



Politechnika Wrocławska

POSTĘPY ROBOTYKI

Pod redakcją Alicji Mazur
i Cezarego Zielińskiego



OFICyna WYDAWNICZA POLITECHNIKI WROCŁAWSKIEJ
WROCŁAW 2025

POLITECHNIKA WROCŁAWSKA

POSTĘPY ROBOTYKI



WROCŁAW 2025

OFICyna WYDAWNICZA POLITECHNIKI WROCŁAWSKIEJ

Recenzenci

Krzysztof Arent, Jacek Bałchanowski, Dominik Belter, Grzegorz Bocewicz, Andrzej Burghardt, Jacek Buśkiewicz, Andrzej Chmielniak, Mateusz Cholewiński, Ryszard Dindorf, Wojciech Domski, Ignacy Dulęba, Mirosław Galicki, Grzegorz Granosik, Zenon Hendzel, Przemysław Herman, Janusz Jakubiak, Włodzimierz Kasprzak, Wojciech Kowalczyk, Bogdan Kreczmer, Ryszard Leniowski, Paweł Malczyk, Alicja Mazur, Maciej Michałek, Robert Muszyński, Michał Olinski, Dariusz Pazderski, Leszek Podseńkowski, Joanna Ratajczak, Adam Ratajczak, Elżbieta Roszkowska, Jerzy Sąsiadek, Karol Seweryn, Barbara Siemiątkowska, Piotr Skrzypczyński, Jarosław Szrek, Wojciech Szynkiewicz, Marek Wojtyra, Katarzyna Zadarnowska, Teresa Zielińska, Cezary Zieliński, Igor Zubrycki

Redakcja naukowa

Alicja Mazur, Cezary Zieliński

Redakcja techniczna

Stanisław Gancarz, Filip Dyba

Realizacja okładki

Robert Muszyński

Wydrukowano na podstawie materiałów dostarczonych przez Autorów

Wszelkie prawa zastrzeżone. Niniejsza książka, zarówno w całości, jak i we fragmentach, nie może być reprodukowana w sposób elektroniczny, fotograficzny i inny bez zgody wydawcy i właścicieli praw autorskich.

© Copyright by Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2025

OFICyna WYDAWNICZA POLITECHNIKI WROCLAWSKIEJ

Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

<http://www.oficyna.pwr.wroc.pl>

e-mail: oficwyd@pwr.wroc.pl

zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-910-2

https://doi.org/10.37190/Postepy_Robotyki_2025

Druk i oprawa: beta-druk, www.betadruk.pl

Spis treści

Słowo wstępne	1
-------------------------	---

Systemy wizyjne i sensoryczne

Odporność ultradźwiękowego czujnika odległości i kierunku przylotu echa na błędy pomiaru czasu	5
<i>Bogdan Kreczmer</i>	
Recent advances in image captioning	15
<i>Gurjinder Singh</i>	
Zastosowanie kamery hiperspektralnej w nawigacji robotów mobilnych	29
<i>Barbara Siemiątkowska, Julian Żebrecki, Mateusz Małkowski</i>	
Estymacja prędkości liniowej i kątowej na podstawie odczytów skanera laserowego 2D z wykorzystaniem sztucznych sieci neuronowych	39
<i>Wojciech Domski, Joanna Chrobot</i>	

Teoria maszyn i mechanizmów

Praca układu sił wykonana przy przemieszczeniu mobilnego robota z kołami mecanum	53
<i>Marcin Szuster, Mateusz Szeremeta</i>	
Zastosowanie algorytmu SLAM w realizacji zadania typu "podążaj do celu"	63
<i>Marcin Szuster, Mateusz Szeremeta, Paweł Penar</i>	

Sterowanie robotów

Sterownik czasu ustalonego dla monocykla w pozycyjnym zadaniu odtwarzania ścieżki	75
<i>Maciej Michałek, Mikołaj Przybylski, Rafał Sobański</i>	
Organizacja przerywania działań robotów	85
<i>Cezary Zieliński</i>	
Sterowanie dwuosiowym manipulatorem maksymalizujące siłę sygnału w systemie komunikacji laserowej	95
<i>Patryk Bartkowiak, Dariusz Pazderski</i>	
Sterowanie z ograniczonym momentem w zadaniu śledzenia ścieżki manipulatora	105
<i>Filip Dyba, Alicja Mazur</i>	

Interakcje robot-człowiek

Selecting the number of key postures for an unsupervised method of human action recognition	117
<i>Esther Elemeyong, Teresa Zielińska, Vibekananda Dutta</i>	
Recognition of human actions by analyzing the trajectories of moved objects . .	127
<i>Jakub Kamiński, Vibekananda Dutta, Teresa Zielińska</i>	
Badania potrzeb informacyjnych użytkowników systemów AMR	137
<i>Jakub Strzałka, Grzegorz Granosik</i>	

Planowanie ruchu robotów

Analityczne wsparcie metody pól potencjałów w zadaniach planowania ruchu .	149
<i>Ignacy Dulęba</i>	
Multi-agent systems - analysis of two-robot deadlocks	159
<i>Wojciech Kowalczyk, Arpit Joon</i>	
Kubiczna interpolacja orientacji w przestrzeni rzutowej	171
<i>Jacek Grzegorzewski</i>	

Perspektywy robotyki

Wyzwania i perspektywy rozwoju technologii robotycznych w kontekście zrównoważonego budownictwa	185
<i>Ryszard Dindorf, Jakub Takosoglu, Piotr Woś</i>	
Wybrane zagadnienia transformacji prędkości do celów sterowania w robotyce .	195
<i>Przemysław Herman</i>	
Zastosowanie manipulatora UR3 w nauczaniu podstaw teoretycznych robotyki .	205
<i>Krzysztof Arent, Katarzyna Zadarnowska, Jakub Kusz, Jakub Malewicz, Dawid Rogaliński</i>	
Przegląd bezzałogowych platform w zastosowaniach wojskowych	217
<i>Jakub Malewicz, Krzysztof Górski</i>	

Roboty medyczne i rehabilitacyjne

Doświadczalna weryfikacja dokładności procedury pomiaru LLD i LOD za pomocą ramienia pomiarowego NaviFast 6D	229
<i>Agnieszka Kobierska, Patryk Skwierczyński, Kacper Sęk, Piotr Kacprzak, Leszek Podśędkowski</i>	
A vision based module for implementing interactive mirror-like therapy using Stretchbox device	239
<i>Igor Zubrycki, Kamila Kuczmarska</i>	

Układ sterowania systemem wspomagającym biopsję	249
<i>Adam Kurnicki, Bartłomiej Stańczyk</i>	

Indeks autorów	259
--------------------------	-----

Słowo wstępne

Drodzy Czytelnicy,

Z wielką przyjemnością oddajemy kolejny Zeszyt naukowy z cyklu **Postępy robotyki**, który przedstawia aktualny i wiarygodny stan badań w dziedzinie robotyki w Polsce w 2025 roku. Począwszy od 1985 roku Zeszyty stanowią publikacje poświęcone robotyce, wydawane najpierw przez Oficynę Wydawniczą Politechniki Wrocławskiej, następnie przez Wydawnictwa Komunikacji i Łączności, a w ostatnich kilkunastu latach także przez Oficynę Wydawniczą Politechniki Warszawskiej. Jednak obecny Zeszyt wraca do źródeł i ponownie pojawia się w druku w Oficynie Wydawniczej Politechniki Wrocławskiej. Zgodnie z tradycją, treść Zeszytu nie tylko odzwierciedla aktualny stan badań w dziedzinie robotyki w Polsce, lecz także pozwala prognozować kierunki rozwoju polskiej robotyki w najbliższej przyszłości.

W obecnym Zeszycie, podobnie jak miało to miejsce w poprzednim tomie z 2022 roku, przedstawiono również wyniki badań prezentowane przez grono członków i sympatyków Komitetu Budowy Maszyn Polskiej Akademii Nauk, które także wnoszą istotny wkład w rozwój polskiej robotyki.

W aktualnym Zeszycie zawarto 23 prace zamieszczone w 7 rozdziałach:

1. Systemy wizyjne i sensoryczne.
2. Teoria maszyn i mechanizmów.
3. Sterowanie robotów.
4. Interakcje robot-człowiek.
5. Planowanie ruchu robotów.
6. Perspektywy robotyki.
7. Roboty medyczne i rehabilitacyjne.

Prace te przedstawiają wyniki badań o charakterze podstawowym, jak i badania stosowane realizowane w ramach pozyskanych grantów. Rozkład geograficzny i przekrój pokoleniowy autorów **Postępów robotyki** jest miarodajny dla oceny aktywności różnych ośrodków naukowych, a także osiągnięć młodego pokolenia polskich robotyków.

Zeszyt zawiera prace zarekomendowane do druku przez recenzentów będących autorytetami naukowymi w dziedzinie robotyki. Pragniemy podziękować wszystkim Autorom prac za powierzenie nam do publikacji najnowszych wyników ich badań. Tradycyjnie, każda praca była poddana ocenie dwóch recenzentów. Znając wnikliwość wielu recenzji i zaangażowanie recenzentów, chcemy wyrazić wielkie uznanie dla ich dokonań. Pragniemy podziękować autorom prac za współpracę z recenzentami, której wynikiem jest poziom merytoryczny prac zamieszczonych w Zeszycie.

Za prace redakcyjne nad Zeszytem składamy podziękowania panu mgrowi Filipowi Dybie oraz drowi Robertowi Muszyńskiemu z Katedry Cybernetyki i Robotyki na Wydziale Elektroniki, Fotoniki i Mikrosystemów Politechniki Wrocławskiej.

Alicja Mazur, Cezary Zieliński

Wrocław-Warszawa, lipiec 2025

SYSTEMY WIZYJNE I SENSORYCZNE

Odporność ultradźwiękowego czujnika odległości i kierunku przylotu echa na błędy pomiaru czasu

Bogdan Kreczmer¹

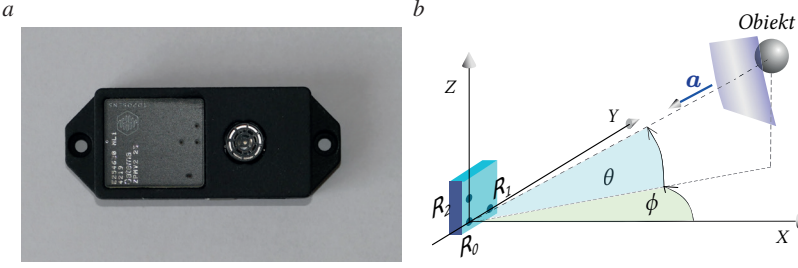
Streszczenie

Referat omawia problem wyznaczenia kierunku przylotu sygnału ultradźwiękowego w kontekście metod bazujących na pomiarze przesunięcia fazy sygnału. Problem ten jest kluczowy dla precyzyjnej lokalizacji obiektów w otoczeniu robota z wykorzystaniem czujników ultradźwiękowych. Jego rozwiązanie wymaga określenia właściwej geometrii rozmieszczenia układu odbiorników. Najlepsze rezultaty można otrzymać jedynie z zastosowaniem mikrofonów MEMS. W referacie przedstawiono wpływ wspomnianej geometrii na odporność układu odbiorczego na błędy pomiaru czasu detekcji wybranej fazy sygnału. Określono też warunki optymalnej geometrycznej konfiguracji odbiorników. Opierając się na tych warunkach zaproponowano układ ich rozmieszczenia, który daje satysfakcjonującą odporność. Co więcej, zapewnia on ją w szerokim zakresie kątów kierunku przylotu sygnału. Przedstawiony został ponadto aktualny stan realizacji konstrukcji modułu czujnika ultradźwiękowego odległości i kierunku przylotu sygnału.

1. Wprowadzenie

Ultradźwiękowe czujniki odległości stosowane w robotyce zwykło się uważać za mało wiarygodne lecz proste w konstrukcji i tanie. Opinia ta jest jak najbardziej słuszna w stosunku do popularnych czujników tego typu dostępnych na rynku. W przypadku bardziej ogólnym nie jest to już jednak prawdą. Wszystko zależy od konstrukcji i użytej metody. Ich właściwy wybór czyni z czujników ultradźwiękowych źródło precyzyjnej informacji. Przykładem jest system Quantum RTLS 2.0 [12] oferowany przez firmę ZeroKey. Pozwala on na lokalizowanie i śledzenia ruchu dłoni w trakcie realizacji różnych procesów technologicznych lub ruchu wybranych obiektów. Dokładność lokalizacji wynosi 1,5 mm. Jest to system aktywny bazujący na lokalizacji położenia nadajników ultradźwiękowych przymocowanych do ręki pracownika lub poruszającego się obiektu. Innym przykładem przemysłowej konstrukcji jest ultradźwiękowy czujnik położenia obiektów ECHO ONE [11]. Pozwala on precyzyjnie określić lokalizację obiektu w najbliższym otoczeniu robota, a będąc bardziej precyzyjnym, miejsce odbicia się sygnału. Możliwe jest to w dużej mierze dzięki zastosowaniu mikrofonów ultradźwiękowych MEMS. Ten typ odbiorników znalazł zastosowanie w wielu konstrukcjach badawczych [2, 3, 8, 6, 7, 10]. Konstrukcje te okazały się na tyle udane, że zostały wykorzystane do implementacji algorytmów SLAM [9]. Ich wadą jest wykorzystanie dużej ilości mikrofonów w celu stworzenia odpowiedniej tablicy fazowej. Rozwiązanie analizowane w niniejszym referacie minimalizuje ich liczbę. Jed-

¹Katedra Cybernetyki i Robotyki Wydziału Elektroniki, Fotoniki i Mikrosystemów Politechniki Wrocławskiej, bogdan.kreczmer@pwr.edu.pl, <http://kcir.pwr.edu.pl>



Rys. 1. Elementarny układ odbiorników pomiaru kierunku przylotu sygnału
a – ultradźwiękowy czujnik TS3, b – współrzędne horyzontalne kierunku przylotu sygnału odbitego od obiektu

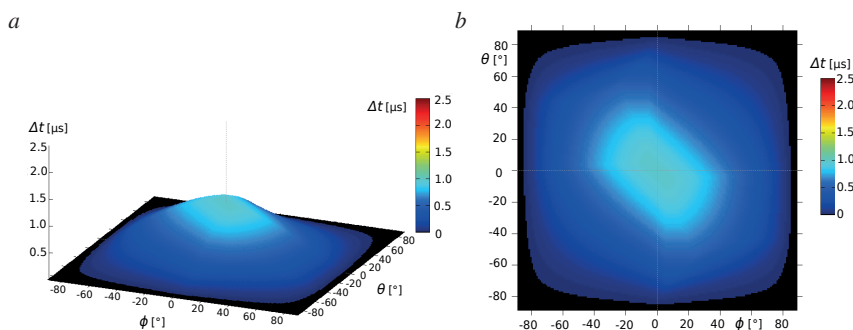
nocześnie jest w stanie zapewnić satysfakcjonującą tolerancję błędów pomiaru czasu detekcji odbieranego sygnału i dokładność wyznaczenia kierunku przylotu sygnału.

2. Wyznaczanie kierunku przylotu sygnału

Układ odbiorczy składających się z trzech odbiorników ułożonych niewspółliniowo jest minimalnym układem, który pozwala na wyznaczenie kierunku przylotu sygnału w przestrzeni trójwymiarowej. Jeżeli taki układ zawiera dwie pary odbiorników, których wzajemne odległości są mniejsze niż pół długości fali nośnej sygnału, to umożliwia on jednoznaczne wyznaczenie kierunku przylotu sygnału. Największą dokładność wyznaczenia współrzędnych horyzontalnych wspomnianego kierunku zapewnia układ odbiorczy, którego odbiorniki zostały rozłożone na obrysie prostokątnego trójkąta równoramiennego (szczegółowa argumentacja dowodząca tej własności znajduje się w dalszej części niniejszego opracowania). Przykładem systemu sensorycznego o takiej geometrii jest ECHO ONE [11]. Jest to przemysłowy system detekcji przeszkód dedykowany dla platform mobilnych. Oferowany jest on przez firmę Toposens. Wcześniej firma ta oferowała prostszy układ TS3 (zobacz rys. 1a). Przemysłowa wersja ECHO ONE choć mocno rozbudowana, to jednak zachowała identyczny układ odbiorników w postaci mikrofonów MEMS. Dla wspomnianej wyżej geometrii i odległości między odbiornikami mniejszej niż pół długości fali nośnej można wyznaczyć kierunek przylotu sygnału mierząc różnicę czasu pojawienia się tej samej fazy sygnału dla poszczególnych odbiorników. Stosując współrzędne horyzontalne, tzn. kąty azymutu ϕ i elewacji θ (zobacz rys. 1b), ich wartości znajdujemy stosując wzory (1)

$$\begin{cases} \phi = \arcsin -\frac{a_y}{\sqrt{1-a_z^2}}, \\ \theta = \arcsin -a_z. \end{cases} \quad \text{przy czym} \quad \begin{aligned} a_y &= v_a \frac{z_2(t_1-t_0)-z_1(t_2-t_0)}{y_1z_2-y_2z_1}, \\ a_z &= v_a \frac{y_1(t_2-t_0)-y_2(t_1-t_0)}{y_1z_2-y_2z_1}, \end{aligned} \quad (1)$$

gdzie v_a jest prędkością dźwięku, t_0 , t_1 i t_2 to czasy rejestracji sygnału odpowiednio przez odbiornik R_0 , R_1 i R_2 . Przedstawione wzory uwzględniają nieco bardziej ogólny przypadek niż ten, który został przedstawiony na rys. 1b. Zakłada się jedynie, że odbiornik R_0 jest w środku układu współrzędnych i wszystkie pozostałe odbiorniki



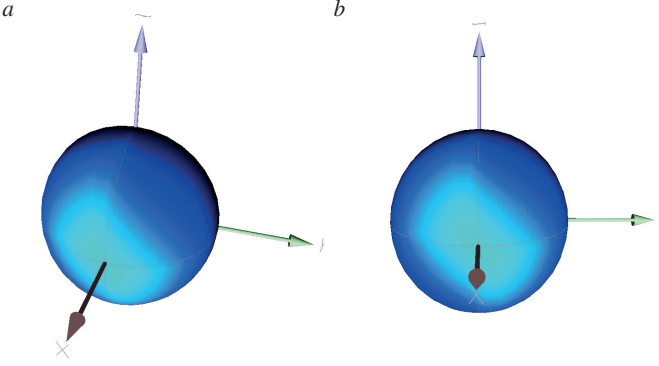
Rys. 2. Rozkład maksymalnej wartości tolerowanego błędu Δt_{max} dla poszczególnych kierunków

a – widok z boku, b – widok z góry

znajdują się na płaszczyźnie YOZ . Tak więc współrzędne kolejnych odbiorników to odpowiednio $(0, 0, 0)$, $(0, y_1, z_1)$ oraz $(0, y_2, z_2)$.

Istotnym parametrem określającym przydatność danej metody wyznaczania kierunku przylotu jest jej odporność na błędy pomiaru czasu przylotu sygnału. Odporność tą można sprawdzić poprzez symulowanie wartości pomiarów dla danego kierunku przylotu. W następnym kroku należy określić maksymalną wartość błędu pomiaru czasu Δt_{max} , dla której wyznaczony kierunek mieści się w zadanych granicach. Bardziej szczegółowy opis tej procedury można znaleźć w [4]. Wspomniane granice określają tolerowany przedział błędu wyznaczenia kierunku przylotu. Należy oczekiwać, że przy zadanej granicy przedziału błędu wyznaczenia kierunku, znalezione wartości Δt_{max} będą różne dla różnych kierunków. Tak więc oprócz samych wartości ważny jest również ich rozkład w przestrzeni współrzędnych kątowych kierunku przylotu sygnału. Opisaną procedurę zastosowano do układu trzech odbiorników o geometrii przedstawionej na rys. 1b, którą charakteryzują się czujników TS3 i ECHO ONE. Jako wartość graniczną tolerowanego błędu wyznaczenia kierunku przylotu przyjęto 10° . Otrzymane rezultaty w przestrzeni współrzędnych kątowych horyzontalnych przedstawione są na rys. 2. Współrzędne kierunków na tym rysunku są przedstawione w odwzorowaniu walcowym równokątnym (odwzorowanie Merkatora). Dla lepszego intuicyjnego zrozumienia tych diagramów zostały one rozpięte na hemisferze (zobacz rys. 3). Pozwala to na bezpośrednie zobrazowanie poszczególnych kierunków. Przedstawione diagramy obrazują mocną zależność wartości Δt_{max} od kierunku przylotu. Jest ona największa dla kierunku na wprost i wynosi $0,98 \mu s$. Szybko jednak maleje wraz z oddalaniem się od tego kierunku. Można też zauważyć, że wykres ten nie zachowuje symetrii zarówno względem płaszczyzny zerowego azymutu, jak też płaszczyzny zerowej wartości kąta elewacji.

Zmniejszenie wpływu błędów pomiaru czasu przylotu sygnału na wartości wyznaczanych kątów można osiągnąć poprzez zwiększenie odległości między odbiornikami. W tym przypadku prowadziłoby to jednak do przekroczenia ograniczenia wyznaczonego przez pół długości fali emitowanego sygnału. Dla częstotliwości sygnału 40 kHz i temperatury powietrza równej $20^\circ C$ jest to ok. 4,3 mm. W [5] przedstawiono metodę pozwalającą rozwiązać ten problem dla przypadku 2D ograniczonego do wy-



Rys. 3. Diagramy wartości Δt_{max} dla poszczególnych kierunków przylotu sygnału rozpięty na hemisferze w dwóch przykładowych widokach

znaczenia kąta azymutu. Jej rozszerzenie na przypadek 3D (wyznaczenie odległości oraz kątów azymutu i elewacji) został przedstawione w [4]. W opracowaniu tym pokazano, że dla kierunku na wprost osiągnięto tolerancję błędu pomiaru na poziomie $1,8 \mu s$. Jednak taka wysoka tolerancja została osiągnięta dla stosunkowo niewielkiego zakresu kątów (ok. $[-10^\circ, 10^\circ]$ zarówno dla azymutu, jak też kąta elewacji). Poza tym obszarem obserwowany jest gwałtowny spadek. Wynika on z zastosowanej geometrii rozmieszczenia odbiorników, która dla wariantu 2D była bardzo korzystna, jednak dla przypadku 3D już nie.

Błąd wyznaczenia kątów kierunku przylotu sygnału jest determinowany przez błąd wyznaczenia znormalizowanych współrzędnych wektora tego kierunku. Zgodnie z (1) wartości kątów ϕ i θ zależą bezpośrednio od współrzędnych a_y i a_z . Posługując się metodą różniczki zupełnej, błędy wyznaczenia ϕ i θ możemy oszacować jako

$$\Delta\phi = \left| \frac{\partial\phi}{\partial a_y} \right| \Delta a_y + \left| \frac{\partial\phi}{\partial a_z} \right| \Delta a_z, \quad \Delta\theta = \left| \frac{\partial\theta}{\partial a_y} \right| \Delta a_y + \left| \frac{\partial\theta}{\partial a_z} \right| \Delta a_z.$$

Wykonując niezbędne operacje pozwala to otrzymać następujące wyrażenia

$$\Delta\phi = \frac{(1 - a_z^2)\Delta a_y + 2 |a_y a_z| \Delta a_z}{(1 - a_z^2) \sqrt{1 - a_z^2 - a_y^2}}, \quad \Delta\theta = \frac{\Delta a_z}{\sqrt{1 - a_z^2}}. \quad (2)$$

Ponownie korzystając ze wzorów (1) i metody różniczki zupełnej możemy wyrazić błąd obliczenia współrzędnych wektora kierunku poprzez błędy pomiaru czasów t_0 , t_1 i t_2 . Zasadnym jest uproszczenie i przyjęcie, że błędy te są jednakowe i wynoszą Δt . Pozwala to otrzymać następujące oszacowanie:

$$\Delta a_y = v_a \frac{|z_1 - z_2| + |z_2| + |z_1|}{|y_1 z_2 - y_2 z_1|} \Delta t, \quad \Delta a_z = v_a \frac{|y_2 - y_1| + |y_1| + |y_2|}{|y_1 z_2 - y_2 z_1|} \Delta t. \quad (3)$$

Pragnąc określić jak błędy $\Delta\phi$ i $\Delta\theta$ zależą od wzajemnego ułożenia odbiorników dla niezmiennych odległości między R_0 i R_1 oraz R_0 i R_2 przyjmujemy wspomniane

odległości jako takie same i równe d . Chcąc uprościć dalszą analizę, jako współrzędne poszczególnych odbiorników zostaną przyjęte odpowiednio $R_0 = (0, 0, 0)$, $R_1 = (0, d, 0)$ oraz $R_2 = (0, d \cos \alpha, d \sin \alpha)$ gdzie $\alpha \in (0, \pi)$. Układ ten zobrazowano na rys. 4a. Mając na uwadze (1), współrzędne wektora kierunkowego a_y i a_z możemy zapisać jako

$$a_y = \frac{v_a(t_1 - t_0)}{d}, \quad a_z = v_a \frac{(t_2 - t_0) - (t_1 - t_0) \cos \alpha}{d \sin \alpha}.$$

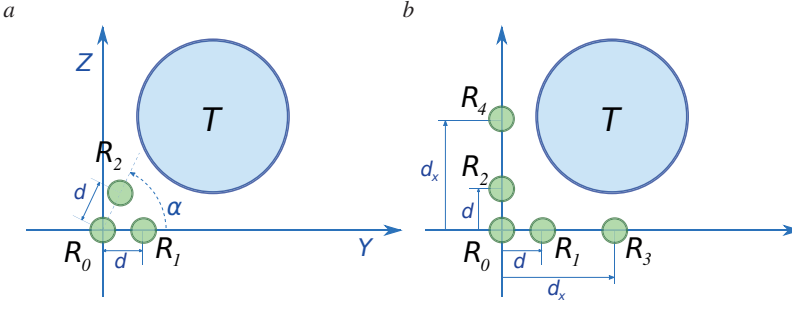
Pozwala to przekształcić wzory (3) do postaci

$$\Delta a_y = \frac{2v_a}{d} \Delta t, \quad \Delta a_z = v_a \frac{|\cos \alpha - 1| + 1 + |\cos \alpha|}{d \sin \alpha} \Delta t. \quad (4)$$

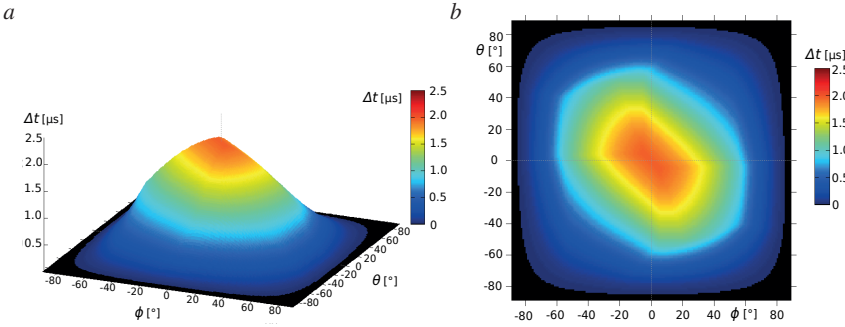
Otrzymany wynik pozwala stwierdzić, że Δa_y nie zależy od kąta α . Tym samym nie zależy od położenia katowego odbiornika R_2 . Postać wyrażenia dla Δa_z umożliwia zauważyć, że przy zadanym Δt najmniejszą jego wartość otrzymujemy dla kąta 90° , tzn. gdy odbiorniki są ułożone na planie równobocznego trójkąta prostokątnego. Wnioski otrzymane dla współrzędnej Δa_z przekładają się natychmiast na analogiczny stwierdzenie dla $\Delta \theta$. Wynika to z tego, że wzór (3) dla $\Delta \theta$ pokazuje, że dla każdego dopuszczalnego kierunku, a więc w tym przypadku każdej dopuszczalnej wartości współrzędnej a_z , wartość $\Delta \theta$ jest najmniejsza, gdy Δa_z jest najmniejsze. Ze względu na to, że Δa_y nie zależy od α , identyczny wniosek otrzymujemy również w przypadku wartości $\Delta \phi$. Podsumowując, przeprowadzone rozumowanie dowodzi następującego twierdzenia.

Zakładając, że dwa odbiorniki znajdują się w płaszczyźnie horyzontalnej, to ułożenie odbiorników na planie prostokątnego trójkąta jest ułożeniem optymalnym pod względem minimalizacji wpływu błędu pomiaru czasu detekcji tej samej fazy sygnału na błąd wyznaczenia współrzędnych horyzontalnych kierunku przylotu sygnału.

Mając to na uwadze, położenie dodatkowych odbiorników również musi być zrealizowane na planie trójkąta prostokątnego, aby minimalizować błędy $\Delta \phi$ i $\Delta \theta$. Nowa zaproponowana geometria rozkładu odbiorników ma więc postać przedstawioną na rys. 4b. Dodatkowe odbiorniki zostały zamontowane w tej samej odległości od środka układu współrzędnych. Zróżnicowanie ich nie prowadzi do nowych rezultatów. Wykonując wcześniej opisaną symulację wyznaczono Δt_{max} dla kolejnych odległości d_x (zobacz rys. 4b) poczynając od 5 mm, a na 30 mm kończąc. W zastosowanej metodzie pośredniej detekcji fazy, aby usunąć niejednoznaczności wyznaczania kątów przylotu sygnału, które powstają gdy odbiorniki znajdują się dalej od siebie niż połowa długości fali, tworzone są podukłady odbiorników. Każdy z nich składa się z 3 odbiorników. Uzyskiwane zbiory kierunków dla tych podsystemów zawsze mają jeden wspólny element, którym jest faktyczny kierunek przylotu sygnału. Błędy pomiarów czasu przylotu ograniczają zakres stosowalności takiego podejścia. Jednak dla odpowiednio dobranej konfiguracji odbiorników można znacząco poprawić własności całego systemu. Dla rozważanego przypadku wybrane zostały dwa podsystemy składające się odpowiednio z odbiorników (R_0, R_1, R_2) oraz (R_0, R_3, R_4) . Największą wartość tolerowanego błędu pomiaru czasu dla kierunku na wprost uzyskano dla odległości d_x



Rys. 4. Geometryczny rozkład odbiorników wraz z nadajnikiem
 a – układ trzech odbiorników, b – układ pięciu odbiorników na planie trójkąta prostokątnego



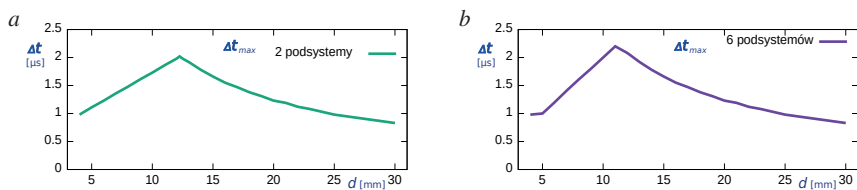
Rys. 5. Rozkład maksymalnej wartości tolerowanego błędu dla poszczególnych kierunków dla $d_x = 12$ mm i zastosowaniu 2 elementarnych układów pomiarowych:
 a – widok z boku, b – widok z góry

równej 12 mm. Wynosiła ona $1,98 \mu s$. Rozkład wartości tolerowanych błędów dla poszczególnych kierunków przedstawiony jest na diagramach na rys. 5. Zmiany wartości Δt_{max} dla kierunku na wprost w zależności od d_x pokazane są na rys. 6a. Przebieg tego wykresu świadczy o istnieniu dwóch przeciwnych mechanizmów. Jeden z nich jest odpowiedzialny za wzrost tolerancji na błąd pomiaru. Efekt działania drugiego z nich jest natomiast przeciwny. Pierwszy ze wspomnianych mechanizmów jest bezpośrednio związany z błędem wyliczenia kierunku przylotu. Rozważając kierunek na wprost, znormalizowane współrzędne wektora dla tego kierunku zgodnie z wybranym układem współrzędnych (zobacz rys. 1b) to $(1, 0, 0)$. Tak więc równania (2) sprowadzają się do postaci

$$\Delta\phi = \Delta a_y, \quad \Delta\theta = \Delta a_z. \quad (5)$$

Mając na uwadze równanie (4) oraz fakt, że wybrana wartość α to 90° , a następnie dokonując podstawienia do równania (5) oraz przekształcając je, aby wyrazić Δt względem błędu wyznaczenia kąta azymutu ϕ oraz elewacji θ otrzymujemy

$$\Delta t_\phi = \frac{\Delta\phi}{2v_a} d, \quad \Delta t_\theta = \frac{\Delta\theta}{2v_a} d.$$



Rys. 6. Zmiany Δt_{max} dla kierunku na wprost w zależności od odległości d_x dodatkowych odbiorników od odbiornika R_0

a – w przypadku wykorzystania 2 elementarnych układów pomiarowych, b – 6 układów

Ze względu na to, że zarówno dla kąta azymutu, jak też dla kąta elewacji została przyjęta ta sama maksymalna wartość tolerancji błędu wyznaczenia kąta, tzn. $\Delta\phi_{max} = \Delta\theta_{max} = 10^\circ$ (wartość ta dalej oznaczana będzie jako $\Delta\gamma_{max}$), otrzymujemy więc $\Delta t_\phi = \Delta t_\theta = \Delta t$ i wartość ta liniowo zależy od odległości d między odbiornikami.

