and Automation, 2009, 1222-1227
- [66] Liu S., Sun D., Zhu Ch., Shang W.: *A dynamic priority strategy in decentralized motion planning for formation forming of multiple mobile robots*, The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, 3774-3779
- [67] Liu S., Sun D., Zhu Ch.: *Motion Planning of Multirobot Formation*, The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, 3848-3853
- [68] Liu S., Zhang H., Yang S. X., Yu J.: *Dynamic control of a mobile robot using an adaptive neurodynamics and sliding mode strategy*, 5th World Congress on Intelligent Control and automation (materiały konferencyjne), 2004, 5007-5011.
- [69] Logistyka i przemysł 4.0 w praktyce. Roboty transportowe skracają czas produkcji, <https://trans.info/pl/logistyka-4-0-roboty-transportowe-skracaja-czas-produkcji-85274>, dostęp 5.01.2023 r.
- [70] López J., Zalama E., Gómez-García-Bermejo J.: *A simulation and control framework for AGV based transport systems*, Simulation Modelling Practice and Theory, 2022, nr 116, nr art. 102430, doi: 10.1016/j.simpat.2021.102430
- [71] Ma H., Wang Z., Wang D., i inni: *Neural-network-Based distributed adaptive robust control for a class of nonlinear multiagent systems with time delays and external noise*, IEEE Transaction on Systems, Man and Cybernetics, 2016, nr 46, s. 750-758.
- [72] Małka P.: *Pozycjonowanie i nadążanie minirobota kołowego*, praca doktorska, Akademia Górniczo - Hutnicza, Kraków, 2007.
- [73] Mariottini G. L., Morbidi F., Prattichizzo D., Vander Valk N., Michael N., Pappas G., Daniilidis K.: *Vision-based localization for leader–follower formation control*, IEEE Transactions on Robotics, 2009, nr 25(6), s.1431–1438.
- [74] Martins F. N., Celeste W. C., Carelli R., Sarcinelli-Filho M., Bastos T.: *Kinematic and adaptive dynamic trajectory tracking controller for mobile robots*, International Conference on Advances in Vehicle Control and Safety AVCS'07 (materiały konferencyjne), 2007, s. 29-34
- [75] Martins F. N., Celeste W. Carelli C., R., Sarcinelli-Filho M., Bastos-Filho T. F.: *An adaptive dynamic controller for autonomous mobile robot trajectory tracking*, Control Engineering Practice, 2008, nr 16, 1354-1363.

- [76] Martins F. N., Sarcinelli-Filho M., Carelli R.: *A Velocity-Based Dynamic Model and Its Properties for Differential Drive Mobile Robots*, Journal of Intelligent & Robotic Systems, 2017, nr 85(2), 277-292.
- [77] Mehdi Fateh M., Asghar Arab A.: *Voltage control strategy for an uncertain mobile robot*, International journal of intelligent computing and cybernetics, 2014, Vol.7 (4), s. 436-452
- [78] Mehrjerdi H., Ghommam J., Saad M.: *Nonlinear coordination control for a group of mobile robots using a virtual structure*, Mechatronics, 2011, nr 21, s. 1147-1155.
- [79] Meng X., He P., Xiong X., Mi H., Li Z., Wu Z.: *Distributed Consensus Algorithm for Nonholonomic Wheeled mobile Robot*, Security and Communication Networks, 2021, s.1-9
- [80] Miyata N., Ota J., Arai T., Asama H.: *Cooperative Transport by Multiple Mobile Robots in Unknown Static Environments Associated With Real-Time Task Assignment*, IEEE Transaction on Robotics and Automation, 2002, vol. 18, nr 5, s. 769-780.
- [81] More J. J.: *The Levenberg-Marquardt Algorithm: Implementation and Theory*, Numerical Analysis, ed. G. A. Watson, Lecture Notes in Mathematics 630, Springer Verlag, 1977, 105–116.
- [82] Ou M., Du H., Li S.: *Finite-time formation control of multiple nonholonomic mobile robots*, International Journal of Robust and Nonlinear Control, 2014, nr 24(1), s. 140-165.