Jak wcześniej zostało powiedziane, w zastosowanej metodzie pośredniej detekcji fazy sygnału do rozważanego przypadku wykorzystuje się dwa podukłady odbiorników, tzn. (R_0, R_1, R_2) oraz (R_0, R_3, R_4) . Maksymalny tolerowany błąd całego systemu jest determinowany przez ten układ, który zapewnia większą dokładność wyznaczenia kąta przylotu sygnału. W tym przypadku będzie to układ (R_0, R_3, R_4) , dla którego odległość d_x jest większa od odległości d między odbiornikami R_0 i R_1 oraz R_0 i R_2 . Z tego też powodu ostatecznie otrzymujemy zależność

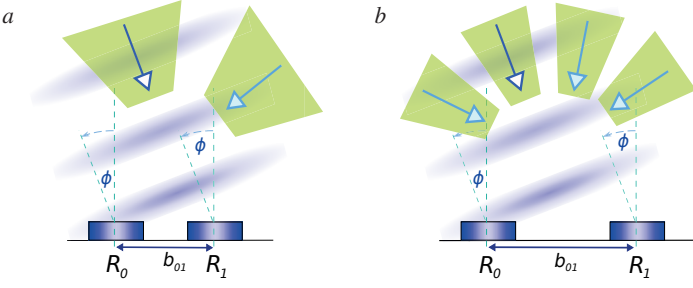
$$\Delta t_{max} = \frac{\Delta\gamma_{max}}{2v_a} d_x.$$

Odzwierciedla ona liniowy wzrost maksymalnej wartości tolerowanego błędu pomiaru w pierwszej części wykresu.

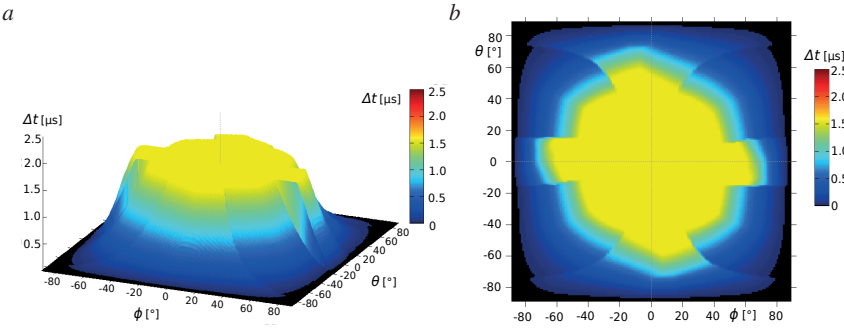
Drugi mechanizm jest nieco bardziej złożony i związany jest z samą naturą metody pośredniej detekcji fazy. Zwiększenie odległości między odbiornikami w drugim z podsystemów prowadzi do pojawienia niejednoznaczności, tzn. większej ilości dopuszczalnych rozwiązań równania (1) niż jedno. Odstęp między poszczególnymi kierunkami wyznacza w tym przypadku maksymalną wartość tolerowanego błędu pomiaru. Wraz ze wzrostem odległości d_x wzrasta ilość dopuszczalnych rozwiązań, tzn. alternatywnych kierunków. To z kolei prowadzi do *zagęszczenia* przestrzeni kierunków i zmniejszenia odstępów (zobacz rys. 7). Tym samym powoduje to zmniejszenia maksymalnej wartości tolerowanego błędu pomiaru, co obserwujemy w drugiej części wykresu przedstawionego na rys. 6a.

Wzięcie dodatkowych elementarnych podsystemów wyznaczania kierunków przylotu sygnału, które można wyróżnić w tej konstrukcji, prowadzi od dalszego zwiększenia wartości Δt_{max} . Po dodaniu elementarnych podsystemów: (R_1, R_3, R_4) , (R_2, R_3, R_4) , (R_0, R_1, R_4) , (R_0, R_2, R_3) ; a więc zwiększenie liczby wszystkich podsystemów do 6, uzyskano zwiększenie Δt_{max} do wartości $2,20 \mu s$. Dodając dwa pozostałe podsystemy, tzn. (R_1, R_2, R_4) i (R_1, R_2, R_3) , można uzyskać dalsze niewielkie zwiększenie do wartości $2,29 \mu s$.

Mimo zwiększenia Δt_{max} wspomniany wcześniej przypadek nie jest najkorzystniejszy. Dalsze zwiększenie odległości d_x , co prawda prowadzi do zmniejszenia war-



Rys. 7. Wzrost ilości dopuszczalnych kierunków jako rozwiązania równania (1) wraz ze wzrostem odległości między odbiornikami sygnału, co powoduje zmniejszenie przedziałów tolerowanych błędów

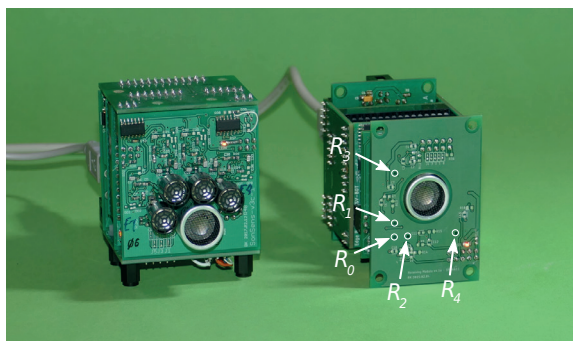


Rys. 8. Rozkład Δt_{max} dla poszczególnych kierunków dla $d_x = 16\text{ mm}$ i zastosowaniu 6 elementarnych układów pomiarowych:
a – widok z boku, b – widok z góry

tości Δt_{max} , jednak z drugiej strony prowadzi do zwiększenia zakresu kątów, dla których obowiązuje dana wartość. Jako kompromisowe rozwiązanie wybrano do praktycznej realizacji konstrukcję, dla której odległość $d_x = 16\text{ mm}$. Dystrybucja wartości Δt_{max} dla tego przypadku przy założeniu wykorzystania 6 podsystemów została przedstawiona na diagramach na rys. 8. Choć wartość Δt_{max} została zredukowana do $1,55\text{ }\mu\text{s}$, to jednak obowiązuje ona w bardzo dużym przedziale kątów dochodzącym do $(-60^\circ, 60^\circ)$.

3. Realizacja ultradźwiękowego czujnika odległości i kierunku przylotu echa

W konstrukcji czujnik, która jest obecnie realizowana, zdecydowano się na wykorzystanie mikrofonów IMP23ABSU. Są to mikrofony MEMS produkowane przez STMicroelectronics. Dostępne są w obudowie RHLGA-5L o wymiarach $3,5 \times 2,65\text{ mm}$. Mając na uwadze, że wykorzystywany sygnał ma częstotliwość 40 kHz , wspomniane wymiary pozwalają na ulokowanie mikrofonów w odległości poniżej pół długości fali tego sygnału. Na rys. 9 przedstawiona jest wcześniejsza konstrukcja wykorzystująca



Rys. 9. Konstrukcja wcześniejszej wersji z odbiornikami piezoelektrycznymi (po lewej stronie) oraz aktualnie tworzonej z mikrofonami MEMS wraz z zaznaczonymi miejscami ich rozmieszczenia

jako odbiorniki, piezoelektryczne przetworniki ultradźwiękowe. Po prawej stronie widoczna jest wersja aktualnej konstrukcji. W momencie pisania tego referatu jest ona poddawana modyfikacjom. Niemniej nowsza wersja tej konstrukcji zasadniczo nie odbiega pod względem szerokości i wysokości od tej, która widoczna jest na zdjęciu. Zostanie jedynie trochę zredukowana jej głębokość, jak też modyfikacji uległo położenie mikrofonów. Ich położenia dla wersji widocznej na zdjęciu dla lepszego zobrazowania zostało odpowiednio zaznaczone, aby ułatwić ich lokalizację na płycie. W obu prezentowanych konstrukcjach, jak też we wspomnianej aktualnej wersji zastosowano moduł mikroprocesorowy Malaga [1], który został opracowany na bazie mikrokontrolera Kinetis MK40DN512VLK10. Jako nadajnik wykorzystano przetwornik piezoelektryczny BPU-1640IOAH12. Szacunkowy pobór mocy to ok. 0,5 W.

4. Wnioski

Dla wariantu metody pośredniej detekcji fazy dla przypadku 2D, jak też dla wariantu 3D, w opracowaniach [4] i [5] udało się potwierdzić zgodność wyników symulacyjnych z faktycznymi pomiarami. Dla kierunków, dla których w trakcie badań symulacyjnych, otrzymano Δt_{max} na poziomie $1 \mu s$, wyliczenia na podstawie rzeczywistych pomiarów wykazywały tylko niewielkie odchylenia od spodziewanych wartości. Można więc na podstawie obecnych wyników symulacyjnych przyjąć, że aktualnie realizowana konstrukcja ultradźwiękowego czujnika odległości i kierunku przylotu echa powinna zapewnić osiągnąć bardzo dobre wyniki w bardzo szerokim zakresie kątów. Wspomniane rezultaty symulacji dla preferowanej geometrii układu odbiorników dają odporność na poziomie nieznacznie powyżej $1,5 \mu s$ dla zakresu $(-60^\circ, 60^\circ)$ zarówno kątów azymutu, jak też elewacji. Biorąc jednak pod uwagę kierunkową charakterystykę nadajnika, która pokazuje znaczne osłabienie emitowanego sygnału dla bardziej odległych kierunków od jego osi akustycznej, jako realny zakres bezpieczniej jest przyjąć $(-40^\circ, 40^\circ)$. Mimo tego ograniczenia należy podkreślić, że jest on dalece większy od tego co zostało przedstawione dla konstrukcji opisanej w [4]. Jest on też niepomniernie większy dla przypadku zastosowania geometrii rozmieszczenia mikrofonów, jaka została zastosowana w czujniku ECHO ONE.

Literatura

- [1] M. Cholewiński et al. Malaga minimodule. Raport instytutowy SPR nr 13/2013, Inst. Inform. Autom. Robot. Politechniki Wrocławskiej, 2013.
- [2] K. Herman, T. Gudra, J. Furmankiewicz. Digital signal processing approach in air coupled ultrasound time domain beamforming. *Archives of Acoustics*, 2014, wolumen 39, numer 1, s. 27–50.
- [3] R. Kerstens, D. Laurijssen, J. Steckel. eRTIS: A fully embedded real time 3d imaging sonar sensor for robotic applications. In: 2019 International Conference on Robotics and Automation (ICRA). *Proceedings*, Maj, 2019, s. 1438–1443.
- [4] B. Kreczmer. DOA estimation of ultrasonic signal by indirect phase shift determination. *Archives of Acoustics*, 2014, Vol. 49, No. 1, s. 107–120.
- [5] B. Kreczmer. Estimation of the azimuth angle of the arrival direction for an ultrasonic signal by using indirect determination of the phase shift. *Archives of Acoustics*, 2019, Vol. 44, No. 3.
- [6] D. Laurijssen, W. Daems, J. Steckel. Broadband MEMS microphone arrays with reduced aperture through 3D-printed waveguides. In: 2024 IEEE SENSORS. *Proceedings*, 2024, s. 1–4.
- [7] D. Laurijssen, W. Daems, J. Steckel. HiRIS: An airborne sonar sensor with a 1024 channel microphone array for in-air acoustic imaging. *IEEE Access*, 2024, wolumen 12, s. 51786–51795.
- [8] J. Steckel, A. Boen, H. Peremans. Broadband 3-d sonar system using a sparse array for indoor navigation. *Robotics, IEEE Transactions on*, 2013, wolumen 29, numer 1, s. 161–171.
- [9] J. Steckel, H. Peremans. Batslam: Simultaneous localization and mapping using biomimetic sonar. *PLOS ONE*, 2013, wolumen 8, numer 1, s. 1–11.
- [10] T. Verellen, R. Kerstens, J. Steckel. High-resolution ultrasound sensing for robotics using dense microphone arrays. *IEEE Access*, 2020, wolumen 8, s. 190083–190093.
- [11] Toposens. *ECHO ONE*,
<https://toposens.com/echo-one>
- [12] ZeroKey. *Quantum RTLS 2.0*,
<https://zerokey.com/quantum-rtls>

Robustness of ultrasonic distance and echo arrival direction sensor to time measurement errors

The paper addresses the problem of determining the direction of arrival of an ultrasonic signal in the context of methods based on measuring the phase shift of the signal. This problem is crucial for precise localization of objects in the robot's environment using ultrasonic sensors. Its solution requires determining the correct geometry of the receiver system arrangement. The best results can only be obtained from using MEMS microphones. The paper presents the effect of the aforementioned geometry on the robustness of the receiving system to errors in the measurement of the detection time of a specific signal phase. Conditions for the optimal configuration of receiver arrangement were also established. Based on these conditions, a receiver arrangement that gives satisfactory resilience has been proposed. This is provided it over a wide range of signal arrival direction angles. In addition, the current status of the implementation of the ultrasonic sensor module design of distance and signal arrival direction is presented.

Recent Advances in Image Captioning

Gurjinder Singh¹

Abstract

Image captioning is a challenging multimodal task that converts visual information into descriptive text by combining computer vision and natural language processing techniques. This review provides a systematic analysis of the evolution of image captioning methods—from early encoder–decoder and attention-based models to state-of-the-art transformer architectures, reinforcement learning, generative adversarial networks, and approaches leveraging large language models. A particular emphasis is placed on the application of image captioning in robotics. Drawing from over seven recent studies, we explore how captioning enhances robotic perception, scene understanding, and human–robot interaction across tasks such as navigation, surveillance, action explanation, and behavior description. The paper also reviews widely adopted datasets, evaluation metrics, and discusses critical challenges including semantic misalignment, hallucinations, and ethical concerns. Emerging solutions like N-shot learning, cross-modal fusion, and lightweight architectures are examined for their potential to enable scalable, adaptable, and context-aware captioning systems. This review offers key insights and directions for future developments in this interdisciplinary domain.

1. Introduction

Image captioning is a multimodal task that generates textual descriptions of an image’s visual content [1]. It integrates computer vision and natural language processing (NLP) to interpret visual elements and translate them into coherent descriptions [2]. Over the years, advancements in deep learning and transformer-based architectures have significantly improved image captioning models, enhancing their accuracy and contextual understanding. This technology has found applications in various domains, such as aiding visually impaired individuals, enhancing content accessibility, improving image search engines, and enabling context-aware assistance in autonomous systems. Furthermore, image captioning plays a critical role in advancing human-robot interaction by enabling robots to better understand and describe their environment, which is crucial for collaborative tasks, real-time decision-making, and effective communication between humans and machines. This paper systematically reviews state-of-the-art methodologies, including classical encoder-decoder frameworks, attention-based models, reinforcement learning techniques, and the recent integration of large language models. The evolution of image captioning not only reflects advancements in model architectures but also highlights the growing demand for systems that can interpret complex scenes in real-time. This is particularly evident in robotics, where the ability to generate detailed and context-aware descriptions can enhance a robot’s situational awareness and facilitate more natural interactions with humans.

¹Wroclaw University of Science and Technology, Faculty of Microsystems, Electronics, and Photonics, Department of Cybernetics and Robotics, Email: gurjinder.singh@pwr.edu.pl

This work was conducted under the guidance of Prof. Elzbieta Roszkowska and Dr. Witold Paluszynski.

The paper is structured as follows: Section 2 describes the method for selecting papers. Section 3 discusses various image captioning techniques, including encoder-decoder frameworks to advance models. Section 4 presents a case study on image captioning in robotics. Section 5 details commonly used evaluation metrics. Section 6 outlines the major datasets, while Section 7 highlights the challenges and future directions. The study concluded in Section 8.

2. Method for Selecting Papers

This review employed a hybrid literature selection strategy that combined keyword-based searches and snowballing techniques. Initial searches were conducted across IEEE Xplore, arXiv, ACM Digital Library, and ACL Anthology using terms such as 'image captioning', 'transformers', 'vision-language models', 'LLMs', and 'robotics'. Further keywords were identified through backward and forward citation tracing, ensuring the inclusion of both foundational studies and the latest advances.

Although papers from 2002 to 2025 were considered, the review places primary emphasis on works published after 2015, reflecting the rapid development of transformer-based architectures, multimodal learning, and their applications in real-world systems. Earlier papers (pre-2015) were included only when they represented seminal contributions, particularly in the development of evaluation metrics.

The goal of this selection process was to answer the following guiding questions:

- **Q1:** What are the current technical trends in image captioning research and how are they evolving?
- **Q2:** In what ways are image captioning methods applied in robotics, and what challenges and outcomes are associated with their use?
- **Q3:** Which approaches offer the most promise for future real-world deployment, especially in dynamic, sensor-driven environments?

This approach ensured that the review is comprehensive in representing both theoretical foundations and state-of-the-art techniques across subfields such as deep learning, reinforcement learning, and robotics. The final selection reflects a diverse and representative sample of the field, capturing both academic quality and real-world applicability.

3. Methods and techniques of image captioning

The field of image captioning has progressed over time. Starting from rule-based methods to more sophisticated deep learning models that combine computer vision and natural language processing techniques seamlessly. In essence, the main goal of image captioning is to teach machines how to understand and explain the visual world using language that mimics communication. A task that involves not identifying objects in a picture but also understanding the broader context actions occurring and even subtle relationships between different elements within the image. Traditionally, these techniques were template-based or used retrieval algorithms to retrieve captions from a specified list. While these techniques worked in limited circumstances, they struggled to generalise across other photos or provide innovative captions.

3.1. Encoder-Decoder

Early models for image captioning relied on the encoder-decoder architecture [1], employing convolutional neural networks (CNNs) for extracting features, while recurrent neural networks (RNNs) or their variations, including LSTMs, produced captions in a sequential manner [3]. However, RNNs struggle to manage long-range dependencies. To address this issue, convolutional decoders were developed, which allow parallel processing but frequently result in overfitting problems [4]. The attention mechanisms [2] were subsequently incorporated to improve the proposed architecture. Two forms of attention were examined: soft attention and hard attention. Soft attention is a fixed method that takes into account all areas of an image, giving greater importance to the parts rated manner using standard backpropagation. In contrast, hard attention is a probabilistic method in which the model selects one image region at a time, considering the attention mechanism as a latent variable. Although it offers results that are easier to understand. Expanding on this, numerous attention-focused improvements, including bottom-up attention [5] and Attention on Attention (AoA) [6], further sharpened the emphasis on important image areas and enhanced caption generation via improved interpretability and more effective context-aware processing. These developments have provided a basis for more advanced image captioning methods.

3.2. Transformer model

The emergence of Transformer models, introduced in Attention Is All You Need [7], represented a major change in sequence modelling by removing recurrence and utilising self-attention mechanisms instead. Transformers facilitate parallel processing and enhanced learning efficiency, with their encoder-decoder structure becoming the norm for numerous sequence-to-sequence applications, such as image captioning. In line with this evolution, recent study proposed a CNN-based encoder-decoder framework [8] combines multiple pre-trained CNNs (e.g., EfficientNet, ResNet) with a transformer decoder for caption generation. This study also explores frequency regularisation to reduce model size while retaining performance, offering a practical approach to lightweight captioning systems in resource-constrained environments. Numerous other Transformer-based models have since advanced image captioning, including the Direction Relation Transformer (DRT) [9], which improves spatial relationship modelling, and the Meshed-Memory Transformer (M2 Transformer) [10], which incorporates memory-based encoding to understand multi-level object relationships. The integration of Vision-Language Models (VLMs) has enhanced image captioning by simultaneously modelling visual and textual inputs.

3.3. Vision Transformers model

Vision Transformers (ViT) [11], which treat image patches as a sequence of tokens similar to words in a sentence, enable global attention throughout the entire image, thereby capturing long-range relationships more efficiently than traditional conventional CNNs. VLMs such as CLIP [12] and BLIP [13] have transformed image captioning through the alignment of image and text representations, significantly improving the diversity and accuracy of generated captions. Furthermore, BLIP-2 [14], a system that combines vision and language, employs object detection and grounding

techniques to generate captions that are more structured and contextually enhanced. These approaches integrate the advantages of both vision and language models to enhance the effectiveness, semantic coherence, and robustness of image captioning across various datasets. With these advancements, Transformers and VLMs have expanded the horizons of image captioning studies by improving the connection between visual features and textual information.

3.4. Reinforcement Learning

Reinforcement Learning (RL) is a learning framework in which an agent learns to make decisions through interaction with an environment to maximise cumulative rewards. In image captioning, RL allows models to generate captions by obtaining feedback based on how well the caption aligns with human-like descriptions or evaluation metrics. Reinforcement Learning (RL) has become a vital component in improving image captioning systems, particularly by addressing the limitations of traditional training approaches such as Maximum Likelihood Estimation (MLE). MLE often faces exposure bias, representing the gap between training (utilising teacher-forced input) and inference (via autoregressive output), and it fails to directly optimise for evaluation metrics related to the task, including BLEU, CIDEr, or METEOR. RL addresses these problems by enhancing the captioning model through reward signals that represent these metrics. A prominent RL-based approach is the Generator-Discriminator Framework [15], which utilises a CNN-LSTM generator featuring recursive attention along with an LSTM-based discriminator. The discriminator evaluates the generated captions using metrics such as CIDEr [16], creating an adversarial cycle that encourages the generator to generate better-quality captions, especially on datasets such as MS COCO [17]. Hybrid RL Methods [18], like Deep Q-Networks (DQN) and Self-Critical Sequence Training (SCST), improve captions by synchronising training objectives with metrics assessed by humans. SCST, for example, utilises the baseline output of the model as a standard to assess reward changes, thereby promoting outputs that exhibit greater semantic depth. Similarly, the Actor-Critic Framework [19] comprises two components: the actor, which produces captions, and the critic, which evaluates captions and provides reward feedback to guide the training process. RL contributes uniquely to the image captioning landscape. RL enables direct optimisation of task-specific rewards and reduces training-inference mismatches.

3.5. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are part of a class of generative models comprising two competing neural networks: a generator focused on producing realistic data (such as image descriptions) and a discriminator intended to distinguish between real and generated data. This competitive setup compels the generator to improve its outputs until it becomes identical to that of real instances. In image captioning, GANs provide an effective method to enhance the naturalness, variety, and originality of the produced captions. For example, Adversarial Training for Diversity [20] employs autoencoders and Conditional GANs (CGANs) [21], along with Wasserstein loss, to improve the diversity and quality of captions. Hybrid GAN-RL methods [22] combine GANs with reinforcement learning techniques such as Policy Gradients and Monte Carlo rollouts to provide intermediate reward signals and enhance caption fluency and conformity with metrics like BLEU [23] and METEOR [24]. Newer

models, such as Semi-Supervised GANs (SVGAN) [25] use both paired and unpaired data, enhancing generalization in low-resource environments. Furthermore, CLIP-guided GANs [12] along with Domain-Specific GANs like CgT-GAN [26], combine CLIP-based semantic rewards with GPT-2 [27] generators and RoBERTa [28] discriminators, thus successfully closing the modality gap between visual and linguistic domains. In the healthcare sector, CapGAN [29] utilises CGANs to produce diagnostic reports, enhancing the precision of captioning models in medical use cases. GANs encourage more human-like, diverse, and domain-adaptive caption generation. Together, RL and GAN push image captioning models closer to human-level understanding and generation across both general and specialised domains.

3.6. N-shot learning

N-shot learning methods—such as zero-shot, one-shot, and few-shot learning—are crucial for enhancing vision-language tasks, thereby allowing models to generalise effectively from minimal supervision. These methods are particularly useful when gathering extensive labelled datasets are expensive, slow, or unfeasible—such as in niche areas (e.g., medical imaging, egocentric perspectives, or customised content). Zero-shot learning allows machine learning models to recognise categories they have not seen previously, removing the necessity for labelled training data. This is often achieved by employing pretrained vision-language models—such as CLIP that align visual data with the corresponding textual descriptions. In contrast, one-shot and few-shot learning focus on gaining knowledge from a single one example or a small set of cases. These techniques typically rely on approaches like meta-learning or transfer learning to improve understanding from limited data. By successfully integrating N-shot learning into image captioning and related tasks, we can achieve several notable benefits: rapid adaptation to new domains or rare objects without complete retraining; lower dependence on large labelled datasets, improving scalability and cutting costs; and personalisation through context-aware captioning, such as generating captions that reflect a person’s personality or preferences with minimal data. For this purpose, TAMF-FSL [30] utilises visual and textual encoders (such as CLIP) along with a meta-fusion module to enhance few-shot classification via better multimodal alignment. In the zero-shot setting, GPT4Ego [31] presents Ego-oriented Text Prompting (EgoTP) and Ego-oriented Visual Parsing (EgoVP) for identifying egocentric activities in videos without requiring further training data. Addressing the challenge of cross-domain generalisation, MFIE [32] improves data diversity by employing multimodal models like Emu and InstructPix2Pix for directed image modification. For customised captioning, Few-Shot Personality-Specific Image Captioning via Meta-Learning [33] employs Model-Agnostic Meta-Learning (MAML) to rapidly adapt to personality-driven caption generation using only a small set of examples, outperforming traditional techniques. Overall, N-shot learning techniques provide a solid foundation for creating more adaptable, scalable, and user-focused image captioning solutions.

3.7. Image Captioning with Large Language Models

Recent progress in Large Language Models has transformed image captioning by facilitating more coherent, diverse, and context-aware text generation. LLMs like GPT-2, GPT-3.5, and GPT-4, are trained on extensive text datasets to comprehend and

produce language akin to human-like language. Their integration into image captioning bridges the semantic gap between visual inputs and natural language outputs adding powerful reasoning, contextualisation, and fluency. By combining LLMs with pretrained vision models, modern captioning systems can move beyond object description tasks to produce richer, more informative captions. For instance, the Enhanced CLIP-GPT Framework [34] uses CLIP [12] for feature extraction and GPT-2 [27] for language generation via a Transformer-based mapping network with adaptive attention, thereby reducing data dependency and training time. The CPLIC framework [35] fuses BLIP-2 [14] generated captions with GPT-3.5 [36], thereby enabling knowledge-based visual question answering without external knowledge bases. In specialized domains like thermal imaging, the LLM-Based Semantic Fusion Framework [37] combines captions from infrared and visible images using a dual-modal fusion network (YOLOv7 + GPT-3.5), improving results under poor lighting conditions. Moreover, recent research has investigated self-supervised and semi-supervised learning methods [38], which allow models to acquire significant visual and linguistic representations from unlabelled data by utilising intrinsic data structures (e.g., predicting missing image regions or caption fragments). These techniques reduce reliance on large annotated datasets while maintaining competitive performance. Collectively, these frameworks demonstrate the growing importance of LLMs in improving caption quality, contextual richness, and data efficiency. Despite their success, challenges such as modality alignment, language hallucination, and fine-grained grounding remain open problems, that guide ongoing research towards more robust and interpretable vision-language integration.

4. Image Captioning in Robotics

Image captioning is a game-changer for robotics, as it empowers machines with the ability to interpret and verbally describe their surroundings in real-time. This capability is especially beneficial in unstructured or dynamic environments where traditional rule-based systems and fixed sensor responses may fall short. By combining visual perception with natural language generation, image captioning allows robots to interact more effectively with humans and make more context-aware decisions.

A standout application is the Human-Aware Context Generator (HACG) system, which merges image captioning with facial recognition and emotion detection to create rich, context-sensitive scene descriptions [39]. This is particularly valuable in security and surveillance scenarios where autonomous agents must report on human behaviour and emotional states. For example, a security robot using HACG can identify a group of people, detect their emotions (e.g., agitation or distress), and generate a caption like "A group of anxious individuals gathered near the emergency exit." Such real-time descriptions are critical for timely human intervention. Another major advancement involves integrating Large Language Models (LLMs) with dense captioning systems. These setups go beyond generating simple labels to produce elaborate textual descriptions of complex scenes. Studies, such as [40], have shown that combining LLMs with vision transformers enables robots to perform more nuanced reasoning and contextual interpretation. This improves not only scene understanding but also communication. For instance, social robots like Pepper employ these techniques to describe objects and ongoing activities, facilitating more engaging and informative interactions in customer service roles.

In autonomous navigation, image captioning plays a pivotal role by offering real-time descriptions of terrain, obstacles, and nearby objects. An approach called the LLM-based Context Translator, described in [41], enables robots to interpret environmental cues—such as distinguishing between “a wet, slippery surface” and “a clear dry path” and adapt their route or speed accordingly. This ability is crucial for operations in unpredictable settings like search-and-rescue missions, where robots may face debris, uneven terrain, or moving obstacles.

To address environmental perception under poor lighting conditions, a novel RGB-IR image captioning model was introduced, which fuses visible (RGB) and infrared (IR) images using dual ResNet-101 encoders and an LSTM-based decoder [42]. The proposed is deployed within a wearable helmet system that provides real-time captioning and audio feedback via a headset—offering enhanced navigational support for visually impaired users. In another direction [43], unsupervised learning was used to generate natural language captions of robot behaviour from actuator and camera data using K-means clustering and byte-pair encoding, along with an encoder-decoder architecture and attention mechanism. Their hybrid model effectively handled data scarcity and generated semantically rich behaviour explanations. A new task—operative action captioning—was introduced to estimate and describe actions required to transition between a current and target state. The approach proposed in [44], leverages a dual dynamic attention (DUDA) model enhanced with an auxiliary scene graph prediction module to generate high-quality captions that capture fine-grained visual changes across states.

Additional innovations enhance the spatiotemporal depth of captioning. A scene graph-guided captioning framework was proposed for understanding the robotic indoor scene from streaming video [45]. This method incrementally updates a scene graph as the robot moves, generating temporally coherent, context-rich captions. The proposed model combines YOLOv3 for object detection, an attention-based LSTM captioning network, and a spatial scene graph update mechanism. Evaluations on real robotic platforms confirmed the system’s effectiveness in real-time video captioning. Moreover, efforts to integrate egocentric and exocentric viewpoints have led to systems capable of capturing both global contexts and robot-centred activities. A captioning framework based on LSTM and Transformer architectures trained on the Global, Action, Interaction (GAI) dataset effectively describes scene-level context, robot actions, and human-robot interactions in tandem [46]. Finally, to improve captioning in constrained environments such as homes or factories, a classification-augmented training method combines object recognition with template-based caption generation [47]. This improves the caption accuracy by incorporating known object categories while maintaining the natural linguistic structure.

These examples highlight how image captioning is transforming robotics enabling emotion-aware surveillance, context-rich social interaction, and intelligent navigation in complex environments. From RGB-IR fusion for low-light perception to unsupervised and action-based captioning for behaviour explanation, these innovations enhance robot understanding and communication. Scene graph-guided and classification-augmented methods further improve caption detail and reliability across dynamic and structured settings. Future research should focus on addressing key limitations such as sensor noise, variable lighting environments, and real-time processing constraints

to further improve caption accuracy and system responsiveness. In addition, ensuring ethical deployment, preserving data privacy, and enabling generalisation across diverse operational domains are critical. As image captioning technology continues to evolve, it promises to deliver robots that are not only more autonomous and perceptive but also capable of seamless, human-friendly communication and interaction in real-world environments.