- [83] Ou M., Sun H., Gu S., Zhang Y.: *Distributed finite-time trajectory tracking control for multiple nonholonomic mobile robots with uncertainties and external disturbances* International Journal of Systems Science, 2017, nr 48 (15), s. 3233-3245
- [84] Panasiuk J., Siwek M., Kaczmarek W., Borys S., Prusaczyk P.: *The concept of using the mobile robot for telemechanical wires installation in pipelines*, International Conference Mechatronic Systems and Materials, MSM 2018 (materiały konferencyjne), 2018, nr 2029, nr art. 020054, doi: 10.1063/1.5066516
- [85] Panasiuk J., Siwek M., *Koncepcja rozproszonego sterowania grupą robotów mobilnych*, Przegląd Mechaniczny, nr 12, 2018, s. 20-24
- [86] Peng Z., Wen G., Yang S., Rahmani A.: *Distributed consensus-based formation control for nonholonomic wheeled mobile robots using adaptive neural network*, Nonlinear dynamics, 2016, nr 86, 605-622
- [87] Peng Z., Yang S., Wen G., Rahmani A., Yu Y.: *Adaptive distributed formation control for multiple nonholonomic wheeled mobile robots*, Neurocomputing, 2016, nr 173, s. 1485-1494
- [88] Plathottam S. J., Ranganathan P.: *Next generation distributed and networked autonomous vehicles: Review*, 10th International Conference on Communication Systems and Networks COMSNETS, 2018, nr. 2018, s. 577-582
- [89] Qu J., Ji Z., Shi Y.: *The Graphical Conditions for Controllability of Multiagent Systems under Equitable Partition*, IEEE Transactions on Cybernetics, 2021, nr. 51(9), s. 4661-4672.
- [90] Ragulski M., Lu Ch., Cao M., Song G., Burdzik R.: *Fault Identification, Diagnosis, and Prognostics Based on Complex Signal Analysis*, Complexity, 2018, nr 2018, nr art.4020729, doi: 10.1155/2018/4020729
- [91] Ren W., Sorensen N.: *Distributed coordination architecture for multi-robot formation control*, Robotics and Autonomous Systems, 2008, nr 56, s. 324-333.
- [92] Roboty KUKA na Ukrainie, <https://prom.ua/Roboty-kuka.html>, dostęp 08.12.2022
- [93] Rykała Ł., Typiak R., Typiak A.: *Analiza wrażliwości neuronowego modelu w aspekcie kształtowania systemu lokalizacji przewodnika*, Wiedza i Innowacje – wiWAT 2019, Wydawnictwo Wojskowej Akademii Technicznej, 2020.

- [94] Saidonr M. S., Desa H., Rudzuan M. N.: *A differentia steering control with proportional controller for an autonomous mobile robot*, IEEE International Colloquium on Signal Processing and its Applications (Materiały konferencyjne), 2011, 90-94.
- [95] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: *Global Sensitivity Analysis. The Primer*, John Wiley and Sons, 2008.
- [96] Shiri H., Wodecki J., Ziętek B., Zimroz R.: *Inspection robotic UGV platform and the procedure for an acoustic signal-based fault detection in belt conveyor idler*, Energies, 2021, nr 24 (22), nr art. 7646, doi: 10.3390/en14227646
- [97] Siegwart R., Nourbakhsh I. R.: *Introduction to Autonomous Mobile Robots*, The MIT Press, Cambridge, Massachusetts, 2004
- [98] Simeon T., Leroy S., Laumond J. P.: *Path coordination for multiple mobile robots: A resolution complete algorithm*, IEEE Trans. Robot. Autom., 2002., vol. 18, nr 1, s. 42–49.