5. Evaluation Metrics for Image Captioning Methods

Evaluation metrics play a crucial role in quantifying the performance of image captioning systems by comparing the generated captions to the human-annotated reference captions. Although no single metric perfectly captures caption quality, a combination of n-gram overlap, semantic alignment, and human judgement offers a comprehensive evaluation. BLEU (Bilingual Evaluation Understudy) [23] measures n-gram precision with a brevity penalty to avoid short outputs but relies on exact word matches, missing synonyms and paraphrases. METEOR [24] improves on this by considering stemming and synonyms and penalising word order errors, although it remains surface-level and computationally heavy. ROUGE [48], focused on recall, captures content coverage but lacks precision and semantic understanding. CIDEr [16] enhances evaluation by comparing captions against multiple references to reflect consensus better, but it can be sensitive to the diversity of those references. SPICE [49] complements this by analysing semantic meaning via scene graphs, although it is also computationally demanding. To balance these limitations, SPIDer [50] combines CIDEr’s lexical consensus with SPICE’s semantic depth, offering a more holistic measure despite inheriting some limitations. Human evaluation remains the gold standard for assessing fluency and creativity [51], although it is costly and subjective. Overall, no single metric suffices—BLEU and METEOR focus on lexical accuracy, ROUGE on recall, CIDEr and SPICE on consensus and semantics, and SPIDer merges both. Human judgement provides deeper insights, emphasising the need for integrated evaluation approaches.

6. Datasets

Several large-scale datasets are crucial for image captioning, classification, and multimodal learning. Table 1 summarises the main datasets discussed in this paper.

Tab. 1. Main Datasets for Image Captioning

Dataset	Size/Scale	Description
MSCOCO (2017) [17]	~120K images	Widely used benchmark for object detection, segmentation, and captioning.
Flickr8k / Flickr30k[52]	8K / 30K images	Small-scale image–text datasets for training and evaluating captioning models.
ImageNet [53]	1.3M images	Extensive image database for feature extraction and pre-training in vision tasks.
ImageNet-21k [54]	14M images	Extended version of ImageNet, covering 21,000 categories.
LAION-5B [55]	5B image-text	Massive multimodal dataset for vision–language pre-training.

In addition to the datasets listed in Table 1, several other datasets are discussed in this paper. JFT-300M, an internal Google dataset, comprises 300M images with

over 30K noisy labels, supporting large-scale vision tasks. For vision-language tasks, Google Conceptual Captions (CC) [56] provides 3.3M images with alt-text captions, enabling robust training for captioning models. VQA v2.0, which is a widely used dataset for Visual Question Answering, features 1.1M QA pairs on more than 200K COCO images, and offers open-ended answers. Finally, VizWiz [57], which is tailored for visually impaired users, includes 36K images with 200K QA pairs, thereby enhancing accessibility-focused AI.

7. Challenges and Future Potentials

7.1. Challenges

- **Semantic Gap:** A major challenge lies in the gap between basic visual elements and sophisticated textual meanings. Models such as CLIP align broad image-text pairs but frequently overlook subtle relationships (e.g., “holding” vs. “near”), resulting in problems such as object hallucination. Closing this gap requires more sophisticated alignment methods [58, 59].
- **Data Scarcity:** There is a shortage of high-quality, domain-specific annotated datasets, particularly in specialised applications like medical images and specific language captioning [60].
- **Computational Demands:** Training large-scale, Transformer-based models requires significant computational resources, which limits their deployment in resource constrained environments [58, 61].
- **Hallucination in Generative Models:** Large Vision-Language Models (LVLMs) are prone to hallucination [58, 59, 61, 62], generating information that does not exist in the visual input. Studies have shown that even state-of-the-art models contain a significant percentage of hallucinatory text, including non-existent objects and inaccurate relationships. This issue reduces trust in critical applications, such as medical diagnostics, in where accuracy is paramount.
- **Ethical and Privacy Concerns:** The deployment of VLMs in sensitive areas such as surveillance and healthcare raises ethical and privacy concerns. For example, facial recognition and medical captioning models may expose sensitive patient data, highlighting the need for robust privacy-preserving techniques [29, 60].

7.2. Future Directions

- Leverage advances in vision–language pre-training and multimodal fusion (e.g., CLIP [12], BLIP [13]) to enhance modality alignment [34, 35].
- Developing models tailored to specific fields, such as medical imaging or robotics, can enhance performance and applicability in niche areas.
- Architectures that combine tasks such as captioning, visual question answering, and segmentation can share visual encoders [63], reducing redundancy and streamlining deployment in robotics applications.
- Design interactive, user-controllable captioning systems for customisable outputs in high-stake applications.
- Explore multilingual captioning and lightweight architectures using techniques like advanced data augmentation, synthetic data generation, and transfer learning [59].
- Robust cross-modal fusion that integrates additional data modalities—such as Li-

DAR and thermal imaging—with vision-language models (VLMs) enhances scene understanding, especially in complex or challenging environments.

Collectively, addressing these challenges through emerging techniques will pave the way for more robust, interpretable, and versatile image captioning systems with significant implications across various domains.

8. Conclusion

This review systematically surveys recent advances in image captioning, from basic rule-driven techniques to complex multimodal systems that integrate computer vision with natural language processing. Improvements in deep learning—spanning encoder–decoder models, attention techniques, cutting-edge transformers, and reinforcement learning systems—have greatly improved the coherence and contextual depth of produced captions. The combination of large language models and vision-language models has increasingly connected visual inputs with natural language outputs, facilitating more nuanced and context-sensitive descriptions. Moreover, new approaches such as N-shot learning and GAN-driven methods have expanded the possibilities by providing scalability and enhanced performance in areas where annotated data are scarce.

Robotics applications and various real-time systems demonstrate the practical effects of these advancements, offering improved situational awareness and interactive features. Notwithstanding these significant advancements, challenges remain, such as tackling the semantic gap, overcoming data scarcity, handling high computational requirements, and alleviating problems like language hallucination and ethical issues. Upcoming studies should emphasise strong cross-modal integration, innovations tailored to specific domains, and ethical, personalised captioning approaches, setting the stage for more adaptable, dependable, and understandable image captioning systems.

References

- [1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3156–3164.
- [2] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” 2016, *arXiv*. [Online]. Available: <https://arxiv.org/abs/1502.03044>
- [3] S. Ugale, D. Jain, D. Chikmurge, and S. S. Barve, “Enhancing image captioning through the integration of image processing and nlp,” in *2024 8th Int. Conf. on Computing, Communication, Control and Automation (ICCUBEA)*.
- [4] P. Dhamanskar, W. Pereira, S. Khot, and M. Bhole, “Multimodal learning for image caption generation: A deep learning approach,” in *2024 8th Int. Conf. on Computing, Communication, Control and Automation (ICCUBEA)*.
- [5] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” *arXiv*, 2018. [Online]. Available: <https://arxiv.org/abs/1707.07998>
- [6] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, “Attention on attention for image captioning,” in *2019 IEEE/CVF Int. Conf. on Computer Vision*.

- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 1706.03762, 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [8] M. A. R. Ridoy, M. M. Hasan, and S. Bhowmick, "Compressed image captioning using cnn-based encoder-decoder framework," 2024. [Online]. Available: <https://arxiv.org/abs/2404.18062>
- [9] Z. Song, X. Zhou, L. Dong, J. Tan, and L. Guo, "Direction relation transformer for image captioning," in *Proceedings of the 29th ACM Int. Conf. on Multimedia*. USA: Association for Computing Machinery, 2021, p. 5056–5064. [Online]. Available: <https://doi.org/10.1145/3474085.3475607>
- [10] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- [11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. on Learning Representations*, 2021.
- [12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *arXiv*, 2103.00020, 2021. [Online]. Available: <https://arxiv.org/abs/2103.00020>
- [13] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," *arXiv*, 2201.12086, 2022. [Online]. Available: <https://arxiv.org/abs/2201.12086>
- [14] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proceedings of the 40th Int. Conf. on Machine Learning*, ser. ICML'23. JMLR.org, 2023.
- [15] A. Verma, S. Agarwal, K. Arya, I. Petrlik, R. Esparza, and C. Rodriguez, "Image captioning with reinforcement learning," in *2023 IEEE Int. Conf. on Computer Vision and Machine Intelligence (CVMI)*, 2023, pp. 1–7.
- [16] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," *arXiv*, 1411.5726, 2015. [Online]. Available: <https://arxiv.org/abs/1411.5726>
- [17] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," *arXiv*, 1405.0312, 2015. [Online]. Available: <https://arxiv.org/abs/1405.0312>
- [18] Y. Tang, "Image captioning: From encoder-decoder to reinforcement learning," in *2022 6th Int. Conf. on Imaging, Signal Processing and Communications (ICISPC)*, 2022, pp. 6–10.
- [19] E. Rashno, M. Safarzadehvahed, F. Zulkernine, and S. Givigi, "Image caption generation based on image-text matching schema in deep reinforcement learning," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*.
- [20] S. Amirian, K. Rasheed, T. R. Taha, and H. R. Arabnia, "Image captioning with generative adversarial network," in *2019 Int. Conf. on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 272–275.

- [21] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv*, 1411.1784, 2014. [Online]. Available: <https://arxiv.org/abs/1411.1784>
- [22] S. Yan, F. Wu, J. S. Smith, W. Lu, and B. Zhang, “Image captioning using adversarial networks and reinforcement learning,” in *2018 24th Int. Conf. on Pattern Recognition (ICPR)*, 2018, pp. 248–253.
- [23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL ’02. USA: Association for Computational Linguistics, 2002, p. 311–318. [Online]. Available: <https://doi.org/10.3115/1073083.1073135>
- [24] A. Lavie and A. Agarwal, “Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, ser. StatMT ’07. USA: Association for Computational Linguistics, 2007, p. 228–231.
- [25] Y. Zhang, W. Zeng, G. He, and Y. Liu, “Svgan: Semi-supervised generative adversarial network for image captioning,” in *2020 IEEE Conf. on Telecommunications, Optics and Computer Science (TOCS)*, 2020, pp. 289–292.
- [26] J. Yu, H. Li, Y. Hao, B. Zhu, T. Xu, and X. He, “Cgt-gan: Clip-guided text gan for image captioning,” in *Proceedings of the 31st ACM Int. Conf. on Multimedia*, ser. MM ’23. ACM, Oct. 2023, p. 2252–2263. [Online]. Available: <http://dx.doi.org/10.1145/3581783.3611891>
- [27] R. Pandey, H. Waghela, S. Rakshit, A. Rangari, A. Singh, R. Kumar, R. Ghosal, and J. Sen, “Generative ai-based text generation methods using pre-trained gpt-2 model,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.01786>
- [28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” 2019. [Online]. Available: <https://arxiv.org/abs/1907.11692>
- [29] L. Song, C. He, L. Xu, and B. Zheng, “Capgan: Medical image captioning with clinical style and pathology preservation using conditional generative adversarial nets,” in *2022 5th Int. Conf. on Pattern Recognition and Artificial Intelligence (PRAI)*, 2022, pp. 1023–1032.
- [30] M. Han, Y. Zhan, Y. Luo, H. Hu, K. Su, and B. Du, “Textual enhanced adaptive meta-fusion for few-shot visual recognition,” *IEEE Transactions on Multimedia*, vol. 26, pp. 2408–2418, 2024.
- [31] G. Dai, X. Shu, W. Wu, R. Yan, and J. Zhang, “Gpt4ego: Unleashing the potential of pre-trained models for zero-shot egocentric action recognition,” *IEEE Transactions on Multimedia*, vol. 27, pp. 401–413, 2025.
- [32] Z. Lin, W. Yang, L. Lan, M. Geng, H. Wang, H. Chi, X. Li, and J. Wang, “Diversifying cross-domain few-shot learning via multimodal image editing,” in *ICASSP 2024 - 2024 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*.
- [33] M. Hosseinzadeh and Y. Wang, “Few-shot personality-specific image captioning via meta-learning,” in *2023 20th Conf. on Robots and Vision (CRV)*.
- [34] R. Song, B. Zhao, and L. Yu, “Enhanced clip-gpt framework for cross-lingual remote sensing image captioning,” *IEEE Access*, vol. 13, pp. 904–915, 2025.
- [35] Y. Yang, C. Cao, F. Yuan, S. Zeng, D. Wang, and Y. Liu, “Enhancing gpt-3.5 for knowledge-based vqa with in-context prompt learning and image captioning,” in

- 2024 *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*.
- [36] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, and X. Huang, "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," 2023. [Online]. Available: <https://arxiv.org/abs/2303.10420>
 - [37] K. Wang, S. Lou, J. Wang, and X. Yuan, "Llm based semantic fusion of infrared and visible image caption," in *2024 9th Int. Conf. on Intelligent Informatics and Biomedical Sciences (ICIBMS)*, vol. 9, 2024, pp. 315–319.
 - [38] C. Jin, "Self-supervised image captioning with clip," *arXiv*, 2306.15111, 2023. [Online]. Available: <https://arxiv.org/abs/2306.15111>
 - [39] R. C. Luo, Y.-T. Hsu, and H.-J. Ye, "Multi-modal human-aware image caption system for intelligent service robotics applications," in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*.
 - [40] L. Grassi, Z. Hong, C. T. Recchiuto, and A. Sgorbissa, "Grounding conversational robots on vision through dense captioning and large language models," in *2024 IEEE Int. Conf. on Robotics and Automation (ICRA)*.
 - [41] C. L. Shek, X. Wu, W. A. Suttle, C. Busart, E. Zaroukian, D. Manocha, P. Tokekar, and A. S. Bedi, "Lancar: Leveraging language for context-aware robot locomotion in unstructured environments," 2024. [Online]. Available: <https://arxiv.org/abs/2310.00481>
 - [42] Y. Wang, S. Lou, K. Wang, Y. Wang, X. Yuan, and H. Liu, "Automatic captioning based on visible and infrared images," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11312–11318.
 - [43] K. Yoshino, K. Wakimoto, Y. Nishimura, and S. Nakamura, "Caption generation of robot behaviors based on unsupervised learning of action segments," 2020. [Online]. Available: <https://arxiv.org/abs/2003.10066>
 - [44] T. Nakamura, S. Kawano, A. Yuguchi, Y. Kawanishi, and K. Yoshino, "Operative action captioning for estimating system actions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 6124–6130.
 - [45] S.-H. Kang and J.-H. Han, "Video captioning based on both egocentric and exocentric views of robot vision for human-robot interaction," *International Journal of Social Robotics*, vol. 15, no. 4, pp. 631–641, 2023.
 - [46] X. Li, D. Guo, H. Liu, and F. Sun, "Robotic indoor scene captioning from streaming video," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6109–6115.
 - [47] R. C. Luo, Y.-T. Hsu, Y.-C. Wen, and H.-J. Ye, "Visual image caption generation for service robotics and industrial applications," in *2019 IEEE Int. Conf. on Industrial Cyber Physical Systems (ICPS)*.
 - [48] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 74–81, 2004, pp. 74–81.
 - [49] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," *arXiv*, 1607.08822, 2016. [Online]. Available: <https://arxiv.org/abs/1607.08822>
 - [50] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, p. 208–221, Jan. 2017. [Online].

Available: <http://dx.doi.org/10.1109/TASLP.2016.2632307>

- [51] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, “From captions to visual concepts and back,” 2015. [Online]. Available: <https://arxiv.org/abs/1411.4952>
- [52] E. van Miltenburg, “Stereotyping and bias in the flickr30k dataset,” 2016. [Online]. Available: <https://arxiv.org/abs/1605.06083>
- [53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*.
- [54] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Z-Manor, “Imagenet-21k pretraining for the masses,” 2021. [Online]. Available: <https://arxiv.org/abs/2104.10972>
- [55] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, “Laion-5b: An open large-scale dataset for training next generation image-text models,” *arXiv*, 2210.08402, 2022. [Online]. Available: <https://arxiv.org/abs/2210.08402>
- [56] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Jul. 2018. [Online]. Available: <https://aclanthology.org/P18-1238/>
- [57] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, “Vizwiz grand challenge: Answering visual questions from blind people,” 2018. [Online]. Available: <https://arxiv.org/abs/1802.08218>
- [58] A. Jamil, Saif-Ur-Rehman, K. Mahmood, M. G. Villar, T. Prola, I. D. L. T. Diez, M. A. Samad, and I. Ashraf, “Deep learning approaches for image captioning: Opportunities, challenges and future potential,” *IEEE Access*, pp. 1–1, 2024.
- [59] T. Ghandi, H. Pourreza, and H. Mahyar, “Deep learning approaches on image captioning: A review,” *ACM Comput. Surv.*, vol. 56, no. 3, Oct. 2023. [Online]. Available: <https://doi.org/10.1145/3617592>
- [60] S. Sukhi, A. Q. Ohi, M. S. Rahman, and M. Mridha, “A survey on bengali image captioning: Architectures, challenges, and directions,” in *2021 Int. Conf. on Science & Contemporary Technologies (ICSCT)*, 2021, pp. 1–5.
- [61] S. Jain, S. Zawar, Y. Rupchandani, and M. Chimanna, “Image description generation using deep learning: A comprehensive overview,” in *2024 IEEE Int. Conf. on Information Technology, Electronics and Intelligent Communication Systems*.
- [62] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, “Semantic compositional networks for visual captioning,” in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5630–5639.
- [63] S. Song, X. Li, S. Li, S. Zhao, J. Yu, J. Ma, X. Mao, W. Zhang, and M. Wang, “How to bridge the gap between modalities: Survey on multimodal large language model,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–20, 2025.

Zastosowanie kamery hiperspektralnej w nawigacji robotów mobilnych

Barbara Siemiątkowska¹, Julian Żebrecki¹, Mateusz Małkowski¹

Streszczenie

W poniższej pracy zostanie przedstawiony przegląd prowadzonych na Wydziale Mechatroniki badań związanych z wykorzystaniem kamery hiperspektralnej w nawigacji robotów mobilnych. Omówione zostaną następujące tematy: wprowadzenie do analizy spektralnej, algorytm rozpoznawania rodzaju powierzchni po której przemieszcza się robot (asfalt, droga gruntowa, trawa, drzewa), metoda oceny stopnia nawodnienia terenu oraz algorytm rozpoznawanie plam wody i oleju na powierzchni. Wykazano dlaczego wyniki badań są istotne z punktu widzenia nawigacji. Badanie dotyczące rodzaju powierzchni były prowadzone przez Juliana Żebreckiego w ramach pracy inżynierskiej. Algorytmy określania stopienia nawodnienia terenu były opracowane przez Mateusza Małkowskiego.

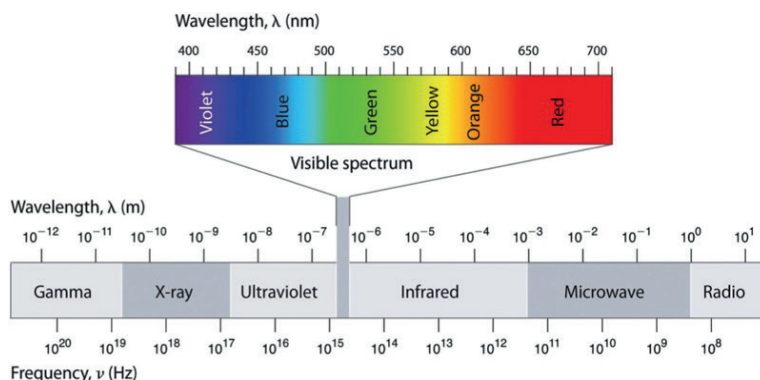
1. Wstęp

Kołowe roboty mobilne przemieszczające się na zewnątrz pomieszczeń muszą dostosowywać algorytmy sterowania do dynamicznie zmieniającego się otoczenia. W terenie naturalnym robot może napotkać różnego typu nawierzchnie: piasek, żwir, błoto, trawę, asfalt, lód itp. Każdy z tych typów nawierzchni charakteryzuje się odmiennymi właściwościami fizycznymi, takimi jak: przyczepność, nośność, wilgotność. Parametry te powinny być uwzględnione w procesie planowania trasy oraz w procesie sterowania [7]. Brak wiedzy o właściwościach podłoża może prowadzić do poślizgu, zakopania się lub nawet przewrócenia się pojazdu. Dlatego współczesne rozwiązania w robotyce mobilnej coraz częściej wykorzystują różnego typu sensory umożliwiające ocenę rodzaju podłoża, takie jak [9]: kamery RGB i hiperspektralne, lidary i radary, czujniki siły i momentu IMU (ang. *Inertial Measurement*). Znając rodzaj nawierzchni możliwa jest ocena przejezdności terenu, dynamiczne dostosowanie parametrów ruchu oraz planowanie bezpiecznej ścieżki przejazdu.

Analiza podłoża jest nieodzownym elementem systemów nawigacyjnych robota, mającym bezpośredni wpływ na bezpieczeństwo i efektywność realizacji zadań.

W poniższej pracy przedstawiony zostanie przegląd badań prowadzonych na wydziale Mechatroniki dotyczących zastosowania kamery hiperspektralnej do analizy powierzchni, część eksperymentów była wykonywana w ramach prac inżynierskich. Przedstawione zostanie porównanie algorytmów klasyfikacji powierzchni, analiza nawodnienia terenu oraz wykrywanie zanieczyszczeń - plam oleju i wody.

¹ Politechnika Warszawska, Wydział Mechatroniki, barbara.siemiatkowska@pw.edu.pl



Rys. 1. Pasma spektralne, źródło [2]

2. Kamera hiperspektralna

Klasyczne kamery rejestrują obrazy w paśmie widzialnym zwykle w trzech kanałach (czerwonym, zielonym i niebieskim). Kamera hiperspektralna rejestruje setki wąskich pasm widma elektromagnetycznego, najczęściej w zakresie od światła widzialnego (VIS) do bliskiej podczerwieni (NIR) [8].

Na rys. 1 przedstawiono pasma hiperspektralne i wyróżniono zakres światła widzialnego.

Kamery hiperspektralne różnią się zakresem długości fal, liczbą pasm oraz sposobem akwizycji danych. Metody skanowania możemy podzielić na:

- Skanowanie obszarowe – pojedynczy detektor skanuje piksele punkt po punkcie.
- Skanowanie liniowe – kamera rejestruje jeden wiersz obrazu naraz, skanując obiekt linia po linii.
- Rejestracja snapshot – kamera rejestruje cały obraz hiperspektralny jednocześnie, co pozwala na analizę dynamicznych scen.

Sposób skanowania wpływa na parametry takie jak czas zbierania informacji, koszt urządzenia oraz pośrednio na zakres możliwych zastosowań. Tabeli tab. 1 przedstawiono podstawowe cechy kamer.

Tab. 1. Cechy kamer

Typ skanowania	Czas zbierania danych	Cena	Zastosowania
liniowe	Szybki, ale wymaga ruchu	średnia	mapowanie terenu
obrazowe	natychmiastowy	wysoka	sceny dynamiczne
punktowe	długi	niska / średnia	laboratoria

W analizie spektralnej istotnym parametrem jest gęstość optyczna (OD, ang. *optical density*). Jest to wielkość opisująca, w jakim stopniu materiał pochłania promieniowanie elektromagnetyczne i określana jest wzorem (1).

$$OD = \frac{I_0}{I}, \quad (1)$$

gdzie:

I_0 - natężenia światła padającego,

I - do natężenia światła odbitego lub transmitowanego przez dany materiał.

Wysoka gęstość optyczna oznacza silne pochłanianie światła, natomiast niska dużą przepuszczalność materiału.

Różne materiały pochłaniają promieniowanie w specyficznych zakresach długości fal [6, 1], co umożliwia nie tylko klasyfikację rodzaju powierzchni, ale nawet określenie stężenia substancji. Typowe opisywane w literaturze zastosowania analizy spektralnej to [6]:

- ocena składu mineralnego gleb i skał,
- analiza zdrowia roślin poprzez pomiar absorpcji chlorofilu,
- wykrywanie zanieczyszczeń środowiska,
- diagnostyka medyczna – np. w ocenie właściwości tkanek.

Z punktu widzenia zastosowania w robotyce mobilnej kluczowe są następujące zadania: rozpoznawanie typu podłoża, określenie stopnia nawodnienia terenu (błoto, powierzchnia twarda), wykrywanie różnego typu wycieków np. woda, czy olej.

3. Tworzenie mapy semantycznej podłoża

W tym rozdziale przedstawiony zostanie system klasyfikacji podłoża i tworzenie na tej podstawie mapy podłoża. Badania były prowadzone przez Juliana Żebreckiego. Algorytm składa się z następujących etapów:

- akwizycja danych
- normalizacja
- selekcja cech
- klasyfikacja
- aktualizacja mapy

Zakładamy, że będziemy rozróżniać cztery klasy obiektów: asfalt, droga gruntowa, trawa i liście zielone, drzewa.

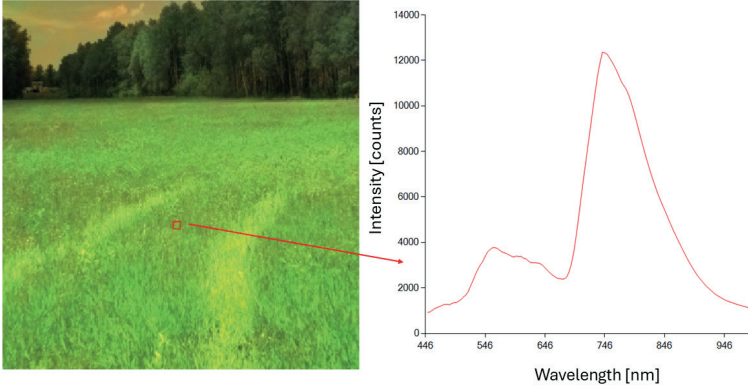
3.1. Akwizycja obrazu

W celu pozyskania obrazów hiperspektralnych wykorzystano kamerę Cubert Q285, która umożliwia rejestrację danych w zakresie widzialnym oraz bliskiej podczerwieni, obejmującym długości fal od 450 nm do 1000 nm. Kamera pozwala na akwizycję do 20 hiperspektralnych kostek danych (ang. hyperspectral cubes) na sekundę.

Dla każdego piksela obrazu otrzymujemy wektor: $[I_1, \dots, I_n]$, gdzie I_i jest intensywnością i -tej długości fali, $n=138$ (mamy 138 pasm hiperspektralnych).

Na rys. 2 przedstawiono przykładową odpowiedź hiperspektralną dla terenu pokrytego trawą. Wyraźnie widać, że gęstość optyczna jest różna dla różnych długości fal.

Wykorzystując kamerę hiperspektralną zebrano dane różnego typu powierzchni w różnych warunkach oświetleniowych. Dane podzielono na dane treningowe, testowe i walidacyjne. Jeśli w zbiorze uczącym klasy są reprezentowane przez różną liczbę należących do nich elementów ma to zwykle negatywny wpływ na wynik klasyfikacji, dlatego zadbano aby klasy były równoliczne.



Rys. 2. Odpowiedź hiperspektralna

3.2. Normalizacja danych

Aby zredukować wpływ zmian oświetlenia na jakość otrzymanych wyników należy przeprowadzić normalizację.

Jeśli I_i jest natężeniem i -tej długości fali (transmitowanej mierzonym po odbiciu światła przez dany materiał), I_{max} jest natężeniem maksymalnym, a I_{min} jest natężeniem minimalnym, to nowa wartość I_i^n opisywana jest wzorem (2).

$$I_i^n = \frac{I_i - I_{min}}{I_{max} - I_{min}}. \quad (2)$$

3.3. Redukcja danych

Redukcja danych odbywa się przy pomocy algorytmu PCA [4, 3]. Metoda składa się z następujących kroków:

- Dla wektorów należących do zbioru uczącego obliczana jest macierz kowariancji.
- Wyznaczane są wartości i wektory własne macierzy kowariancji. Wektory własne wyznaczają nowe kierunki w przestrzeni danych, a odpowiadające im wartości własne wskazują, jaka jest zmienność (wariancja) wzdłuż tych kierunków.
- Wybierane są składowe o najwyższych wartościach wariancji (EV).

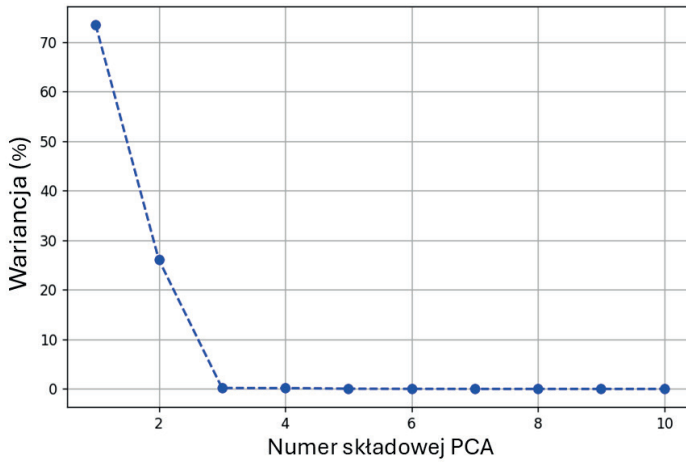
$$EV_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}, \quad (3)$$

gdzie:

n to liczba wszystkich składowych (czyli wymiar oryginalnych danych),

λ_i - i -ta wartość własna

- Dane są rzutowane na przestrzeń utworzoną przez wybrane główne składowe.
Na (3) przedstawiono wartość objaśnionej wariancji dla dziesięciu największych wartości własnych.



Rys. 3. Wyjaśniona wariancja dla głównych składowych

3.4. Klasyfikacja danych

Kolejnym krokiem jest klasyfikacja danych. Testowane były cztery różne algorytmy: SVM, k-NN, drzewa decyzyjne i naiwny klasyfikator Bayesa. Aby sprawdzić skuteczność badanych algorytmów klasyfikacji, zastosowano cztery standardowe metryki oceny. Pierwszą z nich była dokładność klasyfikacji (ang. accuracy). Accuracy określone jest wzorem (4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (4)$$

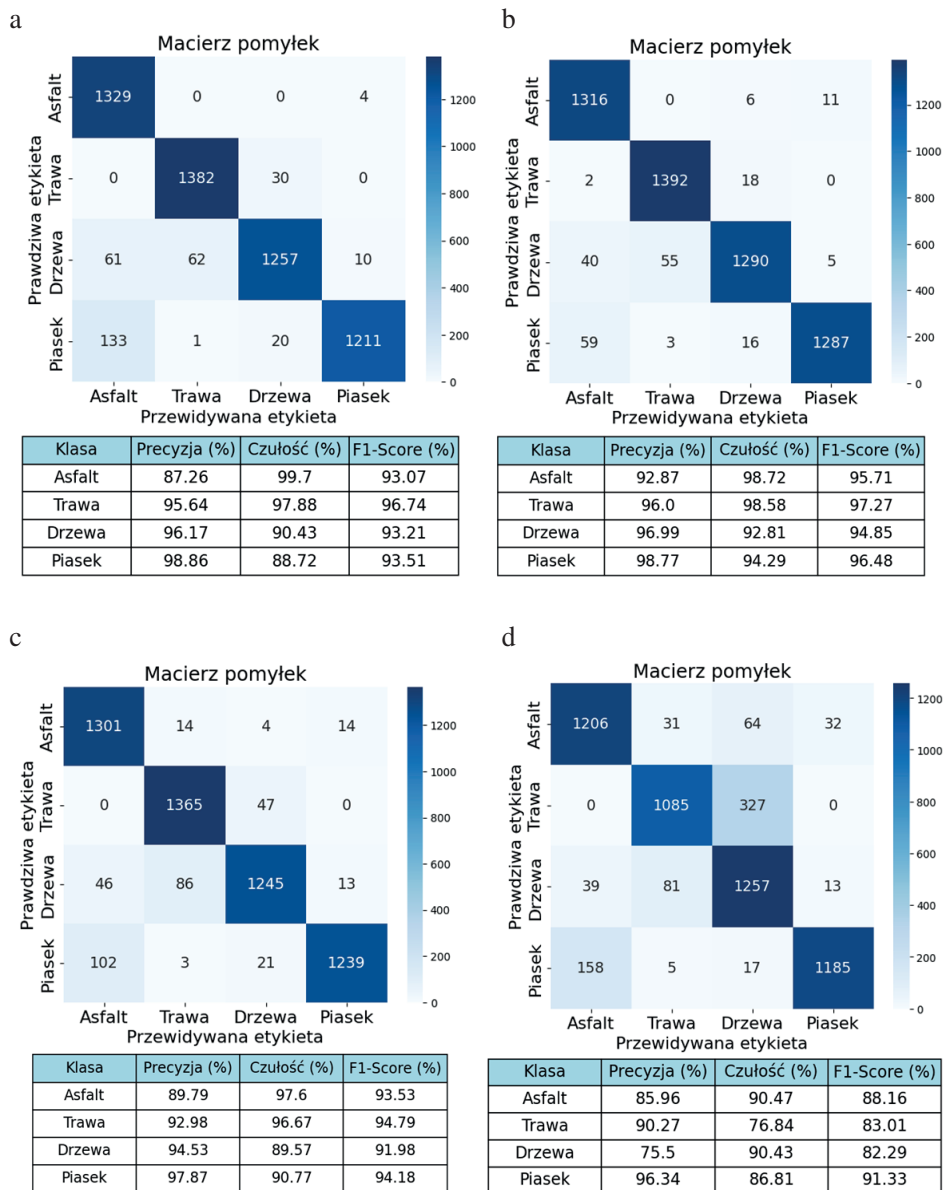
gdzie TP oraz TN to liczba poprawnie wykrytych pozytywnych i negatywnych próbek. A z kolei FP i FN to liczba próbek błędnie oznaczonych jako pozytywne i negatywne. Dodatkowo obliczono metryki: Precyzja (ang. Precision), Czułość (ang. Recall) oraz F1-Score.

$$Precision = \frac{TP}{TP + FP}, \quad (5)$$

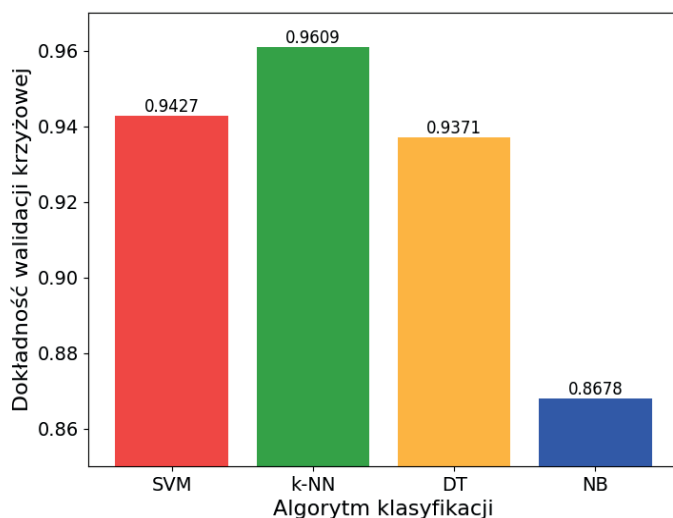
$$Recall = \frac{TP}{TP + FN}, \quad (6)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (7)$$

Na rys. 4 przedstawiono wynik określenia wymienionych powyżej parametrów dla różnych algorytmów klasyfikacji.



Rys. 4. Dokładność klasyfikacji: a – drzewo decyzyjne, b – metoda trzech sąsiadów, c – algorytm SVM, d – Naiwny klasyfikator Bayesa



Rys. 5. Dokładność klasyfikacji

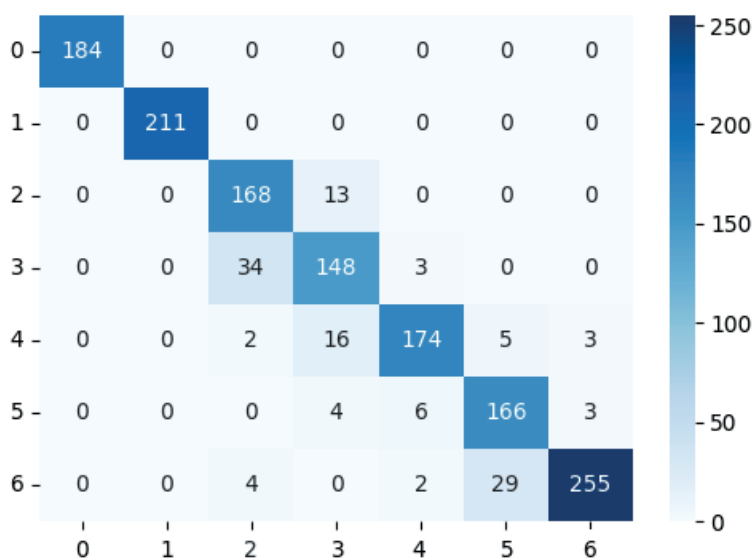
Na rys. 5 przedstawiono wynik walidacji krzyżowej.

Klasyfikacja była prowadzona dla pojedynczych pikseli obrazu. Najlepsze wyniki osiągnięto w przypadku drzewa decyzyjnego, najgorsze dla klasyfikatora naiwnego klasyfikatora Bayesa. Dokładność można poprawić, jeśli klasyfikując pojedyncze piksele uwzględniony jest kontekst. Jeśli piksel zaklasyfikowany jako część drogi gruntowej jest otoczony pikselami zaklasyfikowanymi jako asfalt, to powinniśmy zmienić wynik klasyfikacji na asfalt.

4. Wykrywanie plam oleju i wody

Wykrywanie plam wody i oleju ma istotne znaczenie dla efektywnego działania robota mobilnego, szczególnie gdy robot pracuje w środowisku przemysłowym, w którym tego typu wycieki mogą zdarzać się często. Z jednej strony woda i olej mogą znacząco zmniejszyć przyczepność kół, co grozi poślizgiem. Z drugiej strony w przypadku współpracy robota z ludźmi wykrycie i zgłoszenie wycieku podnosi poziom bezpieczeństwa, umożliwia wykrycie problemów z infrastrukturą (np. nieszczelności). Na podstawie obrazu z klasycznej kamery plamy wody i oleju nie zawsze są widoczne. Można je jednak odróżnić analizując odpowiedź spektralną.

W przypadku wody wykrywana jest duża absorpcja np. przy długości fal 970 nm, 1200 nm, 1450 nm, 1940 nm, a w przypadku oleju przy 1700–1800 nm i 2300–2350 nm. Kolejną istotną cechą jest to, że w przypadku oleju występuje zjawisko fluorescencji, w którym cząsteczki zawarte w oleju pochłaniają promieniowanie o krótszej długości fali (najczęściej nadfiolet, UV) i emitują światło o dłuższej długości fali. Dzięki opisanym powyżej cechom możliwa jest klasyfikacja wykorzystując kamery



Rys. 6. Dokładność określenia stopnia nawodnienia terenu dla algorytmu klasyfikacji realizowanego metodą SVM. Numery klas odpowiadają następującym stopniom nawodnienia substancji klasa 0 - 0%, klasa 1 : 14.28%, klasa 2 : 45.45%, klasa 3 : 50%, klasa 4 : 53.85%, klasa 5 : 57.14 %, klasa 6 : 60%

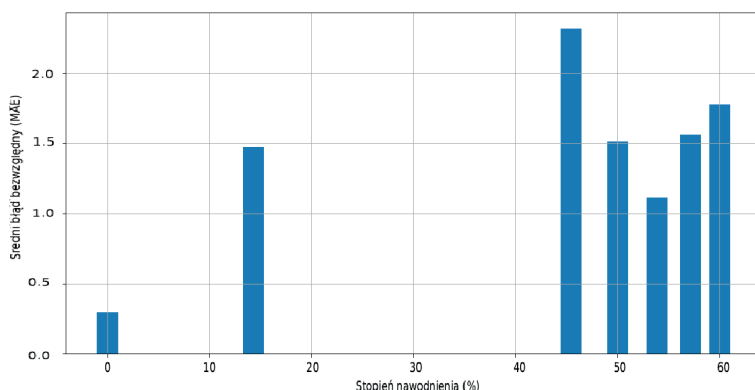
hiperspektralne.

Algorytm klasyfikacji plam jest dokładnie taki jak opisywany poprzednio. Dla wszystkich wymienionych algorytmów dokładność klasyfikacji przekraczała 95%. Należy zaznaczyć, że podobnie jak w zagadnieniach rozpoznawania powierzchni wynik można znacząco poprawić, jeśli każdy piksel będzie analizowany łącznie z jego sąsiedztwem.

5. Określenie stopnia nawodnienia terenu

Problem określenia stężenia substancji może być rozwiązywany na dwa sposoby. Pierwszy z nich to zadanie klasyfikacji, które możemy zdefiniować następująco: mając zadane stopnie stężenia substancji, chcemy dla danej próbki określić do której grupy należy. Druga metoda polega na przewidywaniu wartości ciągłej. W prowadzonych pracach testowane były obie metody. Zagadnienie klasyfikacji było rozwiązywane metodami opisywanymi w poprzednich rozdziałach. Na rys. 6 przedstawiono macierz błędów, dla algorytmu SVM, dla pozostałych algorytmów otrzymano podobne wyniki. Numery klas odpowiadają następującym stopniom nawodnienia substancji klasa 0 - 0%, klasa 1 : 14.28%, klasa 2 : 45.45%, klasa 3 : 50%, klasa 4 : 53.85%, klasa 5 : 57.14 %, klasa 6 : 60%.

W zagadnieniu przewidywania wartości ciągłej (procent nawodnienia) testowane były dwie metody - regresja liniowa i sieć neuronowa. Sieć neuronowa miała następującą architekturę: Warstwa wejściowa zawiera 480 neuronów (wszystkie długości fali). Pierwsza warstwa ukryta ma 256 neuronów i funkcję aktywacji ReLu. Druga



Rys. 7. Błąd określenia stopnia nawodnienia - sieć neuronowa

ukryta warstwa ma rozmiar 128 i też funkcję aktywacji ReLu. Dropout wyłącza losowe 30% neuronów tej warstwy. Warstwa wyjściowa to jeden neuron. W przypadku regresji liniowej błąd średni nie przekraczał 5%. W przypadku przewidywania wartości ciągłej zastosowanie algorytmu PCA powodowało pogorszenie dokładności wyników. Dlatego zarówno w pierwszym i drugim przypadku wektor cech nie był redukowany i składał się z 480 elementów. Dokładność predykcji była sprawdzana dla wartości dla których model nie był uczony.

6. Podsumowanie i przyszłe prace

Z przedstawionych wyników badań wynika, że kamery hiperspektralne i zastosowanie odpowiednich algorytmów sztucznej inteligencji umożliwia określenie różnego typu parametrów terenu z dużą dokładnością. Problemem jest jednak duży koszt sensorów i ogromna ilość danych, które należy przetworzyć. Systemy uczenia pod nadzorem wymagają dostarczenia dużej liczby poetykietowanych, zwykle ręcznie, danych. Aby zminimalizować ilość pracy, chcemy wprowadzić mechanizmy uczenia aktywnego [5]. W kolejnych pracach będziemy się koncentrować na selekcji najbardziej istotnych pasm spektralnych. Zmniejszenie liczby analizowanych pasm powinno znacząco usprawnić proces klasyfikacji, znając charakterystyki spektralne będzie istniała możliwość zastąpienia kamer hiperspektralnych przez dużo tańsze kamery multispektralne. W przypadku badania gęstości gruntu dużym problemem jest sytuacja, gdy górna warstwa grzęzawiska zaschnie. Kamera hiperspektralna bada górną powierzchnię więc błoto będzie zaklasyfikowane jako powierzchnia twarda. W takiej sytuacji warto zastosować dodatkowe sensory np. georadar lub kamerę termowizyjną.

Literatura

- [1] Telmo Adao et al. Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. Remote sensing, 2017, wolumen 9, numer 11, s. 1110.

- [2] B. Averill, P. Eldredge. *Principles of general chemistry: Principles, Patterns, and Applications*. Saylor Foundation 2011.
- [3] Ian T. Jolliffe, Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 2016, wolumen 374, numer 2065, s. 20150202.
- [4] Sidharth Prasad Mishra et al. Multivariate statistical data analysis-principal component analysis (PCA). *International Journal of Livestock Research*, 2017, wolumen 7, numer 5, s. 60–78.
- [5] B. Piotrowski et al. Leveraging active learning techniques for surgical instrument recognition and localization. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 2024, wolumen 72, numer 5, s. e150337.
- [6] Alpana Shukla, Rajsi Kot. An Overview of Hyperspectral Remote Sensing and its applications in various Disciplines. *IRA Int. J. Appl. Sci*, 2016, wolumen 5, numer 2, s. 85–90.
- [7] B. Siemiątkowska et al. Semantic-aware path planning with hexagonal grids and vehicle dynamic constraints. *Energies*, 2023, wolumen 16, numer 13.
- [8] J. Teague et al. A review of current and new optical techniques for coral monitoring. *Oceans*, 2022, wolumen 3, s. 30–45.
- [9] J. Tomsett, C.; Leyland. Development and testing of a UAV laser scanner and multispectral camera system for eco?geomorphic applications. *Sensors*, 2021, wolumen 21, s. 7719.

Application of hyperspectral camera in navigation of mobile robots

The following paper will provide an overview of the reaserch conducted at the Department of Mechatronics related to the use of hyperspectral camera in mobile robot navigation. The following topics are discussed: an introduction to spectral analysis, an algorithm for recognizing the type of surface on which the robot moves (asphalt,dirt road,grass, trees), a method for assessing the degree of hydration of the terrain, and an algorithm for recognizing water and oil stains on the surface. It is shown why the results of the research are relevant to navigation.

Estymacja prędkości liniowej i kątowej na podstawie odczytów skanera laserowego 2D z wykorzystaniem sztucznych sieci neuronowych

Wojciech Domski¹, Joanna Chrobot²

Streszczenie

Praca prezentuje wykorzystanie sztucznych sieci neuronowych do estymacji prędkości ruchu robota. W ramach pracy stworzono środowisko symulacyjne bazujące na oprogramowaniu ROS2 oraz Gazebo. Wykorzystując to środowisko zebrano dane pomiarowe takie jak odczyty odległości pochodzące z skanera laserowego 2D. Zebrane dane zostały użyte do wytrenowania kilku wariantów sieci neuronowych bazujących na warstwach konwolucyjnych i rekurencyjnych. W ramach pracy porównano trzy trajektorie: trajektorię rzeczywistą robota, trajektorię bazującą na odometrii platformy mobilnej oraz trajektorię wyliczoną na podstawie estymowanych prędkości pochodzących z wytrenowanych modeli. W pracy przedstawiono analizę otrzymanych wyników w oparciu o błędy estymacji oraz empiryczne podobieństwo otrzymanych trajektorii wyjściowych.

1. Wstęp

Jednym z podstawowych elementów podczas pracy z platformą mobilną jest właściwa estymacja jej pozycji. Znajomość dokładnej estymacji pozycji oraz orientacji jest kluczowa do wykonania rozmaitych zadań przez robota, takich jak śledzenie ścieżki, czy śledzenie trajektorii. Bez informacji o dokładnej rzeczywistej pozycji robota nie jest możliwe wykonanie tych zadań. Jedną z metod, która zapewnia informacje o pozycji platformy mobilnej w przestrzeni operacyjnej, jest odometria. Odometria pozwala na wyliczenie zadanej pozycji platformy w oparciu o prędkości obrotowe kół. Wykorzystując do tego celu model kinematyki, możliwe jest oszacowanie pozycji robota, jak i jego orientacji na podstawie odczytów sygnałów pochodzących z enkoderów przymocowanych do wałów silników. Jednakże, metoda ta jest mało dokładna w przypadku platformy klasy (2,0), gdyż ze względu na występowanie poślizgów estymata orientacji jest niedokładna, co przekłada się na kumulowanie błędów podczas aproksymacji pozycji w układzie kartezjańskim. Zaletą tej metody jest jej prostota implementacji i brak wymogu wykorzystania zaawansowanych sensorów.

Innym sposobem estymacji pozycji robota w zamkniętej przestrzeni są techniki

¹Katedra Cybernetyki i Robotyki, Wydział Elektroniki, Fotoniki i Mikrosystemów, Politechnika Wrocławska, ul. Janiszewskiego 11/17, 50-372 Wrocław, wojciech.domski@pwr.edu.pl

²Wydział Elektroniki, Fotoniki i Mikrosystemów, Politechnika Wrocławska, ul. Janiszewskiego 11/17, 50-372 Wrocław

należące do grupy jednoczesnej lokalizacji i mapowania (z ang. *Simultaneous Localization and Mapping, SLAM*) [2]. Grupa tych metod pozwala na lokalizację robota w nieznanym otoczeniu przy jednoczesnym tworzeniu jego mapy. Metody te wykorzystują do swojego poprawnego działania dodatkowe sensory takie jak laserowe skanery 2D, lidary, czy kamery [5], [8], [7]. Zaletą tego typu rozwiązań jest tworzenie precyzyjnej mapy oraz lokalizacji na podstawie zebranych danych. Z kolei do wad tych metod można zaliczyć wymóg posiadania dodatkowego sensora, który jest w stanie dostarczyć nowe dane na temat lokalnego otoczenia. Dodatkowo, algorytmy typu SLAM wymagają operacji, które są wymagające obliczeniowo. Dodatkowo, niektóre algorytmy z tej grupy charakteryzują się problemami podczas specyficznych warunków, jak na przykład występowanie cykli.

Jest wiele podejść, które pozwalają na wyliczenie odometrii, a w których finalnie uzyskuje się lepsze estymaty niż przy wykorzystaniu jedynie prędkości kół [4], [1]. Wreszcie można znaleźć rozwiązania, które wykorzystują sieci neuronowe do poprawy działania systemu [6]. W tej pracy skupiono się na podejściu do oszacowania pozycji robota, które wykorzystując dodatkowe oprzyrządowanie, a konkretnie skaner laserowy 2D, jest w stanie zwiększyć dokładność aproksymacji pozycji. Proponowane rozwiązanie bazuje na pomiarach pochodzących ze skanera laserowego oraz modelu sztucznej sieci neuronowej. Zadaniem sieci neuronowej jest estymacja prędkości liniowej oraz prędkości kątowej. Na ich podstawie oraz na podstawie znajomości modelu kinematyki platformy mobilnej klasy (2,0) możliwe jest wyliczenie lokalnej pozycji robota w przestrzeni.

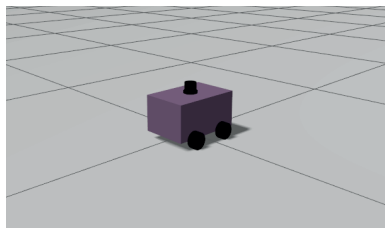
Praca skupia się przede wszystkim na zbadaniu różnych podejść do estymacji prędkości wykorzystując odmienne architektury sieci neuronowych jak i różne formowanie danych wejściowych. W rozdziale 2 przedstawiono wykorzystane środowisko symulacyjne, w którym zbierano dane treningowe oraz wykonano przejazdy testowe. W rozdziale 3 zostały zaprezentowane wykorzystane architektury sieci neuronowych oraz opisano proces formowania danych wejściowych. Rozdział 4 zawiera wyniki przedstawiające jakość estymacji dla zbadanych architektur sieci neuronowych oraz ich analizę. Podsumowanie pracy wraz z potencjalnymi dalszymi kierunkami badań nad zaprezentowaną metodą zostały omówione w rozdziale 5.

2. Środowisko symulacyjne

Środowisko symulacyjne zostało przygotowane w oparciu o framework ROS2 w wersji Jazzy Jalisco oraz symulator Gazebo w wersji Harmonic. Symulator Gazebo pozwala na symulowanie otoczenia robota mobilnego oraz fizyki świata, w jakim ten robot został umieszczony. Oprogramowanie to pozwala na dokładne odzwierciedlenie rzeczywistego robota w danym środowisku. Ponadto oferuje ono również szereg rozmaitych czujników, które precyzyjnie wykonują pomiary. Jednym z takich czujników jest skaner laserowy 2D, który pozwala na wykonywanie dookólnego pomiaru odległości. W oparciu o wykonane pomiary możliwa jest budowa mapy najbliższego otoczenia. W celu połączenia kontrolera robota z jego modelem symulowanym w oprogramowaniu Gazebo wykorzystany został framework ROS2. Pozwala on na odczytywanie wartości pomiarowych ze środowiska pomiarowego oraz ich dalszą obróbkę.

Model robota, który został wykorzystany, przedstawiono na rysunku 1. Jest to plat-

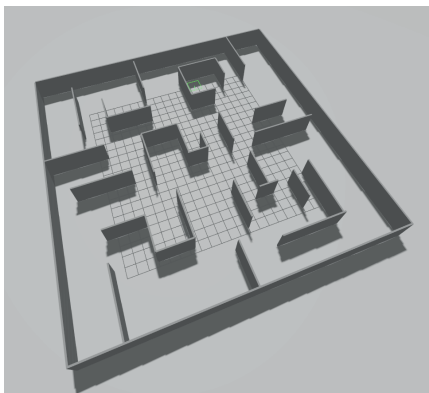
forma typu SSMP (z ang. *Skid-steering mobile platform*) posiadająca 4 niezależnie napędzane koła. W górnej części robota został zamontowany skaner laserowy, który zbiera informacje o otoczeniu w krótkich odstępach czasowych. Dzięki tym odczytom oraz ich analizie robot może m.in. wykrywać przeszkody. Środowisko odzwierciedla-



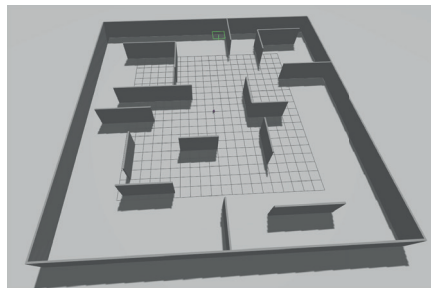
Rys. 1. Model platformy mobilnej

jące pomieszczenie, po którym poruszała się platforma mobilna, zostało przedstawione na rysunku 2a. Zaprojektowane zostało ono w kształcie przypominającym labirynt. Taka budowa pomieszczenia pozwala na wykrywanie punktów charakterystycznych, a w szczególności poprawną estymację prędkości. Natomiast na rysunku 2b przedstawiono pomieszczenie, które posłużyło do ewaluacji wytrenowanych modeli.

a



b



Rys. 2. Środowisko robotaa – widok mapy treningowej, b – widok mapy testowe

Na potrzeby trenowania modeli zebrano łącznie 33611 próbek. Zbiór treningowy liczył około 87% wszystkich próbek, zbiór testowy około 10%, natomiast pozostałe próbki 3% zostały zaliczone do zbioru testowego. Ponadto, w ramach pracy wykorzystano sztuczne rozszerzanie zbioru danych. Pierwotna baza została zaszumiona szumem gaussowskim o niewielkiej wariancji i wartości średniej wynoszącej 0. Dzięki temu pierwotny zbiór został rozszerzony prawie dziesięciokrotnie pozwalając na otrzymanie około 300000 próbek.

3. Architektura sieci neuronowej

W ramach pracy zbadano dwa warianty modeli sieci neuronowych. Pierwszym z nich była sieć wykorzystująca warstwy konwolucyjne [3]. Tego typu warstwy są bardzo często wykorzystywane podczas tworzenia modeli do przetwarzania informacji w formie obrazów. Drugi z wariantów to sieć oparta o warstwy rekurencyjne, które używane są do przetwarzania czasowych serii danych, w celu wykrycia wzorców temporalnych. Dla każdego wariantu sieci neuronowej zostały stworzone po dwa osobne modele. Pierwszy z modeli jest wyspecjalizowany do estymacji prędkości liniowej, natomiast drugi do estymacji prędkości kątowej.

3.1. Wektor wejściowy

Założono, że skaner laserowy 2D dostarcza 360 punktów (odległości) na pojedynczy skan. Dla rekurencyjnych sieci neuronowych cały skan jest wektorem wejściowym. Dzięki specyfice przetwarzania sygnałów w sieciach rekurencyjnych informacja ta jest wystarczająca, ponieważ wewnętrzne struktury sieci RNN rozpoznają zależności między dostarczonymi próbkami w czasie. Z kolei w przypadku sieci konwolucyjnych wymagane jest odpowiednie kształtowanie wejścia. Sieć konwolucyjna nie przetwarza zależności czasowych pomiędzy dostarczonymi próbkami. Z tego powodu wejście sieci konwolucyjnej otrzymuje odpowiednio przetworzone dane. Przetworzenie danych polega na wyliczeniu różnicy pomiędzy próbkami. Dane wejściowe przetwarzane są za pomocą

$$v_i = d_i - d_{i-1}, \quad (1)$$

gdzie d_i to wektor odczytanych odległości pochodzących z skanera laserowego, d_{i-1} to wektor odległości, ale w poprzedniej chwili czasu. Podejście opisane jako (1) jest adekwatne w przypadku estymacji prędkości liniowej poruszającego się robota. Jednakże, chcąc estymować prędkość kątową, której znajomość pozwoli na obliczenie pozycji robota, możliwy jest jeszcze jeden zabieg, który polega na przesunięciu wektora o zadaną liczbę elementów. Wówczas wektor danych wejściowych może zostać poddany następującej operacji

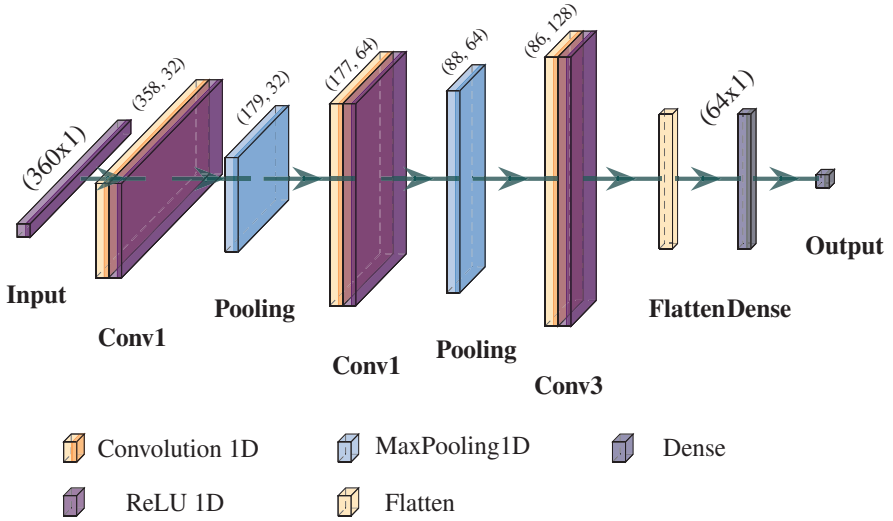
$$d^k = [d_{n-k+1,n}, d_{1,n-k}], \quad (2)$$

gdzie n to liczba elementów wektora, a k to liczba elementów, o ile należy przesunąć wektor w prawą stronę. Innymi słowy, k elementów z końca wektora przechodzi na jego początek. Wówczas formowanie danych wejściowych w celu określenia prędkości kątowej przyjmuje postać

$$\omega_i = d_i - d_{i-1}^1. \quad (3)$$

3.2. Sieci konwolucyjne

Architektura sieci konwolucyjnej została zaprezentowana na rysunku 3. Dla tego typu sieci w przypadku estymaty prędkości liniowej dane wejściowe formowane są jako v_i , natomiast w modelu, który ma za zadanie estymować prędkość kątową dane wejściowe podawane są jako ω_i .



Rys. 3. Architektura sieci knowolucyjnej do predykcji prędkości

3.3. Sieci rekurencyjne

W przypadku sieci rekurencyjnych wektor danych wejściowych został zredukowany z 360 do 36 elementów, w taki sposób, że w celu utworzenia próbki z danymi wejściowymi tylko dziesiąty element z pierwotnego wektora odległości był zapisywany w nowym wektorze. Zabieg ten pozwolił na zredukowanie liczby elementów w wektorze. Ze względu na formowanie danych wejściowych zdecydowano się na wykorzystanie trzech różnych wariantów:

1. dane z pojedynczego pomiaru,
2. połączone dane z bieżącego pomiaru oraz poprzedniego,
3. przeplatane dane z bieżącego pomiaru oraz poprzedniego.

Ponadto dla każdego z trzech powyższych przypadków zostały stworzone dwie osobne architektury sieci rekurencyjnych. Pierwszy wariant wykorzystywał w swojej strukturze komórki LSTM, natomiast drugi komórki GRU.

Architektura sieci rekurencyjnej dla przypadku wektora wejściowego składającego się wyłącznie z pojedynczego pomiaru została pokazana na rysunku 4.

Dane wejściowe dla drugiego wariantu można opisać jako

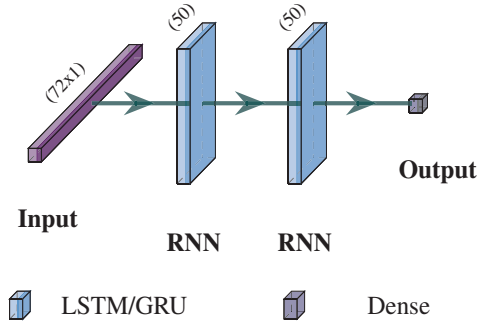
$$d'_i = [d_i, d_{i-1}]. \quad (4)$$

Po takiej operacji rozmiar wektora danych wejściowych został podwojony, co można zauważyć na rysunku 4 przedstawiającym architekturę sieci dla tego wariantu.

Podobnie jak to miało miejsce dla wariantu drugiego sieci rekurencyjnej, wektor danych wejściowych został uformowany jako

$$d''_i = [d_i^1, d_{i-1}^1, d_i^2, d_{i-1}^2, \dots, d_i^n, d_{i-1}^n], \quad (5)$$

gdzie n to liczba elementów w zredukowanym wektorze wejściowym, natomiast notacja d_i^j oznacza pojedynczy j -ty element i -tej próbki. Architektura dla tego modelu



Rys. 4. Architektura sieci rekurencyjnej do predykcji prędkości

jest identyczna jak we wcześniejszym wariancie. Różnica występuje jedynie w uczeniu modelu poprzez zastosowanie innej metody kształtowania wejścia.

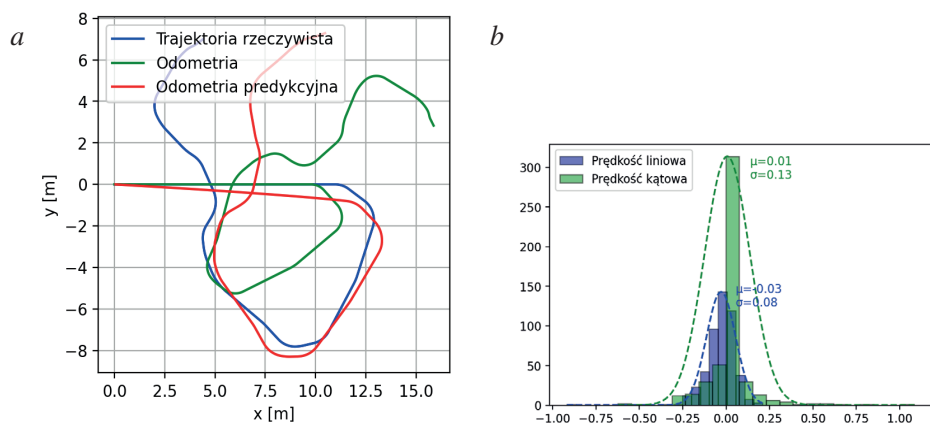
4. Estymacja prędkości i pozycji

W celu porównania jakości estymacji prędkości liniowej oraz prędkości kątowej wykorzystano trajektorię referencyjną. W trakcie przeprowadzonych symulacji platforma mobilna podążała za określoną trajektorią referencyjną. W trakcie przejazdu były zbierane dane o rzeczywistym położeniu robota w przestrzeni, prędkościach obrotowych kół robota oraz skany laserowe. Następnie, w celu lepszego odzwierciedlenia uzyskanych wyników, wyliczono trajektorię na podstawie odometrii robota. Dla każdego modelu wykonano predykcję prędkości liniowej i kątowej oraz na ich podstawie wyliczono trajektorię. Zestawienie trzech trajektorii pozwala na zauważenie różnic pomiędzy tradycyjną odometrią robota, a odometrią wyliczoną na podstawie predykcji.

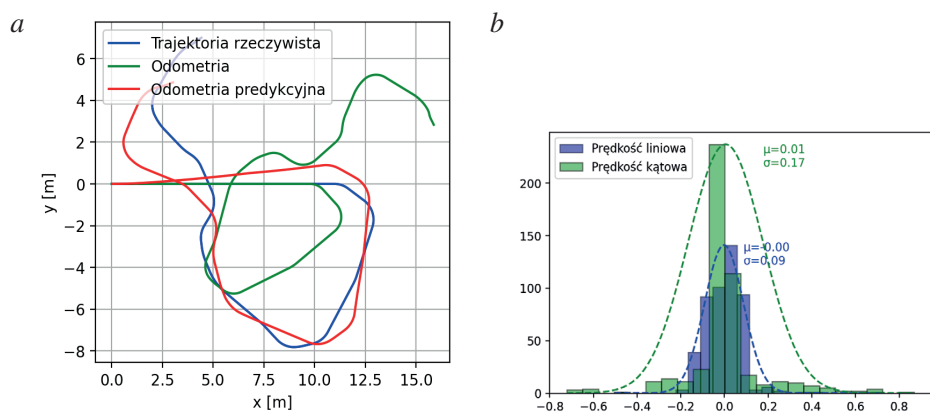
Aby lepiej scharakteryzować jakość predykcji, zdecydowano się na przedstawienie różnic prędkości (pomiędzy rzeczywistą a estymowaną) w formie histogramów. W przypadku, gdy histogramy przedstawiałyby jedynie pojedynczy prążek w zerze, wówczas znaczyłoby to, że predykcja odpowiada idealnie rzeczywistym wartościom prędkości. Jednakże estymacja prędkości jest obarczona błędem, a zatem można określić dwa kryteria, które pozwolą w porównaniu jakości estymat. Pierwszym z nich jest wartość średnia. W przypadku idealnym wartość średnia różnicy predykcji powinna wynosić zero. Drugim parametrem jest odchylenie standardowe; im lepsza predykcja, tym niższa wartość odchylenia.