- [99] Simmons R., Singh S., Hershberg D., Ramos J., Smith T.: *First results in the coordination of heterogeneous robots for large-scale assembly*, Experimental Robotics VII, 2001, s. 323-332
- [100] Siwek M., Baranowski L., Panasiuk, J., Kaczmarek W., Besseghieur K.: *The effect of control signals time delays on the trajectory of a dispersed group of robots*, Engineering Mechanics, 2018, vol. 24, s. 757-760, doi: 10.21495/91-8-757
- [101] Siwek M., Besseghieur K., Baranowski, L.: *The effects of the swarm configuration and the obstacles placement on control signals transmission delays in decentralized ROS-embedded group of mobile robots*, International Conference Mechatronic Systems and Materials (Materiał konferencyjne AIP), 2018, nr art. 020069, doi: 10.1063/1.5066531
- [102] Siwek M., Panasiuk J., Baranowski L., Kaczmarek W., Borys S.: *Trajectory Tracking Control of a Mobile Robot with the ROS System*, Problems of Mechatronics. Armament, Aviation, Safety Engineering, 2022, vol. 13 (4), s. 67-84, doi: 10.5604/01.3001.0016.1458
- [103] Siwek M., Panasiuk J., Baranowski L.: *An analysis of the interactions between individual robots in a decentralized multi-robot transport system during movement initiation*, Technical Sciences, 2019, vol. 22, nr. 4, s. 113-123, doi: 10.31648/ts.5130
- [104] Siwek, M., Baranowski, L., Panasiuk, J., Kaczmarek, W.: *Modeling and simulation of movement of dispersed group of mobile robots using Simscape multibody software*, Conference on Computational Technologies in Engineering (Materiały konferencyjne AIP), 2018, nr art. 020045, doi: 10.1063/1.5092048
- [105] Siwek, M., Panasiuk, J., Baranowski, L., Kaczmarek W., Prusaczyk, P., Borys, S.: *Identification of Differential Drive Robot Dynamic Model Parameters*, Materials, 2023, vol. 16(2), nr art. 683, doi: 10.3390/ma16020683
- [106] Stój J., Kampen A., Cupek R., Smółka I., Drewniak M.: *Industrial Shared Wireless Communication Systems—Use Case of Autonomous Guided Vehicles with Collaborative Robot*, Sensors, 2023, nr 23, nr art. 158, doi: 10.3390/s23010158
- [107] Szrek J., Jakubiak J., Zimroz R.: *A Mobile Robot-Based System for Automatic Inspection of Belt Conveyors in Mining Industry*, 2022, Energies, nr 15, nr art. 327, doi: 10.3390/en15010327
- [108] Takahashi H.: *Autonomous decentralized control for formation of multiple mobile robots considering ability of robot*, IEEE Trans. On Industrial Electronics, 2004, nr 51, s.1272-1279.
- [109] Timofeev A., Daeef, F.: *Control of a transport robot in extreme weather based on computer vision systems*, Journal of Physics: Conference Series, 2021, nr 2131, nr art. 032027, doi: 10.1088/1742-6596/2131/3/032027

- [110] Trybała P., Szrek J., Dębogórski B., Ziętek B., Blachowski J., Wodecki J., Zimroz R.: *Analysis of Lidar Actuator System Influence on the Quality of Dense 3D Point Cloud Obtained with SLAM*, Sensors, 2023, nr 23, nr art. 721, doi: 10.3390/s23020721
- [111] Trybała P., Szrek J., Remondino F., Wodecki J., Zimroz R.: *Calibration of a multi-sensor wheeled robot for the 3D mapping of underground mining tunnels*, 2022, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives (materiały konferencyjne), nr 48, s. 135-142, doi: 10.5194/isprs-archives-XLVIII-2-W2-2022-135-2022
- [112] Utstumo T., Berge T. W., Gravdahl J. T.: *Non-linear Model Predictive Control for constrained robot navigation in row crops*, IEEE International Conference on Industrial Technology (Materiały konferencyjne), 2015, 357-362.
- [113] Uzunovic T., Sabanovic A.: *Formation control of differential-drive mobile robots in the framework of functionally related systems*, 41st Annual Conference of the IEEE Industrial Electronics Society, IECON, 2015, nr. artykułu 7392496
- [114] Wan W., Shi B., Wang Z., Fukui R.: *Multirobot Object Transport via Robust Caging*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, nr. 50, s. 270-280
- [115] Wang M., Luo J., Walter U.: *A non-linear model predictive controller with obstacle avoidance for a space robot*, Advances in space research, 2016, nr 57(8), 1737-1746.