Na rysunku 5b przedstawiono trajektorię i histogram różnic prędkości dla modelu bazującego na warstwach konwolucyjnych. Prędkości w obu przypadkach są estymowane stosunkowo dobrze. Wartość średnia jak i odchylenie standardowe są niskie. Znajduje to również swoje odzwierciedlenie w jakości śledzenia trajektorii (rys. 5a).

Jakość estymacji dla modelu opartego na komórkach LSTM oraz przeplataniu danych wejściowych można zobaczyć na rysunku 6a. Analiza statystyczna pokazuje, że odchylenia standardowe są wyższe niż w poprzednim przypadku. Natomiast wartość średnia błędu dla prędkości kątowej wynosząca $0.01 \frac{\text{rad}}{\text{s}}$ (rys. 6b) może świadczyć o przyczynie odchylenia estymowanej trajektorii od rzeczywistej.



Rys. 5. Model sieci konwolucyjnej – zestawienie trajektorii, b – histogramy błędów

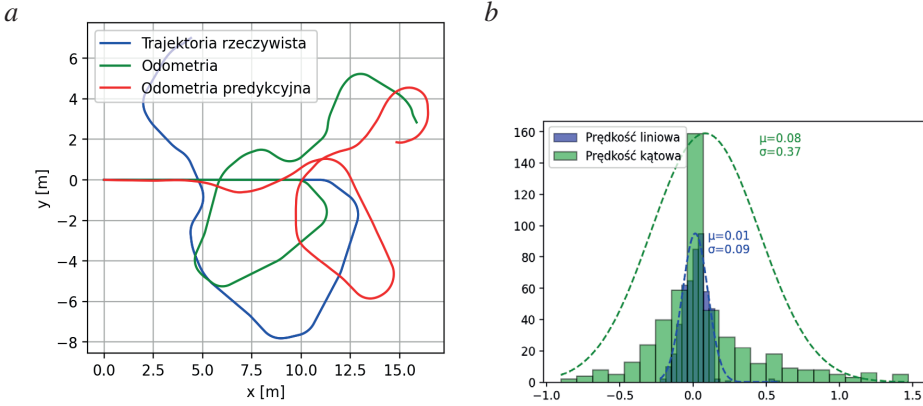


Rys. 6. Model sieci rekurencyjnej LSTM, przeplatane dane wejściowe
a – zestawienie trajektorii, b – histogramy błędów

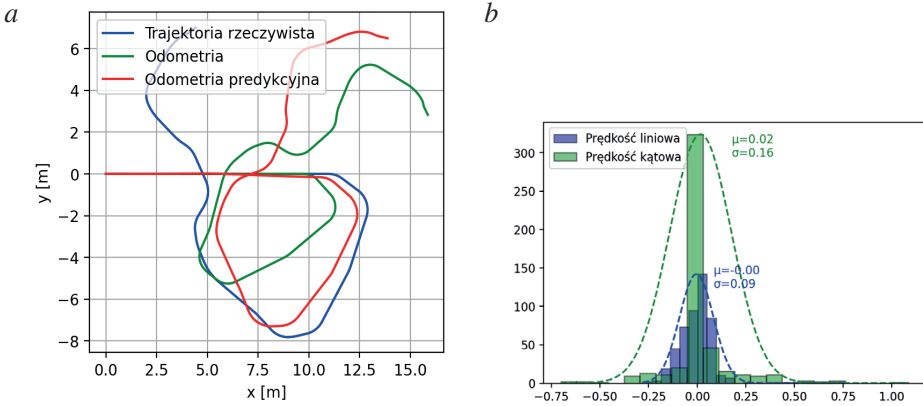
Dla modelu LSTM z wejściem składającym się z połączonych wektorów można zobaczyć, że różnica dla estymaty prędkości kątowej ma stosunkowo duże odchylenie standardowe (rys. 7b). Przekłada się to bezpośrednio na jakość estymowanej trajektorii. Jej kształt znacząco odbiega od trajektorii rzeczywistej, co w dużej mierze może być związane z jakością estymacji prędkości kątowej (rys. 7a).

Zostały przeprowadzone również badania dla sieci RNN z wykorzystaniem komórek GRU. Jak można zobaczyć na rysunku 8a estymowana trajektoria dość dobrze odzwierciedla trajektorię rzeczywistą. Świadczą o tym również niewielkie wartości odchyłeń standardowych, które można dojrzeć na rysunku 8. Są one zbliżone do wartości uzyskanych z modelu bazującego na architekturze CNN.

Wykorzystując tę samą architekturę sieci RNN, to jest bazującą na komórkach GRU oraz łączeniu wektorów zamiast ich przeplatania, można zaobserwować wyniki estymacji odometrii na rysunku 9a. W tym przypadku błędy estymacji prędkości kątowej

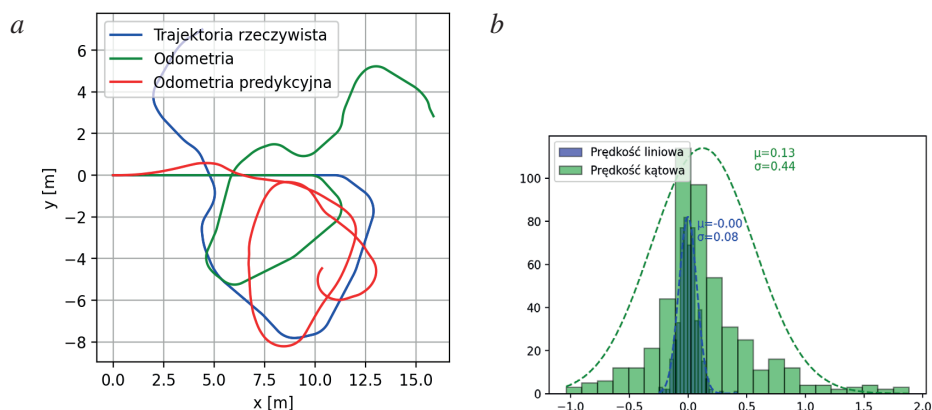


Rys. 7. Model sieci rekurencyjnej LSTM, połączone dane wejściowe
a – zestawienie trajektorii, b – histogramy błędów



Rys. 8. Model sieci rekurencyjnej GRU, przeplatane dane wejściowe
a – zestawienie trajektorii, b – histogramy błędów

wej mają stosunkowo dużą wartość odchylenia standardowego. Porównując otrzymane wyniki, można zauważyć, że sztuczne sieci neuronowe bazujące na architekturze CNN, jak i na architekturze RNN, pozwoliły na otrzymanie dobrych jakości estymat prędkości, a tym samym dobrych przewidywań trajektorii. Zbiorcze zestawienie wartości statystycznych estymat prędkości liniowej i kątowej zostało zebrane w tabeli 1. Jak można zauważyć, każdy rodzaj sieci radzi sobie podobnie w przypadku estymacji prędkości liniowej. Również w tym przypadku widać, że odchylenia standardowe bez względu na rodzaj sieci są podobne. W przypadku estymacji prędkości kątowych okazało się, że najlepiej radzą sobie modele oparte o sieci CNN oraz sieci rekurencyjne, ale tylko w przypadku przeplatane formowania wektora wejściowego. W pozostałych przypadkach wielkość wartości średniej, jak i odchylenia standardowego jest stosunkowo duża. Na dalszym etapie rozważań skoncentrowano się na dwóch modelach. Modelu opartym o sieci CNN oraz modelu rekurencyjnym wykorzystują-



Rys. 9. Model sieci rekurencyjnej GRU, połączone dane wejściowe
a – zestawienie trajektorii, b – histogramy błędów

Tab. 1. Porównanie wielkości statystycznych różnicy estymowanych prędkości

Model	Prędkość liniowa		Prędkość kątowa	
	μ	σ	μ	σ
CNN	0.03	0.08	0.01	0.13
LSTM, przeplatane	0.00	0.09	0.01	0.17
LSTM, połączone	0.01	0.09	0.08	0.37
GRU, przeplatane	0.00	0.09	0.02	0.16
GRU, połączone	0.00	0.08	0.13	0.44

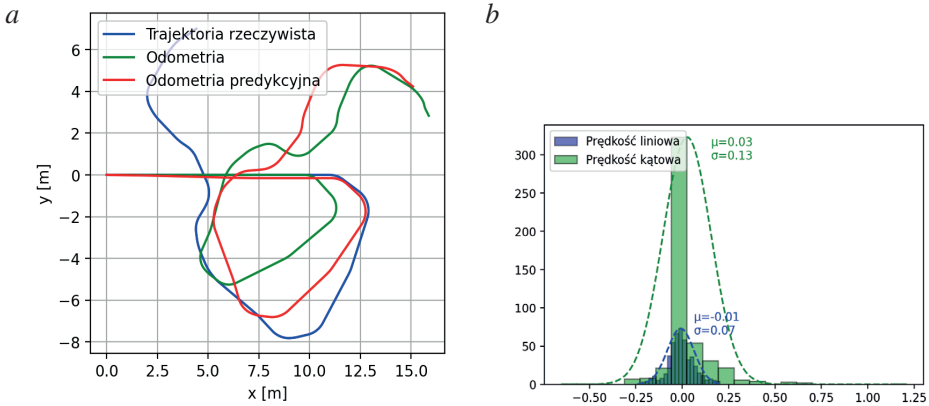
cym komórki GRU z przeplatanym wejściem. Modele zostały wytrenowane ponownie na rozszerzonym zbiorze danych, a następnie przetestowane w dwóch środowiskach (znanym i testowym).

Na rysunku 10 widać nieznaczną poprawę estymacji prędkości liniowej. Estymacja prędkości kątowej nieznacznie się pogorszyła. W tym przypadku estymowana odometria przybliżyła się do trajektorii otrzymanej za pomocą odometrii klasycznej. Model ten został zwalidowany w środowisku testowym (rys. 11). W tym przypadku widać, że estymacja prędkości kątowej jest dużo gorsza, co spowodowało dużą rozbieżność pomiędzy trajektorią rzeczywistą a trajektorią pochodzącą z estymowanej odometrii.

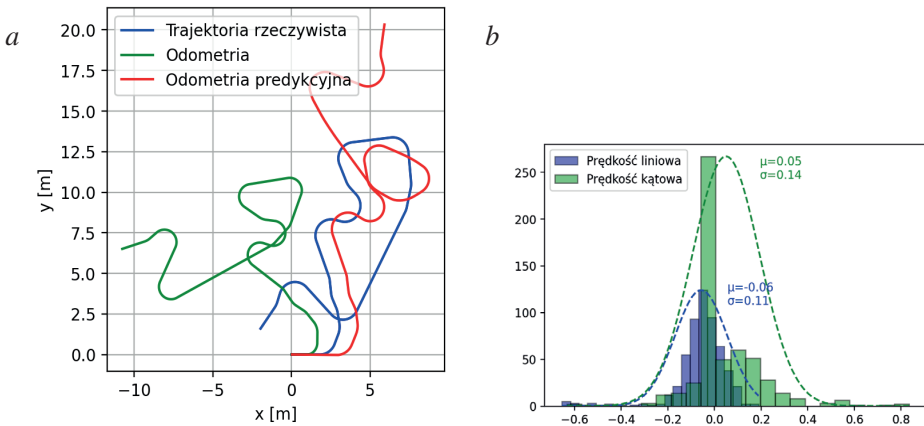
W przypadku rozszerzonego modelu rekurencyjnego GRU operacja rozszerzenia danych przyniosła oczekiwane efekty zarówno dla środowiska treningowego (rys. 12) jak i nieznanego środowiska testowego (rys. 13). W obu wskazanych przypadkach jakość estymat uległa poprawie, a wynikowa trajektoria lepiej odzwierciedlała trajektorię rzeczywistą.

5. Podsumowanie

W pracy skupiono się przede wszystkim na zbadaniu różnych rodzajów architektur sztucznych sieci neuronowych wykorzystanych w celu estymacji pozycji. Odometria



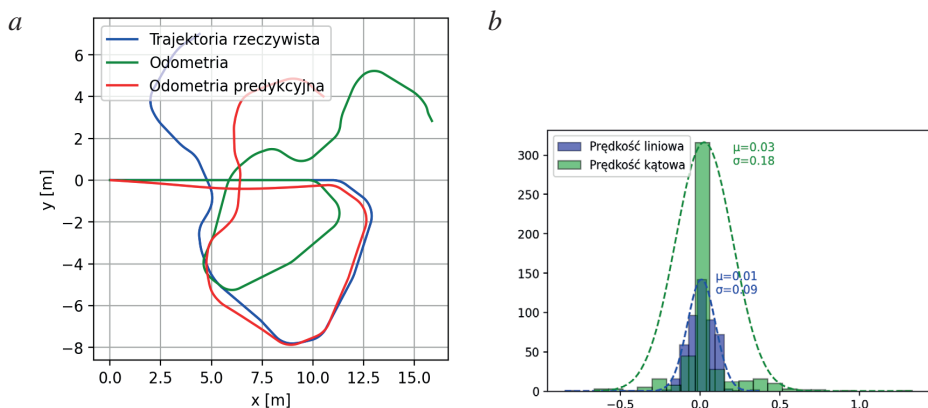
Rys. 10. Rozszerzony model sieci konwulucyjnej
a – zestawienie trajektorii, b – histogramy błędów



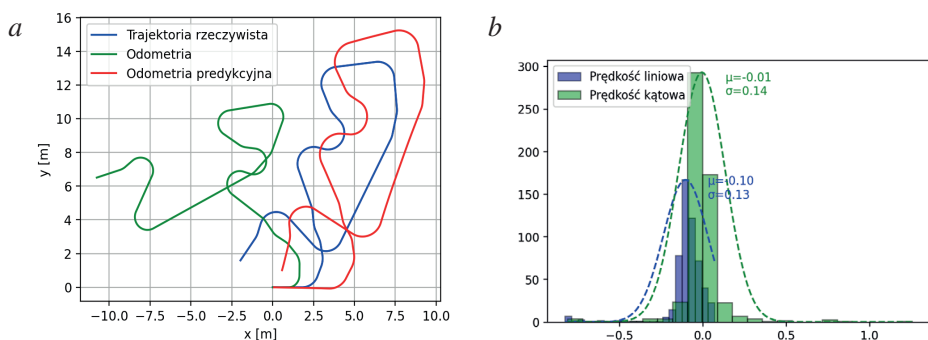
Rys. 11. Rozszerzony model sieci konwulucyjnej, środowisko testowe
a – zestawienie trajektorii, b – histogramy błędów

robota jest bardzo ważnym aspektem planowania i ruchu platformy mobilnej, ponieważ to dzięki niej możliwe jest, często zgrubne, określenie pozycji i orientacji robota w przestrzeni. Metody odometrii robota bazujące jedynie na odczytach prędkości z kół są dokładne, ale tylko w krótkich odcinkach czasu. Błędy estymacji pozycji robota kumulują się, a po dłuższym czasie są na tyle duże, że estymata znacząco odbiegają od rzeczywistych wartości. Na jakość predykcji ma przede wszystkim obecność poślizgów kół pojazdu. Całkowite wyeliminowanie tego zjawiska jest niemożliwe, a zatem odometria bazująca jedynie na prędkości kół jest zawodna.

Wykorzystanie sieci neuronowych do szacowania prędkości liniowej i kątowej oraz użycie dodatkowego źródła danych, jakim są odczyty ze skanera laserowego 2D, daje dużo lepsze rezultaty niż bazowanie wyłącznie na prędkościach obrotowych kół. W badaniach zaproponowano dwie odmienne architektury: sieci konwulcyjne oraz



Rys. 12. Rozszerzony model sieci rekurencyjnej GRU, przeplatane dane wejściowe
a – zestawienie trajektorii, b – histogramy błędów



Rys. 13. Rozszerzony model sieci rekurencyjnej GRU, przeplatane dane wejściowe, środowisko testowe
a – zestawienie trajektorii, b – histogramy błędów

sieci rekurencyjne. Sieci konwolucyjne są powszechnie wykorzystywane w problemach dotyczących analizy obrazu. Natomiast sieci rekurencyjne używane są do analizy oraz predykcji szeregów czasowych, gdzie ewolucja danych w czasie ma kluczowe znaczenie. Aby wykorzystać wspomniane dwie architektury sieci, wymagane było odpowiednie przetworzenie i formowanie danych, które były podawane na wejście modeli. W trakcie badań porównano architekturę sieci CNN oraz cztery warianty sieci rekurencyjnych. Z analizy otrzymanych rezultatów wynika, że sieci konwolucyjne bardzo dobrze nadają się do postawionego zadania estymaty prędkości podczas ruchu robota. Równocześnie sieci rekurencyjne, w zależności od ułożenia danych wejściowych, mogą się spisywać lepiej, bądź gorzej. W przypadku formowania danych jako przeplatające się, widać, że jakość estymat jest dużo lepsza niż w przypadku danych połączonych. Głównie jest to widoczne w przypadku szacowania prędkości kątowych, podczas gdy dla prędkości liniowych modele dają porównywalne wyniki.

W dalszych badaniach planowane jest zrealizowanie eksperymentu na rzeczywi-

stej platformie mobilnej. Dzięki takim badaniom możliwe będzie określenie, jak zaproponowane metody będą w stanie estymować prędkości ruchu robota w realnych warunkach. Kolejnym kierunkiem badań jest sprawdzenie, czy zaproponowanie głębszych modeli posiadających większą liczbę warstw ukrytych będzie korzystne dla jakości estymat. Również na uwagę zasługuje możliwość sprawdzenia, czy zunifikowanie architektur w taki sposób, aby jednocześnie dostarczały szacunków obydwu prędkości, jest możliwe. Takie połączenie korzystnie wpłynęłoby na czas inferencji, skracając go.

Literatura

- [1] L. Blocher et al. Gyroscope-Aided Odometry Navigation Using a Highly-Precise Automotive MEMS IMU Complemented by a Low-Cost Sensor Array. In: IEEE International Symposium on Inertial Sensors and Systems. *Proceedings*, Awinion, Francja, 2022, s. 1–4.
- [2] J. Gaia et al. Mapping the Landscape of SLAM Research: A Review. *IEEE Latin America Transactions*, 2023, s. 1313–1336.
- [3] B. Kaleci, K. Turgut, H. Dutagaci. 2DLaserNet: A deep learning architecture on 2D laser scans for semantic classification of mobile robot locations. *Engineering Science and Technology, an International Journal*, 2021.
- [4] G. L. Rezende, F. O. Silva, L. S. Marques. Comparison of Wheel Odometry and Visual Odometry for Low-Cost Vehicle Navigation. In: Brazilian Symposium on Robotics and Workshop on Robotics in Education. *Proceedings*, Goiânia, Brazylia, 2024, s. 91–96.
- [5] S. H. Liu et al. Improved visual odometry system based on kinect RGB-D sensor. In: International Conference on Consumer Electronics. *Proceedings*, Berlin, Niemcy, 2017, s. 29–30.
- [6] D. Xu, Z. Zhang. Application and Analysis of Recurrent Convolutional Neural Network in Visual Odometry. In: International Conference on Mechatronics and Automation. *Proceedings*, Guilin, Chiny, 2022, s. 1222–1226.
- [7] O. Yilmaz, F. Karakus. Stereo and kinect fusion for continuous 3D reconstruction and visual odometry. In: International Conference on Electronics, Computer and Computation. *Proceedings*, Ankara, Turcja, 2013, s. 115–118.
- [8] D. Zhang, T. Peng, J. S. Loomis. Optimized Deep Learning for LiDAR and Visual Odometry Fusion in Autonomous Driving. *IEEE Sensors Journal*, 2023, s. 29594–29604.

Linear and angular velocity estimation based on 2D laser scanner readings using artificial neural networks

The paper presents the use of artificial neural networks to estimate the velocities of a robot's movement. As part of the paper, a simulation environment based on ROS2 and Gazebo software was created. Using this environment, measurement data such as distance readings from a 2D laser scanner were collected. The collected data were used to train several variants of neural networks based on convolutional and recurrent layers. The paper compares three trajectories: the actual robot trajectory, a trajectory based on the odometry of the mobile platform, and a trajectory calculated based on estimated speeds from trained models. The paper presents an analysis of the obtained results based on estimation errors and the empirical similarity of the resulting output trajectories.

TEORIA MASZYN I MECHANIZMÓW

Praca układu sił wykonana przy przemieszczeniu mobilnego robota z kołami mecanum

Marcin Szuster¹, Mateusz Szeremeta¹

Streszczenie

Roboty mobilne z kołami mecanum dysponują istotną przewagą w zakresie planowania i realizacji ruchu w stosunku do konstrukcji wyposażonych w standardowe koła np. pneumatyczne. Możliwość ruchu w dowolnym kierunku z dowolną orientacją ramy robota jest szczególnie atrakcyjna w przypadku przemieszczania w środowisku, gdzie występują ograniczenia przestrzeni do manewrowania np. wąskie ciągi komunikacyjne. Tak szerokie możliwości ruchu prowadzą do pytania: jaka orientacja ramy robota w trakcie ruchu będzie najkorzystniejsza? W niniejszym artykule sformułowano odpowiedź na to pytanie, za kryterium przyjmując pracę układu sił wykonaną przy przemieszczeniu robota. Na podstawie testów numerycznych i badań weryfikacyjnych dla różnych torów ruchu i orientacji początkowych ramy robota wyznaczono i porównano pracę układu sił. Przeprowadzone badania umożliwiły sformułowanie wniosków dotyczących preferowanej orientacji ramy robota z kołami mecanum w trakcie ruchu.

1. Wprowadzenie

Roboty mobilne są stosowane coraz powszechniej do realizacji różnego typu zadań, które wymagają przemieszczania w środowisku z ograniczeniami. Przeshkody pojawiające się w otoczeniu robotów, czy kształt dostępnej przestrzeni operacyjnej, mogą w znacznym stopniu ograniczać swobodę planowania toru ruchu robota. Współdzielenie dostępnych szlaków transportowych z ruchem pieszych i innych środków transportu może być powodem wzrostu preferencji wyboru konstrukcji robotów zapewniających wysoką manewrowość na ograniczonej przestrzeni. Przykładem takich rozwiązań są roboty z kołami omnikierunkowymi, np. kołami mecanum, które umożliwiają ruch pojazdu w dowolnym kierunku z dowolną orientacją.

Planowanie ruchu mobilnego robota z kołami mecanum (MRKM) wymaga wyznaczenia zadanego toru ruchu wybranego punktu związanego z ramą robota, np. środka masy, oraz kąta obrotu β ramy względem przyjętego nieruchomego układu odniesienia (często przyjmowana jest wartość stała kąta, np. $\beta = 0$). Jeżeli zatem kąt β może przyjmować dowolne wartości w trakcie ruchu, pojawia się pytanie, jaka jego wartość będzie najbardziej korzystna ze względu na wybrane kryterium. W niniejszym artykule przedstawiono wyniki badań wpływu orientacji ramy MRKM na pracę układu sił niezbędną do realizacji założonego ruchu. Przeprowadzone badania umożliwiły sformułowanie wniosków dotyczących preferowanej orientacji ramy robota w trakcie ruchu. Oczywiście jest, że w określonych zadaniach wymagana orientacja ramy

¹ Politechnika Rzeszowska im. Ignacego Łukasiewicza, al. Powstańców Warszawy 12, 35-959 Rzeszów, mszuster@prz.edu.pl, m.szeremeta@prz.edu.pl, <https://prz.edu.pl/>

robota może być ściśle określona (np. robot transportuje ładunek, który w pewnym zakresie orientacji ramy może doznać kolizji z elementami środowiska, czy wymagane jest osiągnięcie określonego położenia z zadaną orientacją) i tego typu kryterium będzie miało dominującą wagę na etapie planowania zadanej trajektorii ruchu.

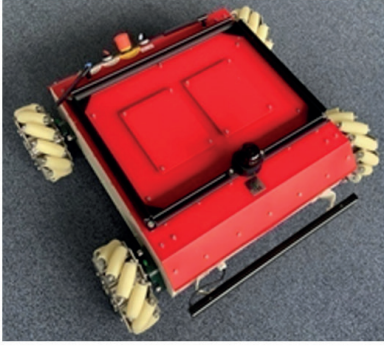
W literaturze można znaleźć pozycje podejmujące tematykę wyznaczania lub pomiaru ilości energii niezbędnej do realizacji ruchu MRKM. W artykule [4] autorzy włączyli kryterium minimalizacji zużycia energii do zaproponowanego wcześniej algorytmu planowania i realizacji ruchu MRKM. Przeprowadzone testy numeryczne oraz badania weryfikacyjne potwierdziły poprawność przyjętego modelu estymacji poboru energii przez robota oraz zakładaną redukcję zużycia energii w przypadku realizacji trajektorii zaplanowanej z zastosowaniem zaproponowanego algorytmu w porównaniu z układem sterowania, w którym nie uwzględniono aspektu minimalizacji zużycia energii. W artykule zrealizowano pomiar zużycia energii w sensie elektrycznym, bazując na zmierzonym natężeniu prądu zasilającego robota. Z całkowitego zużycia energii wyodrębniono zużycie energii przez układ ruchu robota, jednakże nie przekraczało ono 15% całkowitego zużycia energii. Ze względu na masę robota dwukrotnie przekraczającą masę robota stosowanego w niniejszych badaniach i zastosowanie modułów napędowych o czterokrotnie mniejszej mocy, oraz niewielki udział mocy pobieranej przez układy wykonawcze w stosunku do całkowitej mocy dostarczanej, trudno otrzymane wyniki odnieść do prezentowanych w artykule.

W publikacji [5] przedstawiono wyniki badań związanych z dopracowaniem modelu wyznaczania zużycia energii przez MRKM w przypadku ruchu po torze prostoliniowym z uwzględnieniem ruchu po płaskiej powierzchni, wjazdu na równię pochyłą oraz zjazdu z równi. Autorzy zauważyli istotny wpływ kąta nachylenia równi na pobór mocy układów napędowych, określili również maksymalny kąt równi, na który prezentowana konstrukcja była w stanie wjechać, na około 20° . Jednakże w badaniach nie uwzględniono wpływu orientacji ramy robota w trakcie ruchu na pobór mocy, a samo zużycie energii było mierzone na podstawie parametrów zasilania robota.

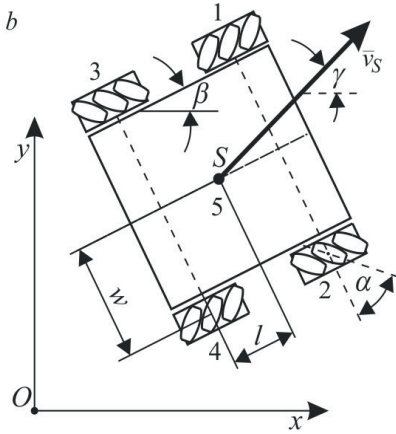
Interesujące wyniki badań przedstawiono w pracy [6], gdzie wyniki zużycia energii przez MRKM otrzymane z zaproponowanego modelu porównano z pomiarami zrealizowanymi w ramach badań weryfikacyjnych. Wyznaczono pobór mocy dla różnych wartości kąta θ MRKM zdefiniowanego jako kąt między wzdłużną płaszczyzną symetrii ramy robota i kierunkiem wektora prędkości środka masy robota. Zauważono, że najmniejszy pobór mocy występuje w trakcie ruchu, gdy $\theta = \frac{\pi}{4}$, a największy gdy $\theta = \frac{\pi}{2}$. Jednakże w omawianej publikacji analizowano jedynie ruch MRKM po torze prostoliniowym, nie uwzględniono ruchu po torze krzywoliniowym, ani kwestii zmiany orientacji ramy robota w trakcie ruchu.

Niniejszy artykuł składa się z czterech części. Po krótkim wstępie nakreślającym istotę problemu omówiono obiekt sterowania. Przedstawiono równania kinematyki niezbędne do wyznaczenia trajektorii ruchu robota, model dynamiki robota zastosowany do rozwiązania zadania odwrotnego dynamiki oraz sposób wyznaczenia pracy układu sił powodujących ruch robota. W kolejnym rozdziale przedstawiono wyniki badań symulacyjnych ruchu MRK, stanowisko badawcze oraz wyniki badań weryfikacyjnych. Artykuł kończy krótkie podsumowanie i wykaz literatury.

a



b



Rys. 1. a – Robot Husarion Panther, b – schemat mobilnego robota z kołami mecanum

2. Obiekt sterowania

Obiektem sterowania jest MRKM Husarion Panther (rys. 1a), którego schemat przedstawiono na rys. 1b, gdzie zastosowano następujące oznaczenia: 1, ..., 4 – koła mecanum, 5 – rama robota, S – środek masy ramy robota, w, l – wymiary charakterystyczne robota, β – kąt obrotu ramy względem osi x nieruchomego układu współrzędnych, γ – kąt między kierunkiem wektora prędkości chwilowej punktu S i osią x.

Ruch MRKM wymaga zaplanowania trajektorii ruchu na podstawie równań kinematyki [1] oraz jej realizacji z zastosowaniem sygnałów sterujących wygenerowanych np. w wyniku rozwiązania zadania odwrotnego dynamiki [2]. W warunkach rzeczywistych, gdzie występują zakłócenia, a model dynamiki robota może nie uwzględniać wszystkich zjawisk związanych z ruchem, zadanie generowania sygnałów sterujących realizuje układ sterowania ruchem nadążnym.

2.1. Opis kinematyki robota

Jako przykład zostanie przeanalizowany ruch punktu S MRKM po torze o kształcie okręgu, gdzie współrzędne x_S i y_S zmieniają się zgodnie z układem równań

$$\begin{aligned} x_S - R_o \sin(\theta) &= 0, \\ y_S - R_o(1 - \cos(\theta)) &= 0, \end{aligned} \quad (1)$$

gdzie: R_o – promień okręgu, θ – kąt między kierunkiem wektora wodzącego punktu S zaczepionego w środku okręgu i osią x. Różniczkując układ równań (1), a następnie dodając stronami otrzymamy zależność (2) określającą wartość wektora prędkości chwilowej punktu S $\vec{v}_S$ w formie:

$$\dot{x}_S^2 + \dot{y}_S^2 = \dot{\theta}^2 R_o^2, \quad (2)$$

gdzie: $\dot{x}_S, \dot{y}_S$ – rzuty wektora prędkości $\vec{v}_S$ na poszczególne osie nieruchomego układu współrzędnych.

Jednoznaczne ustawienie w czasie i przestrzeni modelu MRKM wymaga znajomości wymiarów charakterystycznych obiektu oraz podania następujących współrzędnych: $x_S, y_S, \beta, \gamma, \theta, R_o, \varphi_1, \varphi_2, \varphi_3, \varphi_4$, gdzie $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ – kąty obrotu poszczególnych kół MRKM. Na pochodne tych współrzędnych narzucono więzy, których niezależne równania przyjmują formę

$$\begin{aligned} \dot{x}_S &= v_S \cos(\gamma), \\ v_S &= \sqrt{\dot{x}_S^2 + \dot{y}_S^2} = \dot{\theta} R_o, \\ \dot{x}_S [\cos(\beta - \alpha)] + \dot{y}_S [\sin(\beta - \alpha)] - \dot{\beta} [w \cos(\alpha) + l \sin(\alpha)] &= \dot{\varphi}_1 (R + r) \cos(\alpha), \\ \dot{x}_S [\cos(\beta + \alpha)] + \dot{y}_S [\sin(\beta + \alpha)] + \dot{\beta} [w \cos(\alpha) + l \sin(\alpha)] &= \dot{\varphi}_2 (R + r) \cos(\alpha), \\ \dot{x}_S [\cos(\beta + \alpha)] + \dot{y}_S [\sin(\beta + \alpha)] - \dot{\beta} [w \cos(\alpha) + l \sin(\alpha)] &= \dot{\varphi}_3 (R + r) \cos(\alpha), \\ \dot{x}_S [\cos(\beta - \alpha)] + \dot{y}_S [\sin(\beta - \alpha)] + \dot{\beta} [w \cos(\alpha) + l \sin(\alpha)] &= \dot{\varphi}_4 (R + r) \cos(\alpha), \\ \dot{\gamma} &= \dot{\theta}, \end{aligned} \quad (3)$$

gdzie: α – kąt między osią koła mecanum i osią rolki (parametr konstrukcyjny, $\alpha = \frac{\pi}{4}$), R – promień koła mecanum, r – promień rolki.

Układ równań (3) rozwiązano przy założeniu znajomości następujących parametrów: $\beta(0) = 0, \dot{\beta}(t) = 0, R_o = \text{const.}, v_S(t)$ – znany przebieg. Ruch MRKM podzielono na etapy: rozpędzanie, ruch ze stałą prędkością oraz hamowanie. Trapezowy przebieg wartości v_S aproksymowano profilem ciągłym z zastosowaniem funkcji

$$v_S = v_S^* \left(\frac{1}{1 + e^{-c(t-b_1)}} - \frac{1}{1 + e^{-c(t-b_2)}} \right), \quad (4)$$

gdzie: v_S^* – maksymalna wartość wektora prędkości zadanej punktu S, c – współczynnik wpływający na szybkość zmian wartości wektora prędkości punktu S w okresie rozpędzania i hamowania, b_1, b_2 – współczynniki średniego czasu rozpędzania i hamowania.

2.2. Opis dynamiki robota

Dynamiczne równania ruchu MRKM wyprowadzono korzystając z formalizmu Maggiego'go i zapisano w formie ogólnej [2]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) = u, \quad (5)$$

gdzie poszczególne macierze i wektory:

$$M(q) = \begin{bmatrix} a_1 \sin(\beta) + a_2 \cos(\beta - \alpha) & -a_1 \cos(\beta) + a_2 \sin(\beta - \alpha) & -a_3 \\ a_2 \cos(\beta + \alpha) - a_8 \sin(\beta - \alpha) & a_2 \sin(\beta + \alpha) + a_8 \cos(\beta - \alpha) & -a_4 \\ a_1 \cos(\beta) + a_2 \cos(\beta - \alpha) & a_1 \sin(\beta) + a_2 \sin(\beta - \alpha) & a_3 \end{bmatrix}, \quad (6)$$

$$C(q, \dot{q}) = \begin{bmatrix} -a_2 \sin(\beta - \alpha) \dot{\beta} & a_2 \cos(\beta - \alpha) \dot{\beta} & 0 \\ -a_2 \sin(\beta + \alpha) \dot{\beta} & a_2 \cos(\beta + \alpha) \dot{\beta} & 0 \\ -a_2 \sin(\beta - \alpha) \dot{\beta} & a_2 \cos(\beta - \alpha) \dot{\beta} & 0 \end{bmatrix}, \quad (7)$$

$$F(\dot{q}) = \begin{bmatrix} a_5 \operatorname{sgn}(\dot{\phi}_1) \\ a_6 \operatorname{sgn}(\dot{\phi}_3) \\ a_7 \operatorname{sgn}(\dot{\phi}_4) \end{bmatrix}, \quad q = \begin{bmatrix} x_S \\ y_S \\ \beta \end{bmatrix}, \quad u = \begin{bmatrix} M_1 \\ M_3 \\ M_4 \end{bmatrix}, \quad (8)$$

oraz: q – wektor współrzędnych uogólnionych, a_1 – a_8 – parametry modelu, u – wektor sygnałów sterujących, M_1, M_3, M_4 – momenty napędowe poszczególnych kół MRKM. W zależności (8) występują wartości wektorów prędkości kątowych kół $\dot{\phi}_1, \dot{\phi}_3, \dot{\phi}_4$, które wyznaczono z układu równań (3). Przedstawiony opis dynamiki MRKM umożliwia wyznaczenie trzech sygnałów sterujących, natomiast robot wyposażono w cztery układy wykonawcze. Brakujący sygnał sterujący M_2 wyznaczono na podstawie bilansu mocy [2] zgodnie z zależnością

$$M_2 = M_4 + M_3 - M_1. \quad (9)$$

Znając przebiegi wartości wektorów momentów napędowych oraz prędkości kątowych poszczególnych kół można wyznaczyć moc chwilową poszczególnych układów wykonawczych zgodnie z zależnością

$$P_i = \vec{M}_i \cdot \vec{\dot{\phi}}_i, \quad (10)$$

gdzie: P_i – moc chwilowa i -tego zespołu napędowego, $i = 1, \dots, 4$. Przebiegi wartości mocy chwilowych poszczególnych układów wykonawczych scałkowano, a po dodaniu wyników otrzymano pracę całkowitą L układu momentów napędowych powodujących ruch MRKM.

3. Wyniki badań

W niniejszym rozdziale przedstawiono wybrane wyniki przeprowadzonych testów numerycznych oraz badań weryfikacyjnych ruchu MRKM po zadanym torze ruchu z określoną orientacją ramy robota. Zrealizowano pomiary dla dwóch torów ruchu środka masy robota - po linii prostej i po łuku (tor ruchu w kształcie okręgu), które to są typowymi torami ruchu występującymi w aplikacjach przemysłowych MRKM.

3.1. Wyniki badań symulacyjnych

W przypadku trajektorii z torem ruchu o kształcie prostej przyjęto następujące parametry profilu prędkości (4): $v_S^* = 0.3 \frac{m}{s}$, $c = 5 \frac{1}{s}$, $b_1 = 3 s$, $b_2 = 12 s$, parametry toru ruchu: $x_S(0) = 0 m$, $y_S(0) = 1.0 m$, kąty β i γ dobrano zgodnie z tab. 1. We wszystkich symulacjach przyjęto następujące parametry modelu dynamiki MRKM [3]: $a_1 = 0.18$, $a_2 = 0.16$, $a_3 = 0.04$, $a_4 = 0.21$, $a_5 = 9.0$, $a_6 = 9.0$, $a_7 = 9.0$, $a_8 = 0.23$. Pracę całkowitą układu sił napędzających MRKM (momentów sił) L wyznaczono zgodnie z procedurą opisaną w podrozdziale 2.2, natomiast sumaryczny kąt obrotu kół ϕ_{Sk} wyznaczono zgodnie z zależnością:

$$\phi_{Sk} = \sum_{i=1}^{n=4} \int_{t_0}^{t_k} |\dot{\phi}_i(t)| dt, \quad (11)$$

Tab. 1. Porównanie wybranych parametrów ruchu MRKM po torze o kształcie prostej – testy numeryczne, $\gamma(0) = 0$, $\gamma = \text{const.}$

β [rad]	L [Nm]	φ_{Sk} [rad]
$\beta(0) = 0, \beta = \text{const.}$	466,6	106,4
$\beta(0) = \frac{\pi}{4}, \beta = \text{const.}$	429,7	75,2
$\beta(0) = \frac{\pi}{2}, \beta = \text{const.}$	466,6	106,4

Tab. 2. Porównanie wybranych parametrów ruchu MRKM po torze o kształcie okręgu – testy numeryczne, $\gamma(0) = 0$, $\gamma \in < 0; 2\Pi >$

β [rad]	L [Nm]	φ_{Sk} [rad]
$\beta(0) = 0, \beta = \text{const.}$	1064,0	223,0
$\beta(0) = \frac{\pi}{4}, \beta = \text{const.}$	1069,2	223,0
$\beta(0) = \frac{\pi}{2}, \beta = \text{const.}$	1069,2	223,0
$\beta(0) = 0, \beta \in < 0; 2\Pi >$	1325,5	247,7
$\beta(0) = \frac{\pi}{4}, \beta \in < \frac{\pi}{4}; 2\Pi + \frac{\pi}{4} >$	1196,1	245,4
$\beta(0) = \frac{\pi}{2}, \beta \in < \frac{\pi}{2}; 2\Pi + \frac{\pi}{2} >$	1326,5	247,7

gdzie: $t_0 = 0$ s – czas rozpoczęcia rejestracji sygnałów, t_k – czas zakończenia rejestracji sygnałów. Realizacja wygenerowanej trajektorii ruchu powoduje przemieszczenie punktu S robota z położenia początkowego $S_0(0,0, 1,0)$ do położenia końcowego $S_k(2,7, 1,0)$, przebyta droga to $s = 2,7$ m.

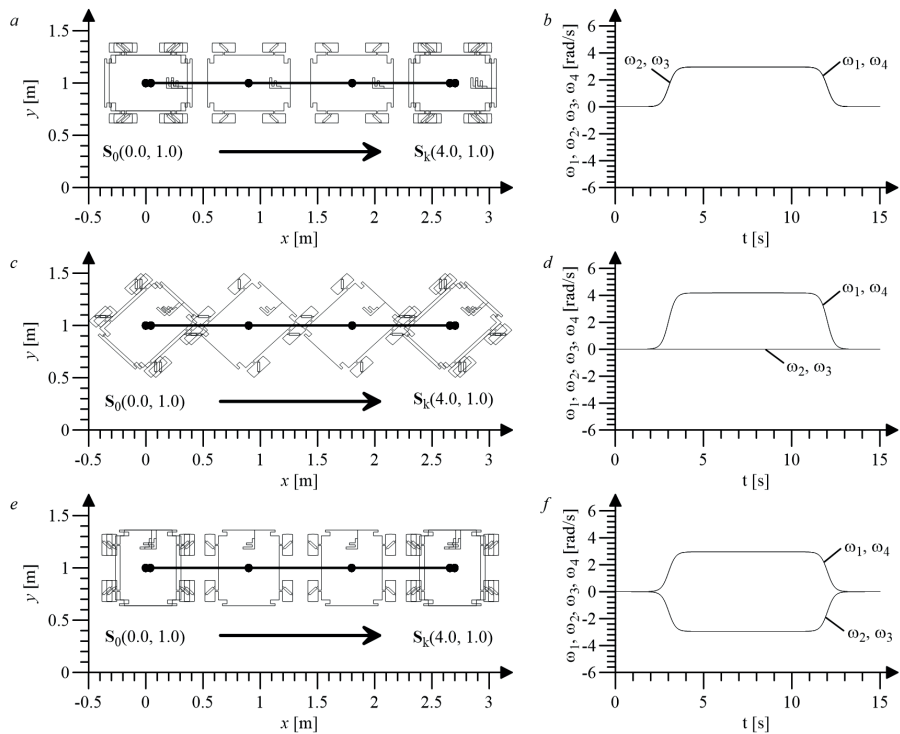
Analizując wyniki symulacji ruchu po torze o kształcie prostej można zauważyć, że w przypadku konfiguracji z kątem początkowym $\beta(0) = \frac{\pi}{4}$ stałym w trakcie ruchu praca L jest mniejsza niż w pozostałych przypadkach, sumaryczny kąt obrotu wszystkich kół φ_{Sk} również jest mniejszy. Wynika to ze specyfiki budowy kół mecanum. We wspomnianej konfiguracji w trakcie ruchu koła 2 i 3 nie obracają się, jedynie występuje swobodne toczenie się rolek mających kontakt z podłożem. W pozostałych analizowanych konfiguracjach otrzymano takie same wyniki.

Na rys. 2a, c, e przedstawiono tory ruchu punktu S MRKM z graficzną interpretacją położenia i orientacji robota dla różnych kątów początkowych orientacji ramy (literą F oznaczono przód robota), kolejno: $\beta(0) = 0$, $\beta(0) = \frac{\pi}{4}$, $\beta(0) = \frac{\pi}{2}$ rad. Przyjęto stałą wartość kąta β w trakcie ruchu. Na rys. 2b, d, f przedstawiono przebiegi wartości wektorów prędkości kątowych kół.

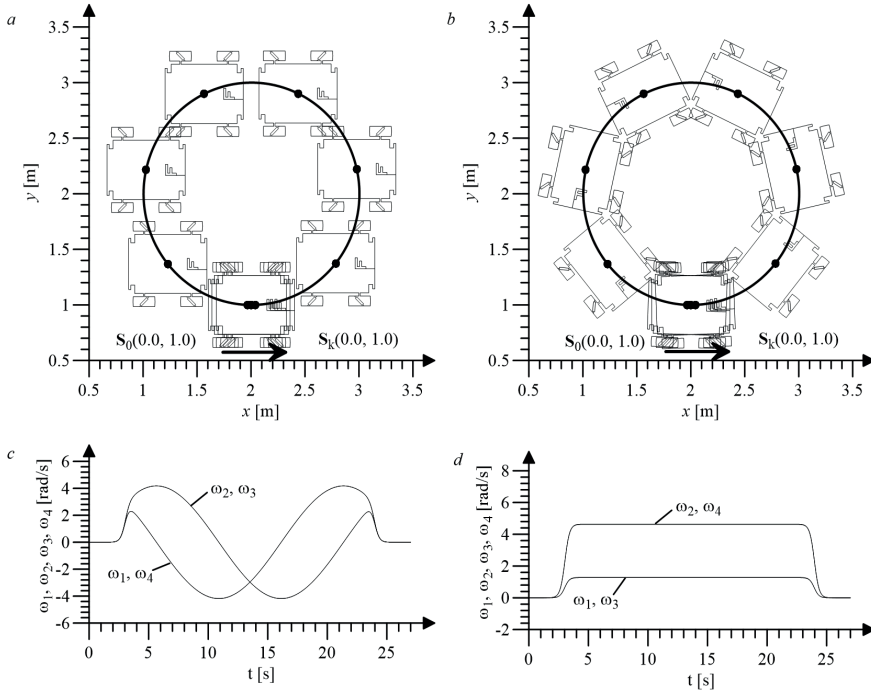
W przypadku trajektorii z torem ruchu o kształcie okręgu dobrano następujące parametry profilu prędkości (4): $v_S^* = 0,3 \frac{m}{s}$, $c = 5 \frac{1}{s}$, $b_1 = 3$ s, $b_2 = 23,95$ s, parametry toru ruchu: $x_S(0) = 2,0$ m, $y_S(0) = 1,0$ m, $R_o = 1,0$ m, kąty β i γ dobrano zgodnie z tab. 2. Realizacja wygenerowanej trajektorii ruchu powoduje przemieszczenie punktu S robota z położenia początkowego $S_0(2,0, 1,0)$ do położenia końcowego $S_k(2,0, 1,0)$, przebyta droga to $s = 6,28$ m.

Analizując wyniki przeprowadzonych badań symulacyjnych można zauważyć, że w przypadku ruchu po okręgu, gdy $\beta = \text{const.}$, początkowa orientacja ramy robota nie ma wpływu na sumaryczne kąty obrotu kół czy pracę układu sił. W przypadku, gdy kąt obrotu ramy $\beta \in < 0; 2\Pi >$, sumaryczne kąty obrotu kół są zbliżone, natomiast praca L w przypadku orientacji początkowej $\beta(0) = \frac{\pi}{4}$ jest mniejsza.

Tor ruchu punktu S MRKM z graficzną interpretacją położenia i orientacji robota



Rys. 2. Tor ruchu punktu S ramy robota i przebiegi wartości wektorów prędkości kątowych kół, gdy: a, b – $\beta(0) = 0, \beta = \text{const.}$, c, d – $\beta(0) = \frac{\pi}{4}, \beta = \text{const.}$, e, f – $\beta(0) = \frac{\pi}{2}, \beta = \text{const.}$



Rys. 3. Tor ruchu punktu S ramy robota i przebiegi wartości wektorów prędkości kątowych kół, gdy: a, c – $\beta(0) = 0$, $\beta = \text{const.}$, b, d – $\beta(0) = 0$, $\beta \in < 0; 2\pi >$

w przypadku założenia $\beta(0) = 0$ przedstawiono na rys. 3a, oraz rys. 3b, przyjmując odpowiednio stałą wartość kąta β w trakcie ruchu oraz zmienną wartość kąta $\beta \in < 0; 2\pi >$. Na rys. 3c i rys. 3d przedstawiono adekwatne przebiegi wartości wektorów prędkości kątowych poszczególnych kół.

3.2. Wyniki badań weryfikacyjnych

Badania weryfikacyjne zrealizowano na stanowisku badawczym wyposażonym w PC, kartę kontrolno-pomiarową dSpace DS1103 oraz MRKM. Robot Husarion Panther o masie $m_R = 55 \text{ kg}$ i maksymalny udźwigu $m_L = 80 \text{ kg}$ wyposażono w cztery zespoły napędowe złożone z silników BLDC 80PMB800K.80RBL (o mocy znamionowej $P = 473 \text{ W}$ każdy), przekładni oraz enkoderów inkrementalnych. Wymiary robota to $0.724 \times 0.766 \times 0.273 \text{ m}$, zastosowano koła mecanum o promieniu $R = 0.09 \text{ m}$ oraz promieniu rolek $r = 0.015 \text{ m}$. Sygnały sterujące układami napędowymi generowano w czasie rzeczywistym z zastosowaniem karty kontrolno-pomiarowej dSpace DS1103. Zrezygnowano z zastosowania przebiegów sygnałów sterujących będących wynikiem rozwiązania zadania odwrotnego dynamiki ze względu na zauważone rozbieżności między założonymi i zrealizowanymi parametrami ruchu, wynikającymi z występujących niedokładności modelu. Z tego powodu we wszystkich eksperymentach zastosowano układ sterowania zbudowany z regulatora PD z takimi samymi nastawami $K_D = 1$, $K_P = 5$.

Tab. 3. Porównanie wybranych parametrów ruchu MRKM po torze w kształcie prostej – badania weryfikacyjne, $\gamma(0) = 0$, $\gamma = \text{const}$.

β [rad]	L [Nm]	φ_{sk} [rad]
$\beta(0) = 0$, $\beta = \text{const}$.	449,7	106,3
$\beta(0) = \frac{\pi}{4}$, $\beta = \text{const}$.	461,5	75,2
$\beta(0) = \frac{\pi}{2}$, $\beta = \text{const}$.	511,2	105,8

Tab. 4. Porównanie wybranych parametrów ruchu MRKM po torze w kształcie okręgu – badania weryfikacyjne, $\gamma(0) = 0$, $\gamma \in < 0; 2\pi >$

β [rad]	L [Nm]	φ_{sk} [rad]
$\beta(0) = 0$, $\beta = \text{const}$.	1203,3	218,2
$\beta(0) = \frac{\pi}{4}$, $\beta = \text{const}$.	1216,0	219,4
$\beta(0) = \frac{\pi}{2}$, $\beta = \text{const}$.	1208,3	217,8
$\beta(0) = 0$, $\beta \in < 0; 2\pi >$	1507,1	247,6
$\beta(0) = \frac{\pi}{4}$, $\beta \in < \frac{\pi}{4}; 2\pi + \frac{\pi}{4} >$	1600,2	245,1
$\beta(0) = \frac{\pi}{2}$, $\beta \in < \frac{\pi}{2}; 2\pi + \frac{\pi}{2} >$	1674,2	247,0

W tab. 3 przedstawiono wartości wskaźników wyznaczonych w trakcie realizacji badań weryfikacyjnych ruchu MRKM po torze prostoliniowym, stosując analogiczne parametry jak w badaniach symulacyjnych.

Analiza porównawcza wyników badań weryfikacyjnych i testów numerycznych wskazuje na istotne rozbieżności świadczące o niedokładności modelu obliczeniowego dynamiki MRKM. Praca układu sił L w przypadku ruchu po torze prostoliniowym ze stałą wartością kąta $\beta = \frac{\pi}{2}$ jest większa od pracy L w przypadku kąta $\beta = \frac{\pi}{4}$ – podobnie jak w testach numerycznych, ale nie jest zbliżona do pracy L w przypadku kąta $\beta = 0$. Jednocześnie praca L w przypadku kąta $\beta = \frac{\pi}{4}$ jest większa niż w przypadku kąta $\beta = 0$, co jest istotnym odstępstwem od wyników badań symulacyjnych.

W tab. 4 przedstawiono wartości wskaźników wyznaczonych w trakcie zrealizowanych badań weryfikacyjnych ruchu MRKM po torze o kształcie okręgu.

4. Podsumowanie

Analiza wyników przeprowadzonych testów numerycznych i badań weryfikacyjnych umożliwia sformułowanie następujących wniosków. Ruch postępowy ramy MRKM wymaga wykonania mniejszej pracy momentów napędowych, niż ruch płaski, więc jeżeli realizowane zadanie nie wymaga zmiany orientacji ramy lub osiągnięcia zadanego położenia końcowego z określoną orientacją, biorąc pod uwagę kryterium minimalizacji pracy momentów napędowych, zalecany jest brak zmiany orientacji ramy. W przypadku zmiany orientacji ramy robota w czasie ruchu wykryto rozbieżność między wynikami testów numerycznych i badań weryfikacyjnych. W testach numerycznych najmniejszą pracę wyznaczono w przypadku początkowej orientacji ramy $\beta(0) = \frac{\pi}{4}$, natomiast w przypadku badań weryfikacyjnych w przypadku $\beta(0) = 0$. To samo zjawisko zaobserwowano stosując trajektorię o torze prostoliniowym i różne stałe orientacje ramy robota w trakcie ruchu. Efekt ten może wynikać z braku ujęcia w modelu dynamiki robota zjawiska oporów toczenia rolek swobodnych kół meca-

num (przy określonej orientacji ramy w trakcie ruchu wybrane koła mecanum mogą być nieruchome). Badania weryfikacyjne wskazują również na brak symetrii wartości wektorów momentów oporów ruchu kół MRKM w zakresie obrotu w prawo i w lewo, co jest powodem konieczności wykonania większej pracy w przypadku, gdy koła obracają się w kierunku odpowiadającym ruchowi postępowemu ramy robota w tył. Taka sytuacja występuje przykładowo w trakcie ruchu MRKM po torze o kształcie linii prostej ze stałą orientacją ramy i początkową wartością kąta $\beta(0) = \frac{\pi}{2}$, wartości wektorów momentów M_2 i M_3 są ujemne. Zastosowany model dynamiki MRKM jest niewystarczający do opisu ruchu robota w przypadku zastosowania złożonych trajektorii w zakresie doboru parametrów modelu oraz braku uwzględnienia oporów ruchu toczących się rolek swobodnych kół mecanum dla wybranych orientacji ramy ($\beta(0) = \frac{\pi}{4}$), co będzie tematyką dalszych badań.

Literatura

- [1] Z. Hendzel. A Description of the Motion of a Mobile Robot with Mecanum Wheels – Kinematics. In: Automation 2019. Advances in Intelligent Systems and Computing. Red. R. Szewczyk, C. Zieliński, M. Kaliczyńska. Cham, Springer 2019. Vol. 920, s. 346–355.
- [2] Z. Hendzel. A Description of the Motion of a Mobile Robot with Mecanum Wheels – Dynamics. In: Automation 2019. Advances in Intelligent Systems and Computing. Red. R. Szewczyk, C. Zieliński, M. Kaliczyńska. Cham, Springer 2019. Vol. 920, s. 337–345.
- [3] M. Kołodziej. *Synteza ruchu mobilnego robota z kołami mecanum*. Rozprawa doktorska, Rzeszów, 2024.
- [4] L. Xie et al. Power-minimization and energy-reduction autonomous navigation of an omnidirectional Mecanum robot via the dynamic window approach local trajectory planning. IJARS, 2018, Vol. 15, s. 1–12.
- [5] L. Xou et al. Practical Model for Energy Consumption Analysis of Omnidirectional Mobile Robot. Sensors, 2021, Vol. 21, s. 1–18.
- [6] L. Zhang et al. Energy Modeling and Experimental Validation of Four-Wheel Mecanum Mobile Robots for Energy-Optimal Motion Control. Symmetry, 2019, Vol. 11, s. 1–14.

Work of the force system performed during the movement of a mobile robot with mecanum wheels

Mobile robots with mecanum wheels have a significant advantage in the field of motion planning and execution compared to structures equipped with standard wheels, e.g. pneumatic wheels. The ability to move in any direction with any orientation of the robot frame is particularly attractive when moving in an environment where there is limited space for manoeuvring, such as narrow traffic lanes. Such a wide range of movement possibilities leads to the question: which orientation of the robot frame during movement is the most advantageous? This paper formulates the answer to this question, taking the work done by the force system during the robot's motion as the criterion. Based on numerical tests and verification studies for different trajectories and initial orientations of the robot frame, the work of the force system has been determined and compared. The studies carried out allowed conclusions to be drawn regarding the preferred orientation of the robot frame with mecanum wheels during movement.

Zastosowanie algorytmu SLAM w realizacji zadania typu "podążaj do celu"

Marcin Szuster¹, Mateusz Szeremeta¹, Paweł Penar¹

Streszczenie

Pomiar położenia i orientacji ramy robota mobilnego w nieznanym środowisku jest ważnym zagadnieniem w planowaniu jego autonomicznego ruchu. W artykule przedstawiono porównanie realizacji zadania typu "podążaj do celu" z zastosowaniem metody wyznaczania położenia i orientacji robota na podstawie pomiaru kątów obrotu kół napędzających oraz z zastosowaniem hybrydowej metody łączącej ciągle określanie położenia i orientacji ramy na bazie odometrii z okresową aktualizacją parametrów wyznaczaną z zastosowaniem algorytmu SLAM na bazie pomiarów z czujnika LIDAR. Sterowanie ruchem robota realizuje hierarchiczny układ sterowania z warstwą generowania trajektorii zbudowaną z zastosowaniem układów z logiką rozmytą oraz warstwą realizacji ruchu w formie regulatora PD. Przeprowadzone badania weryfikacyjne algorytmu z zastosowaniem robota mobilnego z kołami mecanum oraz systemu motion capture Vicon potwierdziły pozytywny wpływ uwzględnienia korekty lokalizacji robota na precyzję osiągnięcia zadanego położenia końcowego.

1. Wprowadzenie

Autonomiczne roboty mobilne znajdują coraz więcej zastosowań, m. in. w zadaniach transportowych. Realizacja tego typu zadań wiąże się z koniecznością wyposażenia układu sensorycznego robota w urządzenia umożliwiające skanowanie otoczenia w celu detekcji przeszkód oraz określania położenia i orientacji robota, np. czujniki typu LIDAR. Ze strony programowej związanej z przetwarzaniem danych pomiarowych ze środowiska powszechnie używanymi metodami do jednoczesnej lokalizacji robota oraz tworzenia mapy otoczenia są algorytmy SLAM (ang. Simultaneous Localization and Mapping) [1, 12], czy algorytmy realizujące wyłącznie zadania lokalizacji pojazdu w otoczeniu [13].

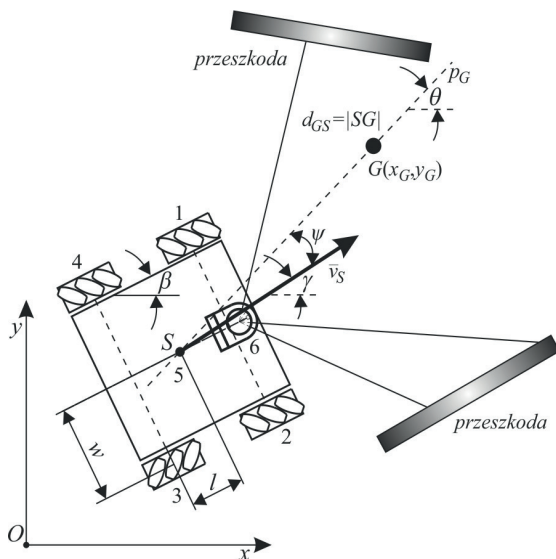
Interesującym zagadnieniem jest wpływ danych zgromadzonych przez różne układy sensoryczne na jakość lokalizacji. W pracy [10] porównano dokładność lokalizacji mobilnego robota na podstawie wyników uzyskanych przy pomocy: odometrii kołowej, inercyjnego urządzenia pomiarowego - IMU (ang. Inertial Measurement Unit), wizualnej odometrii oraz trilateracji sygnałów szerokopasmowych - UWB (ang. Ultra Wideband). W wyniku badań stwierdzono, że w środowisku uniemożliwiającym nawigację satelitarną, metoda UWB była jedyną metodą zapewniającą zadowalające efekty. Pozostałe metody, ze względu na występujący dryft pozycji, nie mogły zostać użyte w procesie lokalizacji. Metoda UWB nie umożliwia jednak wyznaczania orientacji robota w przestrzeni. Autorzy sugerują możliwość rozwiązania tego problemu z zastosowaniem fuzji danych z pozostałych czujników lub zastosowanie kilku znaczników UWB w celu aproksymacji orientacji robota.

¹ Politechnika Rzeszowska im. Ignacego Łukasiewicza, al. Powstańców Warszawy 12, 35-959 Rzeszów, mszuster@prz.edu.pl, m.szeremeta@prz.edu.pl, ppenar@prz.edu.pl, <https://prz.edu.pl/>

Fuzja sygnałów, stanowiąca połączenie danych pochodzących z różnych czujników, to metoda używana w celu poprawy jakości lokalizacji obiektów lub w celu wyeliminowania ograniczeń występujących przy zastosowaniu określonego rodzaju sensora. W pracy [4] połączono dane z sensora GPS oraz LIDAR. Fuzja danych została zrealizowana z zastosowaniem biblioteki *amcl* (ang. Autonomous Monte Carlo Localization) w środowisku ROS. Na podstawie otrzymanych wyników zauważono poprawę dokładności lokalizacji systemu nawigacji LIDAR + GPS w porównaniu z samym czujnikiem LIDAR. W pracy [9] zmierzono błąd lokalizacji z zastosowaniem odometrii kołowej robota z kołami mecanum. Zauważono, że błąd w kierunku poprzecznym do orientacji ramy robota jest większy od błędu w kierunku wzdłużnym. Następnie na podstawie danych z czujnika LIDAR wprowadzono współczynniki korekcji, wynikiem czego była zbieżność toru ruchu zrealizowanego z zastosowaniem pomiarów z odometrii do zrealizowanego z zastosowaniem czujnika LIDAR. W pracy [3] porównano błędy odwzorowywania zadanego toru ruchu na podstawie odometrii kołowej, odometrii wizualno-inercyjnej (składającej się z dwóch kamer oraz czujnika IMU), algorytmu LIO-SAM (ang. Lidar-Inertial Semantic SLAM, z układem sensorycznym składającym się z czujnika LIDAR i IMU) oraz fuzji odometrii kołowej i wizualno-inercyjnej zrealizowanej z zastosowaniem filtru Kalmana. Zastosowanie algorytmu LIO-SAM spowodowało redukcję błędów, ale działanie algorytmu było podatne na zakłócenia. Najniższy błąd uzyskano w przypadku połączenia odometrii kołowej z odometrią wizualno-inercyjną.

Poprawa jakości autonomicznego ruchu robota jest również rozwijana poprzez nowe rozwiązania dotyczące metod lokalizacji [6] i nawigacji [7]. W pracy [5] przedstawiono zmodyfikowany filtr cząsteczkowy oparty na lokalizacji Monte Carlo (MCL, ang. Monte Carlo Localization). Filtr poprzez odrzucenie części cząstek o najniższej wadze miał poprawić jakość lokalizacji, co było pierwszą hipotezą badawczą. Druga hipoteza dotyczyła zastosowania techniki dopasowania skanów 2D pomiędzy rzeczywistym skanem, a wirtualnym skanem przewidywanej lokalizacji robota (otrzymanej z MCL) w celu poprawy jakości estymacji pozycji. Ostatnia hipoteza dotyczyła wprowadzenia do MCL sprzężenia zwrotnego otrzymanej nowej estymaty pozycji, pochodzącej z techniki dopasowania skanów. W wyniku przeprowadzonych badań potwierdzono wszystkie hipotezy. Dla algorytmu MCL zauważono, że zarówno dla populacji cząstek, których wagi były większe od średniej wagi, jak i dla populacji 10% największych wag błąd pozycji jest mniejszy od błędu algorytmu MCL (gdzie wszystkie cząstki są brane pod uwagę). Jedynie metoda wyboru wyłącznie pojedynczej cząstki o największej wadze dała negatywne wyniki. We wszystkich testach technika dopasowania skanów zmniejszyła błąd pozycji.