- [116] Wang Q., Wu M., Huang Y., Wang L.: *Formation Control of Heterogeneous Multi-Robot Systems*, Proceedings of 17th World Congress The International Federation of Automatic Control, 2008, nr 41, t. 2, s. 6596-6601.
- [117] Wang Y., Ye Y., Zhang H.: *A stable tracking control method for an Autonomous Welding Mobile Robot*, Applied Mechanics and Materials, 2011, nr. 79, s. 264-269.
- [118] Wayne W. D.: *Kendall's tau*, Applied Nonparametric Statistics (2nd ed.), Boston, 1990, s. 365-377.
- [119] Wei W., Huilin T.: *Reconfigurable multi-robot system kinematic modeling and motion planning*, Industrial Electronics and Applications, 2011, s. 1672-1677.
- [120] Wei Z., Li G., Qi L.: *New quasi-Newton methods for unconstrained optimization problems*, Applied Mathematics and Computation, 2006, nr. 175, s. 1156-1188, doi: 10.1016/j.amc.2005.08.027
- [121] Wiśniewski M.: *Propozycja metody pomiaru dokładności i powtarzalności pozycjonowania robotów przemysłowych w warunkach przemysłowych*, Technologia i Automatyzacja montażu, 2004, nr 3, s. 39-43.
- [122] Wu W., You J., Zhang Y., Li M., Su K.: *Parameter identification of Spatial-Temporal Varying Processes by a multi-robot system in realistic diffusion fields*, Robotica, 2021, vol. 39, 842-861, doi:10.1017/S0263574720000788
- [123] Yang F., Wang C., Jing G.: *Adaptive tracking control for dynamic nonholonomic mobile robots with uncalibrated visual parameters: ADAPTIVE TRACKING CONTROL FOR NONHOLONOMIC MOBILE ROBOTS*, International journal of adaptive control and signal processing, 2013, Vol.27 (8), s.688-700
- [124] Yang W., Zimroz R., Papaalias M.: *Advances in Machine Condition Monitoring and Fault Diagnosis*, Electronics, 2022, nr 11, nr art. 1563, doi: 10.3390/electronics11101563
- [125] Ye H., Wang S.: *Trajectory Tracking Control for Nonholonomic Wheeled Mobile Robots with External Disturbances and Parameter Uncertainties*, International Journal of Control, Automation and Systems, 2020, 18(12), nr. 3015-3022.
- [126] Ying T., Zhong-Yang Z.: *Research Advanced in Swarm Robotics*, Defence technology, 2013, vol. 9(1), s. 18-39.
- [127] Zeliaś A.: *Metody statystyczne*, Polskie Wydawnictwo Ekonomiczne, 2000

- [128] Zhang R., Chi R., Hou Z.: *Consensus tracking of multi-agent Systems with Time-delays using Adaptive iterative learning Control*, IEEE 6th Data Driven Control and learning Systemes Conference, 2017, s. 281-285
- [129] Zhang Y., Hong D., Chung J. H., Velinski S. A.: *Dynamic Model Based Robust Tracking Control of a Differential Steered Wheeled Mobile Robot*, Materiały konferencyjne American Control Conference, 1998, s. 850-855.
- [130] Zimroz P., Trybała P., Wróblewski A., Góralczyk M., Szrek J., Wójcik A., Zimroz R.: *Application of UAV in search and rescue actions in underground mine—A specific sound detection in noisy acoustic signal*, Energies, 2021, nr 14, nr art. 3725, doi: 10.3390/en14133725
- [131] Życzkowski M., Stawicki Ł.: *Wpływ opóźnień w sieciach IP na skuteczność monitoring wizyjnego (część I)*, Zabezpieczenia, 2009, nr 2, s. 36-40.
- [132] Życzkowski M., Stawicki Ł.: *Wpływ opóźnień w sieciach IP na skuteczność monitoring wizyjnego (część II)*, Zabezpieczenia, 2009, nr 3, s. 40-48.