Podstawowym zadaniem algorytmu SLAM jest wyznaczenie aktualnego położenia i orientacji robota. W niniejszym artykule tę funkcjonalność zastosowano w hybrydowym algorytmie wyznaczania położenia i orientacji ramy mobilnego robota z kołami mecanum (MRKM), widzianym jako element warstwy generowania trajektorii hierarchicznego układu sterowania ruchem robota w zadaniu typu "podążaj do celu" GS (ang. Goal Seeking). Ze względu na ograniczenia sprzętowe związane z przepustowością łącza komunikacyjnego, częstotliwością skanowania otoczenia przez czujnik LIDAR oraz mocą obliczeniową wymaganą do realizacji algorytmu SLAM, wyznaczanie położenia i orientacji ramy robota w czasie rzeczywistym na podstawie sygna-



Rys. 1. Schemat mobilnego robota z kołami mecanum

łów z otoczenia nie jest możliwe. Z tego powodu zaproponowano hybrydową metodę łączącą ciągły pomiar położenia i orientacji robota na podstawie odometrii kołowej z okresową korektą na bazie danych dostępnych z algorytmu SLAM.

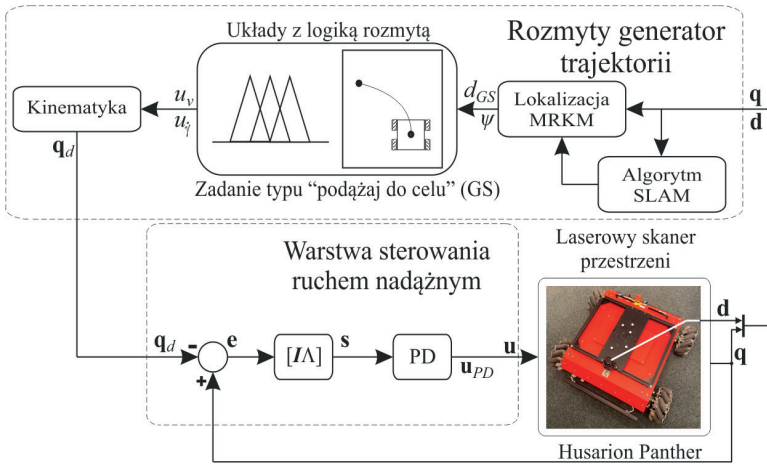
Artykuł składa się z pięciu części. Po krótkim wprowadzeniu w tematykę problemu omówiono budowę hierarchicznego układu sterowania ruchem MRKM w zadaniu typu GS. Następnie przedstawiono budowę stanowiska badawczego z uwzględnieniem krótkiej charakterystyki każdego z elementów systemu. W kolejnym rozdziale przedstawiono wyniki badań weryfikacyjnych ruchu MRKM do celu z zastosowaniem odometrii do wyznaczania położenia i orientacji robota oraz z uwzględnieniem zaproponowanego rozwiązania z korektą lokalizacji ramy robota na podstawie informacji z algorytmu SLAM. Artykuł kończy podsumowanie otrzymanych rezultatów.

2. Układ sterowania

Obiektem sterowania jest MRKM przedstawiony na rys. 1, gdzie 1,2,3,4 – koła mecanum, 5 – rama MRKM, 6 – czujnik typu LIDAR, β – kąt obrotu ramy względem osi x nieruchomego układu współrzędnych (przyjęto $\beta = 0$), d_{GS} – odległość p. S od celu – p. G, θ – kąt między prostą p_G przechodzącą przez punktu G i S, a osią x , γ – kąt między kierunkiem wektora prędkości chwilowej p. S, a osią x , w , l - wymiary charakterystyczne robota, kąt ψ określony zależnością

$$\psi = \gamma - \theta. \quad (1)$$

Sterowanie ruchem MRKM w zadaniu typu GS zrealizowano z zastosowaniem hierarchicznego układu sterowania, w którym złożony problem generowania i realizacji zadanej trajektorii ruchu dekomponowano i rozwiązano z zastosowaniem dwóch



Rys. 2. Schemat hierarchicznego układu sterowania

warstw. Warstwa nadrzędna realizuje zadanie generowania trajektorii ruchu robota z zastosowaniem układów z logiką rozmytą (ULR), na podstawie znajomości położenia celu oraz lokalizacji i orientacji ramy robota. Warstwa realizacji ruchu składa się z regulatora PD. Jej zadaniem jest realizacja trajektorii wygenerowanej przez warstwę nadrzędną. Schemat hierarchicznego układu sterowania przedstawiono na rys. 2.

Warstwę generowania trajektorii ruchu w zadaniu GS zrealizowano z zastosowaniem dwóch ULR [11], w których użyto modelu Mamdaniego z trójkątnymi i trapezowymi funkcjami przynależności do zbiorów rozmytych przesłanek oraz konkluzji. Pierwszy z ULR generuje sygnał u_v sterujący wartością wektora prędkości zadanej p. S na podstawie odległości do celu d_{GS} . Funkcje przynależności do zbiorów rozmytych dobrano w taki sposób, aby przy "dużej" i "średniej" odległości do celu robot poruszał się z maksymalną prędkością, dopiero gdy odległość jest "mała" zadana prędkość jest stopniowo redukowana. Drugi ULR generuje sygnał u_γ sterujący zadaną prędkością kątową $\dot{\gamma}$, co przekłada się na sterowanie kierunkiem wektora prędkości chwilowej p. S ramy robota w celu minimalizacji kąta ψ i powoduje ruch ramy robota w kierunku celu.

Nowym elementem wprowadzonym w algorytmie wyznaczania zadanej trajektorii ruchu jest sposób obliczania położenia p. S ramy robota oraz kąta β orientacji ramy. W poprzednich aplikacjach algorytmu współrzędne te były określone na podstawie pomiaru kątów obrotu kół MRKM oraz rozwiązania zadania prostego kinematyki. Taki sposób lokalizacji prowadził do znacznej rozbieżności między zrealizowanym przemieszczeniem p. S robota oraz wyznaczonym na podstawie pomiarów. Z tego względu rozbudowano układ sensoryczny robota o czujnik typu LIDAR, a układ sterowania o algorytm SLAM umożliwiający wyznaczanie współrzędnych p. S i orientacji robota na podstawie pomiaru odległości do przeszkód i budowy mapy otoczenia.

Algorytm SLAM zaimplementowano z zastosowaniem pakietu Navigation Toolbox w programie Matlab. Działanie algorytmu polega na konstruowaniu grafu pozycji, w którym wierzchołki grafu to estymaty pozycji robota wraz z lokalnymi po-

miarami (chmura punktów o współrzędnych wyznaczonych z zastosowaniem czujnika LIDAR). Krawędzie łączące wierzchołki reprezentują względne przemieszczenie pomiędzy kolejnymi pomiarami, które wyznaczane jest wyłącznie na podstawie danych z czujnika LIDAR z zastosowaniem algorytmu dopasowywania skanów [2]. Co iterację przeprowadzana jest optymalizacja grafu w celu znalezienia zamknięcia pętli, czyli nałożenia się części skanu na obszar wcześniej zmapowany i tym samym wprowadzenia dodatkowych krawędzi (więzów) pomiędzy wierzchołkami grafu. Następnie w celu zmniejszenia błędu pozycji cały graf jest optymalizowany przy użyciu algorytmu Trust-Region-Dogleg. Algorytm umożliwia również budowę mapy otoczenia, w której każda komórka zawiera wartość od 0 do 1, gdzie wartości bliskie 0 oznaczają wolną przestrzeń, a wartości bliskie 1 oznaczają wysokie prawdopodobieństwo wystąpienia przeszkody.

Ze względu na ograniczenia techniczne aktualizacja współrzędnych p. S i kąta β z zastosowaniem algorytmu SLAM nie jest możliwa z częstotliwością działania algorytmu sterowania $h = 100$ [Hz], tylko co $t_{SL} = 0.5$ [s]. Z tego powodu zaproponowano algorytm, który wyznacza współrzędne p. S i kąt β w trybie ciągłym na podstawie sygnałów z enkoderów zamontowanych w zespołach napędowych kół z cykliczną korektą parametrów na podstawie pomiarów z zastosowaniem czujnika LIDAR i algorytmu SLAM. Aby zapewnić ciągłość przebiegu trajektorii zadanej zastosowano algorytm korekty współrzędnych w formie zależności

$$x_S = x_{Se} + (x_{Sli} - x_{Se})(1 - e^{-\kappa(t-t_{ki})}), \quad (2)$$

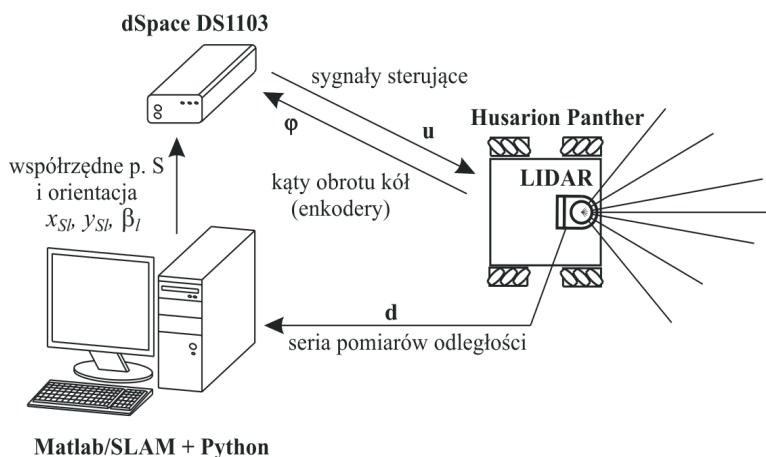
obowiązującej w okresie korekty dla $t \in [t_{ai}, t_{ai} + \Delta t]$, gdzie: x_{Se} - współrzędna p. S na kierunku osi x wyznaczona na podstawie pomiaru kątów obrotu kół z zastosowaniem enkoderów, x_{Sli} - współrzędna p. S na kierunku osi x wyznaczona na podstawie pomiarów z zastosowaniem czujnika LIDAR i algorytmu SLAM na początek i -tego okresu korekty, κ - współczynnik projektowy, t_{ki} - czas rozpoczęcia okresu wprowadzenia i -tej korekty parametrów na podstawie i -tego pomiaru z czujnika LIDAR. Po zakończeniu okresu korekty $x_S = x_{Sli}$, a kolejne zmiany współrzędnej są wyznaczane na podstawie sygnału z enkoderów aż do rozpoczęcia kolejnego okresu korekty. Analogicznie realizowana jest korekta współrzędnej y_S oraz kąta β . Na podstawie współrzędnych p. S robota x_S, y_S i współrzędnych celu $G(x_G, y_G)$ wyznaczana jest odległość d_{GS} oraz kąt θ , który umożliwia obliczenie kąta ψ zgodnie z zależnością (1). Wielkości te po normalizacji stanowią wejścia do ULR.

W module "Kinematyka" sygnały sterowania warstwy generowania trajektorii ruchu są przeliczane na zadane prędkości zgodnie z zależnością (3)

$$\begin{bmatrix} \dot{x}_{Sd} \\ \dot{y}_{Sd} \\ \dot{\gamma}_d \end{bmatrix} = \begin{bmatrix} v_{Smax} \cos(\gamma) & 0 \\ v_{Smax} \sin(\gamma) & 0 \\ 0 & \dot{\gamma}_{max} \end{bmatrix} \begin{bmatrix} u_v \\ u_{\dot{\gamma}} \end{bmatrix}, \quad (3)$$

gdzie: $\dot{x}_{Sd}, \dot{y}_{Sd}$ - rzuty wektora zadanej prędkości chwilowej p. S na osie nieruchomego układu współrzędnych, v_{Smax} - maksymalna wartość wektora prędkości p. S, $\dot{\gamma}_{max}$ - maksymalna wartość wektora prędkości kątowej $\dot{\gamma}$.

Zadana trajektoria jest realizowana z zastosowaniem sygnałów sterujących warstwy realizacji ruchu zbudowanej z regulatora PD.



Rys. 3. Schemat stanowiska badawczego

3. Stanowisko badawcze

Stanowisko badawcze pokazano na rys. 3. Składa się ono z: czterokołowego MRKM Husarion Panther, wyposażonego w czujnik LIDAR, systemu Motion Capture (MC) firmy Vicon, karty kontrolno-pomiarowej dSpace DS1103 oraz komputera osobistego.

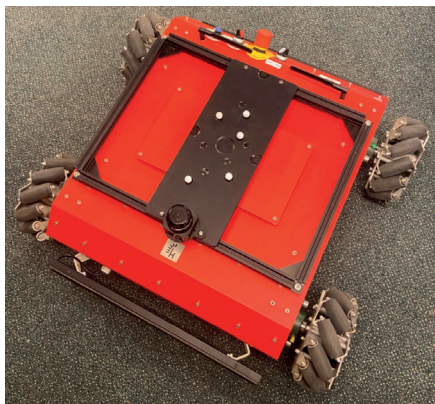
Robot Husarion Panther o wymiarach $0.724 \times 0.766 \times 0.273$ [m] z kołami mecanum o promieniu $R=0.09$ [m] oraz promieniu rolek $r=0.015$ [m], przedstawiono na rys. 4. Masa robota wynosi $m_R = 55$ [kg], a maksymalny udźwig $m_L = 80$ [kg]. Pojazd wyposażono w cztery silniki BLDC 80PMB800K.80RBL (o mocy znamionowej $P = 473$ [W] każdy) z przekładniami planetarnymi oraz enkoderami inkrementalnymi.

Czujnik LIDAR RPLIDAR S1 zamontowano na profilach znajdujących się w przedniej części ramy robota. Skaner 2D wykonuje obrót o kąt 360° z częstotliwością $f_S = 10$ [Hz]. Zasięg czujnika wynosi $d_L = 40$ [m] (dla obiektów czarnych $d_{LB} = 10$ [m]), rozdzielczość kątowna $\phi_L = 0,391$ [°], a maksymalny błąd pomiaru $d_{Le} = 0.01$ [m].

Zastosowany w badaniach system MC firmy Vicon umożliwia pomiar pozycji i orientacji obiektu w oparciu o zestaw kamer śledzących położenie znaczników umieszczonych na obiekcie. System wyposażono w 10 kamer Vicon Vero v2.2. Kamery o rozdzielczości 2.2 [MP] (2048×1088) i częstotliwości odświeżania obrazu $f_c = 330$ [Hz] wyposażono w obiektywy zmienneoogniskowe Vicon 6-12 [mm]. Sygnały z kamer trafiają do komputera. Za pomocą programu Tracker 3.9 możliwe jest śledzenie pozycji obiektu z dokładnością do dziesiątych części milimetra w czasie rzeczywistym.

Sygnały sterujące układami napędowymi wygenerowano w czasie rzeczywistym z zastosowaniem karty kontrolno-pomiarowej dSPACE ds1103, która umożliwia również akwizycję danych pomiarowych i realizację algorytmu sterowania zaprogramowanego w środowisku Matlab/Simulink.

Realizacja algorytmu sterowania wymaga akwizycji danych z czujnika LIDAR oraz ich transferu do karty kontrolno-pomiarowej dSPACE. Czujnik LIDAR, karta kontrolno-pomiarowa oraz robot mobilny mogą być traktowane jako systemy cyber-



Rys. 4. Robot Husarion Panther

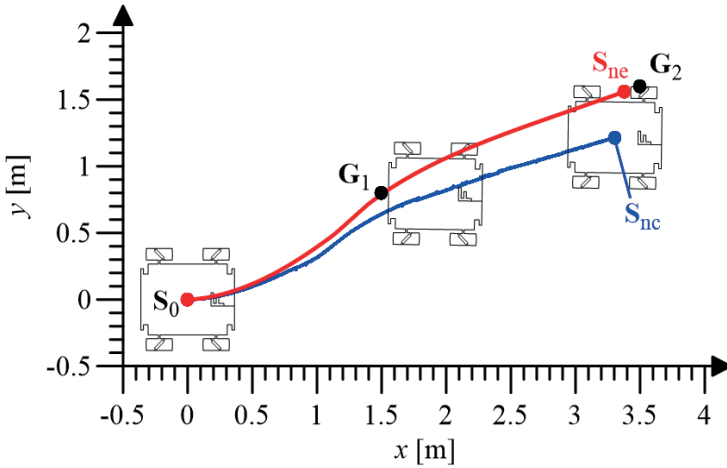
fizyczne (CPS, ang. Cyber-Physical Systems), które są niekompatybilne programowo – sprzętowo. W celu rozwiązania tego problemu i zapewnienia komunikacji pomiędzy systemami CPS zaprojektowano modułową aplikację: RoboDataLink.py [8]. Aplikacja umożliwia niezależną czasowo obsługę danych wejściowych pochodzących z systemu MC Vicon oraz LIDAR RPLIDAR S1 i danych wyjściowych przesyłanych do karty dSPACE DS1103.

4. Wyniki badań weryfikacyjnych

Przeprowadzono badania weryfikacyjne zaproponowanego hierarchicznego algorytmu sterowania w dwóch wersjach: z zastosowaniem metody wyznaczania położenia p. S i orientacji ramy MRKM na podstawie pomiaru kątów obrotu kół (punkt 4.1) oraz z zastosowaniem zaproponowanej metody, gdzie pomiary bazujące na odometrii były aktualizowane co $t_{akt} = 0.5$ [s] zgodnie z algorytmem przedstawionym w rozdziale 2 na podstawie sygnałów z czujnika LIDAR i metody SLAM (punkt 4.2). W ramach obydwu testów zastosowano takie same nastawy algorytmów sterowania poszczególnych warstw hierarchicznego układu sterowania, te same warunki początkowe ($x_S(0) = 0$, $y_S(0) = 0$ [m], $\beta(t) = 0$ [rad]), przyjęto współrzędne celu $G_1(1.5, 0.8)$, $G_2(3.5, 1.6)$ oraz promień $r_G = 0.15$ [m] wokół celu. Przyjęto, że osiągnięcie odległości $d_{GS} \leq r_G$ oznacza zakończenie realizacji zadania typu GS z sukcesem i prędkość v_S jest redukowana do $v_S = 0[\frac{m}{s}]$ zgodnie z zadaną procedurą. W prezentowanych eksperymentach po osiągnięciu celu G_1 i zatrzymaniu robota na czas $t = 5$ [s] wprowadzane są do systemu nowe współrzędne celu G_2 i ruch jest kontynuowany. W badaniach weryfikacyjnych przyjęto $v_{Smax} = 0.3[\frac{m}{s}]$ oraz $\dot{\gamma}_{max} = 0.9[\frac{rad}{s}]$.

4.1. Pomiar z zastosowaniem odometrii

Wyniki pierwszego testu przedstawiono na rys. 5, gdzie: S_0 – położenia p. S dla czasu $t = 0$ [s], $S_{ne}(x_{Se}, y_{Se})$ – położenie p. S po zakończeniu ruchu wyznaczone na podstawie sygnału z enkoderów, $S_{nc}(x_{Sc}, y_{Sc})$ – położenie p. S po zakończeniu ruchu zweryfikowane na podstawie sygnału z systemu MC.



Rys. 5. Porównanie toru ruchu punktu S wyznaczonego na podstawie sygnałów z enkoderów i z zastosowaniem systemu Motion Capture

Analizując przebiegi przedstawione na wykresach można wnioskować o poprawności realizacji algorytmu sterowania typu GS, bazującego na położeniu i orientacji ramy MRKM wyznaczonych na podstawie pomiarów kątów obrotu kół z zastosowaniem enkoderów ($d_{GSne} = 0.132$ [m] - odległość p. S do celu G_2 po zakończeniu ruchu, wyznaczona na podstawie pomiaru z enkoderów). Jednakże pomiar z zastosowaniem systemu MC wykazał znaczną różnicę między wyznaczonym numerycznie i zrealizowanym torem ruchu, gdzie $d_{GSnc} = 0.431$ [m] - odległość p. S do celu G_2 po zakończeniu ruchu, wyznaczona z zastosowaniem systemu MC. Odległość między punktami S_{ne} i S_{nc} po zakończeniu ruchu wynosi $d_{Snc} = 0.349$ [m] i można wnioskować, że przy większej liczbie etapów ruchu mogłaby wzrosnąć.

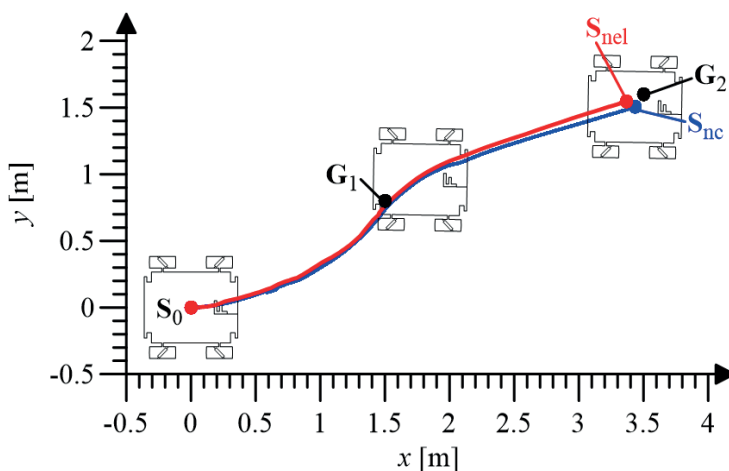
4.2. Pomiar z zastosowaniem algorytmu SLAM

Wyniki drugiego testu przedstawiono na rys. 6, gdzie: $S_{nel}(x_{Sel}, y_{Sel})$ – położenie p. S po zakończeniu ruchu wyznaczone na podstawie sygnału z enkoderów z cykliczną aktualizacją na podstawie informacji z czujnika LIDAR i algorytmu SLAM. Zarówno $d_{GSnel} = 0.137$ [m] (odległość p. S od celu G_2 po zakończeniu ruchu, wyznaczona na podstawie pomiaru z enkoderów z uwzględnieniem cyklicznych korekt na podstawie sygnałów z algorytmu SLAM), jak i $d_{GSnc} = 0.117$ [m] mieszczą się w zakładanym przedziale poniżej 0.15 [m].

Uwzględnienie w algorytmie wyznaczania położenia p. S i orientacji ramy MRKM cyklicznej korekty na podstawie sygnałów z algorytmu SLAM umożliwiło poprawną realizację zadania typu GS.

5. Podsumowanie

W artykule przedstawiono porównanie realizacji zadania typu "GS" przez MRKM z zastosowaniem dwóch metod wyznaczania położenia i orientacji ramy robota: odo-



Rys. 6. Porównanie toru ruchu punktu S wyznaczonego na podstawie sygnałów z enkoderów i algorytmu SLAM, oraz z zastosowaniem systemu Motion Capture

metrii kołowej oraz zaproponowanego algorytmu łączącego okresową aktualizację lokalizacji robota z zastosowaniem algorytmu SLAM w oparciu o dane z czujnika LIDAR z ciągłym pomiarem z zastosowaniem enkoderów inkrementalnych.

Zauważono znaczną redukcję błędu osiągnięcia zadanego położenia końcowego w eksperymencie, w którym zastosowano zaproponowane rozwiązanie ($d_{GSnc} = 0.117$ [m]), w porównaniu z użyciem jedynie sygnałów z enkoderów ($d_{GSnc} = 0.431$ [m]). Przedstawiony algorytm umożliwił redukcję problemu błędów odwzorowania zadanego toru ruchu wybranego punktu ramy robota. Błędy spowodowane są dwoma czynnikami: po pierwsze wyznaczaniem położenia i orientacji ramy MRKM z zastosowaniem odometrii przy możliwym poślizgu w punkcie styku rolki z podłożem. Po drugie nierównościami podłoża - gdy jedno z kół robota o sztywnej ramie i braku podatności układu zawieszenia kół traciło kontakt z podłożem, błędy odwzorowania zadanego toru ruchu i orientacji ramy robota zwiększały się.

Kierunki dalszych badań będą obejmowały opracowanie algorytmu realizacji zadania typu GS z osiągnięciem zadanego położenia końcowego p. S MRKM i określonej orientacji ramy robota (β_n - zdefiniowany zadany kąt obrotu ramy w położeniu końcowym).

Literatura

- [1] I. Belkin et. al. Real-time lidar-based localization of mobile ground robot. *Procedia Comput. Sci.*, 2021, Vol. 186, s. 440–448.
- [2] P. Biber et. al. The normal distributions transform: a new approach to laser scan matching. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. Proceedings*. Las Vegas, USA, October 27-31, 2003. s. 2743-2748
- [3] L. Chaehyun et. al. Insufficient environmental information indoor localization

- of mecamum mobile platform using wheel-visual-inertial odometry. JMST, 2024, Vol. 38, s. 5007–5015.
- [4] D. Chikurtev et. al. Mobile robot localization and navigation using LIDAR and indoor GPS. IFAC-PapersOnLine, 2021, Vol. 54, No. 13, s. 351–356.
 - [5] A. Filotheou et. al. Pose selection and feedback methods in tandem combinations of particle filters with scan-matching for 2D mobile robot localisation. JINT, 2020, Vol. 100, s. 925–944.
 - [6] X. Li et. al. Integrate point-cloud segmentation with 3D LiDAR scan-matching for mobile robot localization and mapping. Sensors, 2020, Vol. 20, No. 237, s. 1–23.
 - [7] P. Nguyen et. al. Autonomous mobile robot navigation in sparse LiDAR feature environments. Appl. Sci., 2021, Vol. 11, No. 5963, s. 1–16.
 - [8] P. Penar et. al. Hardware-software compatibility in robotic cyber-physical systems - an application based approach. ASTRJ, 2025, Vol. 12, No. 6, s. 1–10.
 - [9] S. Quan et al. Mecanum wheeled mobile platform odometry error analysis and correction. In: Proceedings of IncoME-V & CEPE Net-2020. *Proceedings*. Zhuhai, China, October 23–25, 2020. Vol. 105, s. 728–736.
 - [10] J. Szrek et. al. Accuracy evaluation of selected mobile inspection robot localization techniques in a GNSS-denied environment. Sensors, 2021, Vol. 21, No. 141, s. 1–23.
 - [11] M. Szuster. Rozmyte sterowanie behawioralne mobilnym robotem kołowym w nieznanym środowisku. Zesz. Nauk. Politech. Rzeszow., Mech. OWPRz, 2014, t. XXXI, z.86, s. 603–619.
 - [12] P. Trybała et. al. Designing and evaluating a portable LiDAR-based SLAM system. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., 2023, Volume XLVIII-1/W3-2023, s. 191–198.
 - [13] T. Yang et. al. 3D ToF LiDAR in mobile robotics: a review. arXiv, 2022, Vol. 2202.11025, s. 1–16.

Application of the SLAM algorithm in the implementation of the "goal seeking" task

Measuring the position and orientation of a mobile robot frame in an unknown environment is an important issue in planning its autonomous movement. The paper presents a comparison between the implementation of the "goal seeking" task using the method of determining the robot's position and orientation based on the measurement of the rotation angles of the driving wheels and using a hybrid method that combines continuous determination of the frame's position and orientation based on odometry with periodic parameter updates determined using the SLAM algorithm based on measurements from the LIDAR sensor. The robot's motion is controlled by a hierarchical control system with a trajectory generation layer built using fuzzy logic systems and a tracking control layer in the form of a PD controller. Verification tests of the algorithm using a mobile robot with mecamum wheels and the Vicon motion capture system confirmed the positive impact of taking into account the robot's position correction on the precision of reaching the goal final position.

STEROWANIE ROBOTÓW

Sterownik czasu ustalonego dla monocykla w pozycyjnym zadaniu odtwarzania ścieżki*

Maciej Marcin Michałek¹, Mikołaj Jerzy Przybylski¹, Rafał Mateusz Sobański^{1,2}

Streszczenie

Artykuł przedstawia projekt oraz analizę stabilności i odporności układu sterowania pozycją wybranego punktu prowadzenia robota mobilnego klasy (2,0) w zadaniu odtwarzania ścieżki nieparametryzowanej. Zaproponowany sterownik wykorzystuje metodykę sterowania w czasie ustalonym (ang. *fixed-time control*), gdzie zbieżność błędu odtwarzania ścieżki zbiega do zera w skończonym horyzoncie czasowym, którego górne ograniczenie nie zależy od wyboru warunku początkowego konfiguracji robota względem ścieżki. Dodatkowa analiza ujawnia, że układ zamknięty z zaproponowanym sterownikiem czasu ustalonego wykazuje większą odporność na zakłócenia obecne w torze głównym w porównaniu do układu ze sterownikiem wykładniczym. Wyniki teoretyczne zostały zweryfikowane symulacyjnie.

1. Wprowadzenie

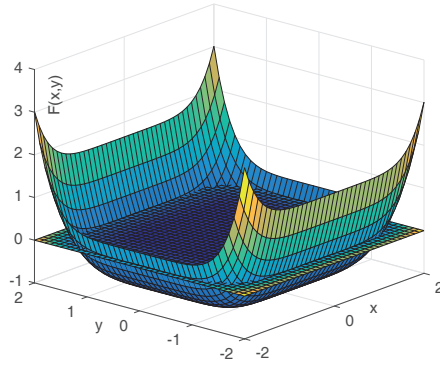
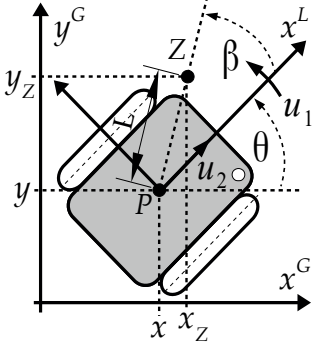
Zadanie odtwarzania ścieżki jest obecnie jednym z najbardziej popularnych praktycznych zadań ruchu robotów mobilnych i zautomatyzowanych pojazdów. W literaturze można znaleźć liczne prace poświęcone projektowaniu sterowników dla tego typu zadania [2, 3, 6, 9, 4]. Klasyczne rozwiązania wykorzystują parametryczną reprezentację ścieżki referencyjnej jej długością krzywoliniową i proponują sterowniki oparte na aproksymacji liniowej dynamiki uchybu lub nieliniowe wersje sterowników kinematycznych gwarantujących asymptotyczną zbieżność uchybu w czasie nieskończonym. W ostatnich latach z dużym zainteresowaniem badaczy spotyka się koncepcja projektowania sterowników czasu skończonego (ang. *finite-time control*, [7]) i czasu ustalonego (ang. *fixed-time control*, [10, 1]). Sterowniki takie zapewniają zbieżność uchybu do zera w czasie skończonym, a w przypadku algorytmów czasu ustalonego horyzont zbieżności jest dodatkowo ograniczony od góry poprzez stałą wartość maksymalnego czasu ustalania, która jest niezależna od wyboru warunku początkowego obiektu sterowania (tu: konfiguracji robota względem ścieżki referencyjnej). Gwarancja istnienia górnego ograniczenia czasu zbieżności może być przydatna w takich zastosowaniach, jak synchronizacja czasowa systemów wielorobotowych lub certyfikowane zarządzanie zdarzeniami krytycznymi ze względu na bezpieczeństwo.

W niniejszym artykule przedstawiamy projekt i analizę układu sterowania czasu ustalonego dla robota mobilnego klasy (2,0) w zadaniu pozycyjnego odtwarzania nieparametryzowanej ścieżki referencyjnej reprezentowanej za pomocą poziomiczy zero-

*Praca finansowana z subwencji badawczej nr 0211/SBAD/0124.

¹Instytut Automatyki i Robotyki, Politechnika Poznańska, ul. Piotrowo 3A, 60-138 Poznań, email (autor korespondencyjny): maciej.michalek@put.poznan.pl

²Université Polytechnique Hauts de France, LAMIH UMR CNRS 8201, Valenciennes, France



Rys. 1: Schemat kinematyczny robota mobilnego (strona lewa) oraz przykładowa reprezentacja ścieżki referencyjnej (tu: supereliptycznej) jako poziomica zerowa pewnej funkcji $F(x, y) = z$ (strona prawa)

wej pewnej funkcji zależnej od współrzędnych pozycji wybranego punktu prowadzenia robota [4, 11, 12]. W odróżnieniu od sterownika czasu ustalonego zaproponowanego w [5] dla zadania sterowania do punktu, rozwiązanie prezentowane w niniejszym artykule dotyczy zadania pozycyjnego i wykorzystuje znaną koncepcję linearyzacji statycznym sprzężeniem zwrotnym [2]. Ponadto przedstawiona analiza odporności wynikowego układu zamkniętego ilustruje, jak własność zbieżności w czasie ustalonym jest powiązana ze wzrostem odporności układu zamkniętego na pewien rodzaj zakłóceń obecnych w torze głównym. Artykuł prezentuje wyniki formalne oraz ich weryfikację numeryczną.

2. Model robota, założenia i definicja zadania sterowania

Rozważmy kinematykę monocykla (por. schemat na rys. 1, [2]):

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{u} = \begin{bmatrix} 1 & 0 \\ 0 & \cos \theta \\ 0 & \sin \theta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{q} \in \mathbb{R}^3, \quad \mathbf{u} \in \mathbb{R}^2, \quad (1)$$

z wektorem konfiguracji $\mathbf{q} \triangleq [\theta \ x \ y]^\top \in \mathbb{R}^3$ i wektorem sterowania składającym się z prędkości kątowej u_1 platformy robota i prędkości postępowej u_2 punktu P platformy (por. rys. 1).

Punktem prowadzenia pojazdu będzie punkt Z , zaznaczony na rys. 1, o współrzędnych (x_Z, y_Z) . Zakłada się, że punkt Z jest odsunięty od osi y^L układu lokalnego robota o odległość $L \cos \beta$, przy czym $L > 0$ i $\beta \in (-\pi; \pi]$, $|\beta| \neq \pi/2$, są stałymi parametrami projektowymi [2]. W konsekwencji, równanie (1) można uzupełnić równaniem wyjścia

$$\mathbf{y} \triangleq \begin{bmatrix} x_Z \\ y_Z \end{bmatrix} = \begin{bmatrix} x + L \cos(\theta + \beta) \\ y + L \sin(\theta + \beta) \end{bmatrix} \in \mathbb{R}^2 \quad \text{takim, że zachodzi} \quad L \cos \beta \neq 0. \quad (2)$$

Założenie z prawej strony (2) oznacza, że punkt Z nie leży na osi y^L układu lokalnego

robotu. Spełnienie tego warunku będzie konieczne dla dobrego określenia strategii sterowania pozycyjnego zaproponowanego w rozdziale 3.

W celu ścisłego określenia zadania sterowania wprowadźmy pewien spójny podzbiór pozycji $\mathcal{D} = \{(a_1, a_2)\} \subseteq \mathbb{R}^2$ oraz określoną w tym podzbiórze pewną funkcję $F(a_1, a_2)$ i odpowiadającą jej referencyjną ścieżkę pozycji w postaci zbioru S_d punktów $\{(a_1 = x_d, a_2 = y_d)\}$ zdefiniowanych następująco [4]:

$$S_d \triangleq \{(x_d, y_d) \in \mathcal{D} : F(x_d, y_d) = \sigma f(x_d, y_d) = 0\}, \quad \sigma \in \mathbb{R} \setminus \{0\}, \quad (3)$$

gdzie $F(x_d, y_d) = 0$ reprezentuje poziomice zerową funkcji $F(a_1, a_2)$, natomiast parametr skalujący σ (patrz rys. 1) jest elementem projektowym; znak parametru σ decyduje o żądanej strategii ruchu *prawo-lewo* wzdłuż ścieżki. Postać funkcji $F(a_1, a_2)$ determinuje kształt ścieżki referencyjnej pozycji w przestrzeni roboczej robota. Ponadto przyjmiemy, że funkcja F spełnia następujące założenia [6, 4]:

Z1. funkcja $F(a_1, a_2)$ jest ograniczona, tj. $\forall (a_1, a_2) \in \mathcal{D}$ zachodzi $F(a_1, a_2) \in [\underline{F}; \overline{F}] \subset \mathbb{R}$, gdzie $\underline{F} \in (-\infty; 0)$, $\overline{F} \in (0; \infty)$,

Z2. funkcja $F(a_1, a_2)$ jest różniczkowalna względem swoich argumentów oraz

$$\forall (a_1, a_2) \in \mathcal{D} \quad \|\nabla F(a_1, a_2)\| \in [g; G], \quad \nabla F(a_1, a_2) \triangleq \begin{bmatrix} \partial F(a_1, a_2)/\partial a_1 \\ \partial F(a_1, a_2)/\partial a_2 \end{bmatrix}, \quad (4)$$

gdzie $0 < g < G < \infty$, natomiast $\|\cdot\|$ oznacza euklidesową normę wektora.

Od tej pory wartość funkcji $F(a_1, a_2)$ dla $a_1 := x_Z$ i $a_2 := y_Z$ (tj. dla $F(a_1, a_2) := F(\mathbf{y})$, por. (2)) będziemy interpretować jako pewną miarę (nieeuklidesowej) odległości ze znakiem punktu Z prowadzenia pojazdu od zbioru referencyjnego S_d , ponieważ $F(\mathbf{y}) = 0$ tylko, gdy $\mathbf{y} = [x_Z \ y_Z]^\top = [x_d \ y_d]^\top$. Miarę tę wykorzystamy w roli uchybu pozycyjnego w następującym zadaniu sterowania.

Problem 1. *Znaleźć ograniczoną funkcję sterującą $\mathbf{u} = \mathbf{u}(\mathbf{q})$ dla kinematyki (1) gwarantującą zbieżność wartości funkcji $F(\mathbf{y}(t))$ do zera w czasie ustalonym, tj.: dla każdego warunku początkowego $\mathbf{y}_0 = \mathbf{y}(t=0) \in \mathcal{D} \subseteq \mathbb{R}^2$ zachodzi*

$$\forall t \geq t_s(\mathbf{y}_0) \quad F(\mathbf{y}(t)) = 0 \quad \text{oraz} \quad \sup_{\mathbf{y}_0 \in \mathcal{D}} t_s(\mathbf{y}_0) \leq T_s < \infty, \quad (5)$$

gdzie $t_s(\mathbf{y}_0)$ reprezentuje czas ustalania zależny od warunku początkowego $\mathbf{y}_0$, natomiast T_s jest stałym i niezależnym od warunku początkowego górnym ograniczeniem horyzontu czasowego zbieżności.

Warunki wprowadzone w (5) wymagają, aby zbieżność uchybu pozycji do zera nastąpiła w czasie skończonym i nie dłuższym niż T_s , niezależnie od wyboru warunku początkowego $\mathbf{y}_0$ w całym i dowolnie dużym podzbiórze pozycji $\mathcal{D} \subseteq \mathbb{R}^2$.

3. Równania sterownika czasu ustalonego

Łatwo sprawdzić, że dynamika w torze między wejściem sterującym $\mathbf{u}$ a wyjściem (2) przyjmuje postać:

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{x}_Z \\ \dot{y}_Z \end{bmatrix} = \mathbf{D}(\theta, \beta) \mathbf{u} = \begin{bmatrix} -L \sin(\theta + \beta) & \cos \theta \\ L \cos(\theta + \beta) & \sin \theta \end{bmatrix} \mathbf{u}, \quad (6)$$

przy czym macierz $\mathbf{D}(\theta, \beta)$ jest nieosobliwa dla każdego kąta θ , gdy tylko spełniony jest warunek podany po prawej stronie definicji (2). Stosując klasyczne podejście wynikające z koncepcji linearyzacji statycznym sprzężeniem zwrotnym proponuje się definicję pętli linearyzująco-odsprzęgającej

$$\mathbf{u} \triangleq \mathbf{D}^{-1}(\theta, \beta)\mathbf{w}, \quad (7)$$

z nowym wejściem sterującym $\mathbf{w} \in \mathbb{R}^2$. W celu uzyskania zbieżności uchybu pozycji w czasie ustalonym proponuje się następującą postać sterownika pętli zewnętrznej:

$$\mathbf{w} \triangleq k_p \text{sign}(F(\mathbf{y})) \left[|F(\mathbf{y})|^{1+\delta} + |F(\mathbf{y})|^{1-\delta} \right] \boldsymbol{\vartheta} + v_d \mathbf{R} \boldsymbol{\vartheta}, \quad (8)$$

gdzie $\text{sign}(\cdot)$ jest standardową funkcją znaku taką, że $\text{sign}(0) = 0$, natomiast $k_p > 0$, $\delta \in (0; 1)$ oraz $v_d = \text{const} > 0$ to parametry projektowe, przy czym

$$\boldsymbol{\vartheta} \triangleq \frac{-\nabla F(\mathbf{y})}{\|\nabla F(\mathbf{y})\|}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (9)$$

reprezentują, odpowiednio, wersor ujemnego gradientu funkcji $F(\mathbf{y})$ oraz macierz rotacji o kąt $-\pi/2$. Na podstawie założenia Z2 wersor $\boldsymbol{\vartheta}$ jest dobrze określony w całym podzbiorze $\mathcal{D}$. Definicja (8) ma intuicyjną interpretację geometryczną. Pierwszy składnik odpowiada za sprzężenie zwrotne i wskazuje w kierunku zbioru S_d , zgodnie z ujemnym gradientem funkcji $F(\mathbf{y})$. Drugi składnik ma charakter sprzężenia wyprzedzającego (ortogonalnego do składnika sprzężenia zwrotnego), które w przypadku $F(\mathbf{y}) \equiv 0$ pozwala na prowadzenie punktu Z robota wzdłuż ścieżki referencyjnej ze stałą zadaną prędkością v_d . Przy ustalonej prędkości v_d proporcję między oboma wektorowymi składnikami można narzucać poprzez dobór wartości parametru k_p .

4. Analiza stabilności i zbieżności w czasie ustalonym

Zanim przejdziemy do analizy stabilności przytoczymy definicję stabilności w czasie ustalonym oraz ogólny wynik dotyczący stabilności w czasie ustalonym w postaci stosownego lematu. W tym celu zakładamy, że dany jest pewien system dynamiczny S opisany równaniem $\dot{\mathbf{z}}(t) = \mathbf{f}(\mathbf{z}(t))$, którego odpowiedź startującą z warunku początkowego $\mathbf{z}(0) = \mathbf{z}_0 \in \mathbb{R}^n$ oznaczmy jako $\mathbf{z}(t, \mathbf{z}_0)$. Ponadto zakładamy, że dla systemu S istnieje rozwiązanie stałe $\mathbf{z}(t, \mathbf{z}_0 = \mathbf{0}) \equiv \mathbf{0} \in \mathbb{R}^n$ będące jego punktem równowagi.

Definicja 1 (por. [10, 1]) Punkt równowagi $\mathbf{z} = \mathbf{0}$ systemu S nazywamy stabilnym w czasie ustalonym, jeżeli jest on stabilny w sensie Lapunowa oraz $\forall t \geq t_s(\mathbf{z}_0)$ zachodzi $\mathbf{z}(t, \mathbf{z}_0) = \mathbf{0}$, przy czym czas ustalania $t_s(\mathbf{z}_0) < \infty$ oraz istnieje skończona górna granica $T_s < \infty$ taka, że $t_s(\mathbf{z}_0) \leq T_s$ dla dowolnego warunku początkowego $\mathbf{z}_0 \in \mathbb{R}^n$.

Lemat 1 (por. [10, 1]) Punkt równowagi $\mathbf{z} = \mathbf{0}$ systemu S jest stabilny w czasie ustalonym, jeśli istnieje dodatnio określona ciągła funkcja $V : \mathbb{R}^n \rightarrow \mathbb{R}_+ \cup \{0\}$ taka, że dowolna trajektoria $\mathbf{z}(t, \mathbf{z}_0)$ systemu S startująca z warunku początkowego $\mathbf{z}_0 \in \mathbb{R}^n$ spełnia nierówność:

$$\dot{V}(\mathbf{z}) \leq -\gamma_1 V^{p_1}(\mathbf{z}) - \gamma_2 V^{p_2}(\mathbf{z}) \quad (10)$$

dla $\gamma_1, \gamma_2 > 0$, $p_1 > 1$ oraz $p_2 \in (0; 1)$.

Lemat 2 (por. [8]) *Jeżeli punkt równowagi systemu S jest stabilny w czasie ustalonym, tzn. jeżeli istnieje funkcja V spełniająca warunki Lematu 1, oraz jeśli $p_1 = 1 + \alpha$, $p_2 = 1 - \alpha$, dla $\alpha \in (0; 1)$, to górną granicę T_s czasu zbieżności można oszacować następująco:*

$$T_s = \frac{\pi}{2\alpha\sqrt{\gamma_1\gamma_2}}. \quad (11)$$

Pokażemy teraz, że sterownik (7) z nowym wejściem $\mathbf{w}$ zdefiniowanym w (8) prowadzi do rozwiązania Problemu 1 i gwarantuje stabilność punktu $F(\mathbf{y}) = 0$ w czasie ustalonym. W tym celu wyprowadzamy równanie dynamiki zmiany funkcji $F(\mathbf{y})$ w układzie zamkniętym ze sterownikiem zdefiniowanym równaniami (7) i (8):

$$\begin{aligned} \dot{F} &= \nabla F^\top \dot{\mathbf{y}} \stackrel{(6)}{=} \nabla F^\top \mathbf{D}(\theta, \beta) \mathbf{u} \stackrel{(7)}{=} \nabla F^\top \mathbf{w} = \\ &\stackrel{(8)}{=} \nabla F^\top k_p \operatorname{sign}(F) \left[|F|^{1+\delta} + |F|^{1-\delta} \right] \boldsymbol{\vartheta} + \underbrace{\nabla F^\top v_d \mathbf{R} \boldsymbol{\vartheta}}_{\equiv 0} = \\ &= -k_p \operatorname{sign}(F) \left[|F|^{1+\delta} + |F|^{1-\delta} \right] \frac{\nabla F^\top \nabla F}{\|\nabla F\|} = \\ &= -k_p \operatorname{sign}(F) \|\nabla F\| \left[|F|^{1+\delta} + |F|^{1-\delta} \right], \end{aligned} \quad (12)$$

przy czym tożsamościowe zerowanie się składnika z drugiej linii powyższego wyprowadzenia wynika wprost z definicji (9), gdzie $\mathbf{R}$ jest macierzą skośnie symetryczną. Na podstawie założenia Z2 można wnioskować, że $F = 0$ jest punktem równowagi dynamiki (12). Wprowadźmy teraz dodatkowo określoną funkcję $V(F) \triangleq F^2$, której pochodną po czasie wzdłuż trajektorii systemu dynamicznego (12) możemy szacować jak następuje:

$$\begin{aligned} \dot{V}(F) &= 2F\dot{F} \stackrel{(12)}{=} -2k_p \|\nabla F\| \left[|F|^{2+\delta} + |F|^{2-\delta} \right] = \\ &= -2k_p \|\nabla F\| \left[V^{(2+\delta)/2}(F) + V^{(2-\delta)/2}(F) \right] \leq \\ &\leq -2gk_p \left[V^{1+\delta/2}(F) + V^{1-\delta/2}(F) \right], \end{aligned} \quad (13)$$

przy czym w ostatniej linii skorzystano z założenia Z2. Porównanie uzyskanej nierówności (13) z wynikiem ogólnym (10) daje $\gamma_1 = \gamma_2 = 2gk_p > 0$ oraz $p_1 = (1 + \delta/2) > 1$ i $p_2 = (1 - \delta/2) \in (0; 1)$, co pozwala wnioskować o stabilności punktu równowagi $F(\mathbf{y}) = 0$ w czasie ustalonym i tym samym spełnienie warunków (5) dla trajektorii $F(\mathbf{y}(t))$ startującej z warunku początkowego $F(\mathbf{y}_0)$ dla dowolnego $\mathbf{y}_0 \in \mathcal{D}$. Ponadto, porównując nierówność (13) z (10) i korzystając z Lematu 2 można zauważyć, że $\alpha = \delta/2$. W konsekwencji, górna granica T_{sF} czasu zbieżności funkcji $F(\mathbf{y}(t))$ przyjmuje postać:

$$T_{sF} \stackrel{(11)}{=} \frac{\pi}{2\delta g k_p}. \quad (14)$$

Uwaga 1. Przyjmując w definicji (8) parametr projektowy $\delta = 0$ (wartość spoza postulowanego przedziału) uzyskamy wersję sterownika prowadzącą (zamiast do (13)) do nierówności $\dot{V}(F) \leq -4gk_p V(F)$, i tym samym do wykładniczej zbieżności uchybu w czasie nieskończonym, tj. $F(\mathbf{y}(t)) \rightarrow 0$ dla $t \rightarrow \infty$.

5. Analiza odporności na zakłócenie w torze głównym

Zbieżność w czasie ustalonym niesie ze sobą konsekwencje związane z odpornością układu zamkniętego m.in. na zakłócenia obecne w torze głównym dynamiki (6). Aby przeanalizować tego rodzaju odporność dla układu z zaproponowanym sterownikiem założmy, że nominalną dynamikę wyjścia (6) zastąpimy jej wersją zaburzoną

$$\dot{\mathbf{y}} = \mathbf{D}(\theta, \beta)\mathbf{u} + \boldsymbol{\mu}, \quad \|\boldsymbol{\mu}\| \in [\underline{m}; \bar{m}], \quad 0 \leq \underline{m} < \bar{m} < \infty, \quad (15)$$

gdzie $\boldsymbol{\mu} = [\mu_x \ \mu_y]^\top \in \mathbb{R}^2$ reprezentuje ograniczone zaburzenie. Teraz równanie dynamiki uchybu pozycyjnego (12) przyjmie następującą postać zaburzoną:

$$\begin{aligned} \dot{F} &= \nabla \mathbf{F}^\top \dot{\mathbf{y}} \stackrel{(15)}{=} \nabla \mathbf{F}^\top [\mathbf{D}(\theta, \beta)\mathbf{u} + \boldsymbol{\mu}] \stackrel{(7)}{=} \nabla \mathbf{F}^\top \mathbf{w} + \nabla \mathbf{F}^\top \boldsymbol{\mu} = \\ &\stackrel{(8)}{=} \nabla \mathbf{F}^\top k_p \operatorname{sign}(F) \left[|F|^{1+\delta} + |F|^{1-\delta} \right] \boldsymbol{\vartheta} + \nabla \mathbf{F}^\top \boldsymbol{\mu} = \\ &= -k_p \operatorname{sign}(F) \|\nabla \mathbf{F}\| \left[|F|^{1+\delta} + |F|^{1-\delta} \right] + \nabla \mathbf{F}^\top \boldsymbol{\mu}. \end{aligned} \quad (16)$$

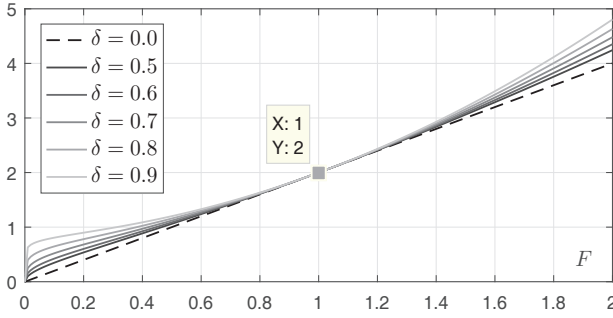
Stosując dodatnio określoną funkcję $V(F) \triangleq F^2$, jej pochodną po czasie możemy w przypadku zaburzonym szacować jak następuje:

$$\begin{aligned} \dot{V}(F) &= 2F\dot{F} \stackrel{(16)}{=} -2k_p \|\nabla \mathbf{F}\| \left[|F|^{2+\delta} + |F|^{2-\delta} \right] + 2F\nabla \mathbf{F}^\top \boldsymbol{\mu} \leq \\ &\leq -2k_p \|\nabla \mathbf{F}\| \left[|F|^{2+\delta} + |F|^{2-\delta} \right] + 2|F| \|\nabla \mathbf{F}\| \bar{m} \leq \\ &\leq -2gk_p |F| \left[|F|^{1+\delta} + |F|^{1-\delta} - \bar{m}/k_p \right] = -2gk_p |F| H(F, \bar{m}). \end{aligned} \quad (17)$$

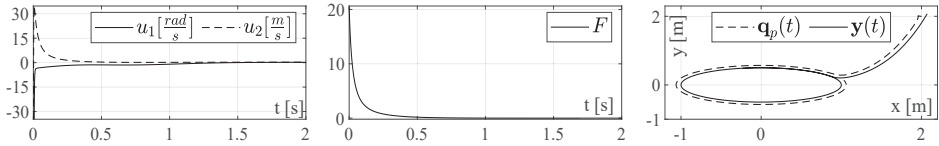
Zatem (konserwatywny) warunek zbieżności uchybu $F(\mathbf{y}(t))$ w przypadku obecności zaburzenia $\boldsymbol{\mu}$ wymaga, aby $H(F, \bar{m}) > 0$, co prowadzi do nierówności:

$$|F|^{1+\delta} + |F|^{1-\delta} > \bar{m}/k_p. \quad (18)$$

Przy ustalonej wartości wzmocnienia $k_p > 0$ najmniejsza wartość $|F(\mathbf{y})|$, dla której spełniony będzie warunek (18) będzie zależeć od wartości parametru projektowego δ . Rysunek 2 przedstawia wykres funkcji $f(F) = |F|^{1+\delta} + |F|^{1-\delta}$ dla kilku różnych wartości δ , łącznie z przypadkiem $\delta = 0$, który odpowiada sterownikowi prowadzącemu do wykładniczej zbieżności uchybu w czasie nieskończonym. Na podstawie wykresów z rys. 2 można stwierdzić, że im większa wartość parametru $\delta \in (0; 1)$, tym spełnienie nierówności (18) będzie możliwe dla mniejszej wartości $|F|$ (poza jednym punktem wspólnym wszystkich wykresów). Tym samym odporność układu zamkniętego na zakłócenie $\boldsymbol{\mu}$ będzie tym większa, im większą wartość δ wybierze projektant (np. wykres dla $\delta = 0.9$ leży zawsze, poza pojedynczym punktem wspólnym, powyżej wykresu dla $\delta < 0.9$). Ponadto poza punktem odpowiadającym $|F| = 1$ sterownik czasu ustalonego (tj. dla $\delta \in (0; 1)$) wykazuje większą odporność na obecność zakłócenia $\boldsymbol{\mu}$ niż sterownik prowadzący do zbieżności wykładniczej (tj. dla $\delta = 0$).



Rys. 2: Wykres funkcji $f(F) = |F|^{1+\delta} + |F|^{1-\delta}$ dla różnych wartości parametru δ (pośrodku zaznaczono punkt wspólny wykresów)



Rys. 3: Przykładowe wyniki symulacyjne otrzymane dla $\mathbf{q}_0 = [0 \ 2 \ 2]^\top$ i $\delta = 0.2$, przy czym $\mathbf{q}_p = [x \ y]^\top$. Wykres sygnałów sterujących został ograniczony w osi rzędnych w celu zachowania czytelności (maksymalne wartości bezwzględne sygnałów sterujących wyniosły: $\sup_{t \geq 0} |u_1(t)| = 676.50 \text{ rad/s}$, $\sup_{t \geq 0} |u_2(t)| = 59.06 \text{ m/s}$)

6. Weryfikacja symulacyjna

W celu weryfikacji jakości działania sterownika przeprowadzono testy symulacyjne w środowisku Matlab dla zadania odtwarzania ścieżki eliptycznej danej równaniem $F(x_d, y_d) = x_d^2 + 4y_d^2 - 1 = 0$. We wszystkich testach użyto następujących wartości parametrów: $L = 0.1 \text{ m}$; $\beta = \pi/4 \text{ rad}$; $k_p = 1$; $v_d = 0.25 \frac{\text{m}}{\text{s}}$.

W celu przedstawienia wpływu początkowej wartości funkcji F na maksymalne wartości sygnałów sterujących przeprowadzono test dla $\mathbf{q}_0 = [0 \ 2 \ 2]^\top$ i $\delta = 0.2$. Początkowa wartość F wynosiła 20.44. Uzyskane przebiegi pozycji punktu Z, pozycji robota $\mathbf{q}_p = [x \ y]^\top$, sygnałów sterujących oraz funkcji F przedstawiono na rys. 3. Rzeczywisty czas zbieżności t_s funkcji $F(y(t))$ do otoczenia zera o promieniu 10^{-3} wyniósł 1.21 s, natomiast obliczona wg wzoru (14) wartość górnej granicy T_{sF} czasu zbieżności wyniosła 3.93 s. Należy zauważyć, że dla warunków początkowych położonych dalej od ścieżki referencyjnej zbieżność w czasie ustalonym wymaga większego kosztu sterowania, w tym większych maksymalnych bezwzględnych wartości sygnałów sterujących. Wartości maksymalne sterowań można ograniczyć wprowadzając operację skalowania prędkości (por. [4]), jednak w konsekwencji wymaga to modyfikacji postaci górnej granicy czasu zbieżności ze wzorów (11) i (14).

W celu zobrazowania zbieżności w czasie ustalonym wykonano testy dla 84 różnych konfiguracji początkowych robota, przy czym poszczególne zmienne konfiguracyjne przyjmowały wartości z następujących zbiorów: $\theta_0 \in \{0, 0.5\pi, \pi, 1.5\pi\}$,

Tab. 1: Wartości F^* uzyskane dla wybranych wartości wartości parametru δ

δ	0	0.1	0.3	0.5	0.7	0.9
F^*	$6.3 \cdot 10^{-3}$	$5.5 \cdot 10^{-3}$	$1.9 \cdot 10^{-3}$	$1.56 \cdot 10^{-4}$	$4.53 \cdot 10^{-7}$	$9.31 \cdot 10^{-20}$

$x_0 \in \{-1.5, -1, -0.5, 0, 0.5, 1, 1.5\}$ oraz $y_0 \in \{-1, 0, 1\}$. Symulacje przeprowadzono dla $\delta \in \{0.2, 0.5, 0.8\}$. Uzyskane przebiegi pozycji punktu Z , sygnałów sterujących oraz funkcji F przedstawiono na rys. 4-6. W każdym przypadku można zaobserwować zbieżność wartości funkcji $F(\mathbf{y}(t))$ do otoczenia zera o promieniu 10^{-3} . Wartości górnych ograniczeń T_{sF} czasu zbieżności obliczone zgodnie z (14) dla przyjętych wartości δ wyniosły odpowiednio 13.47 s, 5.39 s oraz 3.37 s. Na podstawie wyników z rys. 4-6 można zaobserwować, że w każdym przypadku rzeczywisty czas zbieżności był mniejszy niż obliczona górna granica (tzn. $t_s(\mathbf{y}_0) < T_{sF}$).

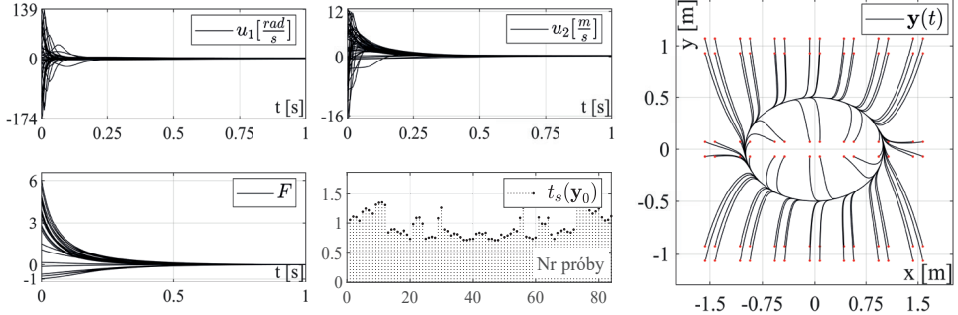
Obliczono również numerycznie najmniejsze wartości $|F|$ spełniające nierówność (18), tzn. $F^* = \inf_{F \in \mathbb{R}} |F| : |F|^{1+\delta} + |F|^{1-\delta} > \bar{m}/k_p$, w zależności od wybranych wartości δ , przyjmując $\bar{m} = 0.0125$ m/s i $k_p = 1$. Uzyskane wyniki, przedstawione w tab. 1, pozwalają zauważyć, że zwiększanie wartości parametru δ pozwala na spełnienie nierówności (18) dla coraz mniejszych wartości $|F|$, co oznacza, że negatywny skutek oddziaływania zaburzenia μ na jakość odtwarzania ścieżki będzie maleć wraz ze wzrostem wartości parametru δ w kierunku jedności.

7. Podsumowanie

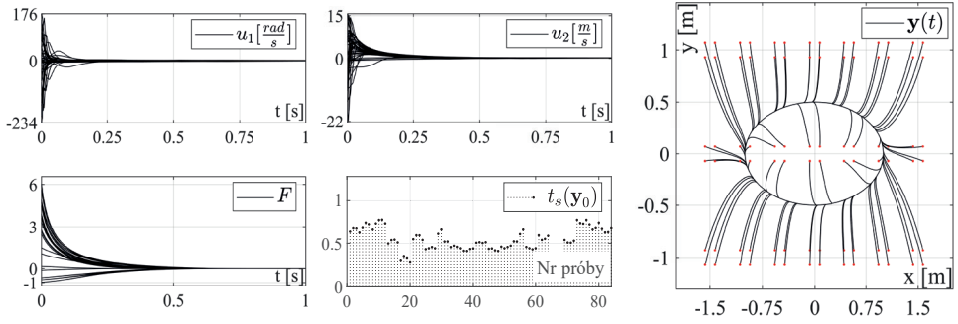
Artykuł przedstawia propozycję prawa sterowania czasu ustalonego dla kinematyki monocykla w zadaniu pozycyjnego odtwarzania ścieżki nieparametryzowanej. Uzyskane wyniki teoretyczne zostały potwierdzone wynikami symulacji numerycznych – w każdym analizowanym przypadku osiągnięto zbieżność funkcji F w kierunku zera w skończonym czasie, krótszym niż obliczone teoretyczne górne ograniczenie T_{sF} .

Analiza czasów zbieżności pozwala zaobserwować pewien konserwatyzm użytej metody szacowania ograniczenia T_{sF} . Można wskazać dwie główne przyczyny takiego stanu rzeczy. Pierwszą z nich jest natura koncepcji sterowania w czasie ustalonym, dla której rozważa się czas zbieżności dla całego gęstego zbioru warunków początkowych. Z tego względu, że wykorzystana metoda szacowania ograniczenia T_{sF} dotyczy przypadku globalnego, zbiór warunków początkowych $F(\mathbf{y}(0))$ może być dowolnie duży w $\mathbb{R}$. Drugą przyczyną jest konserwatywny sposób szacowania pochodnej funkcji $V(F)$, dla której uzyskuje się nierówność (13) zamiast równania. Mniej konserwatywne wyniki można uzyskać ograniczając rozważania wyłącznie do ograniczonego zbioru warunków początkowych (wersja lokalna); analiza lokalna może być jednym z przyszłych kierunków prac badawczych, z jednoczesnym uwzględnieniem obecności ograniczeń sygnałów sterujących.

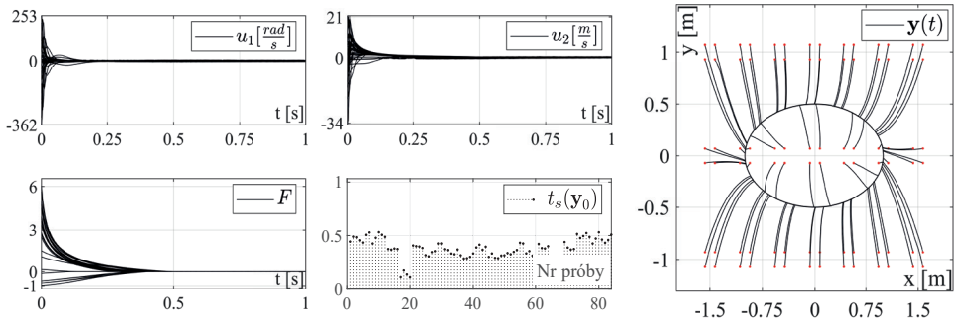
Na koniec warto zauważyć, że przeprowadzona analiza (potwierdzona wynikiem obliczeń numerycznych) wskazuje, iż układ zamknięty z zaproponowanym sterownikiem czasu ustalonego charakteryzuje się większą odpornością na zakłócenia obecne w torze głównym w porównaniu do układu zamkniętego ze sterownikiem prowadzącym do zbieżności jedynie wykładniczej.



Rys. 4: Wyniki symulacji oraz czasy zbieżności dla różnych warunków początkowych oraz $\delta = 0.2$; górna granica T_{sF} obliczona wg (14) dla tego przypadku wynosi 13.47 s



Rys. 5: Wyniki symulacji oraz czasy zbieżności dla różnych warunków początkowych oraz $\delta = 0.5$; górna granica T_{sF} obliczona wg (14) dla tego przypadku wynosi 5.39 s



Rys. 6: Wyniki symulacji oraz czasy zbieżności dla różnych warunków początkowych oraz $\delta = 0.8$; górna granica T_{sF} obliczona wg (14) dla tego przypadku wynosi 3.37 s

Literatura

- [1] R. Aldana-López, D. Gómez-Gutiérrez, E. Jiménez-Rodríguez, J. D. Sánchez-Torres, M. Defoort. Enhancing the settling time estimation of a class of fixed-time stable systems. *Int J Robust Nonlinear Control*, 29:4135–4148, 2019.
- [2] C. Canudas de Wit, B. Siciliano, G. Bastin. *Theory of Robot Control*. Springer-Verlag, New York, 1996.
- [3] L. Lapiere, D. Soetanto, A. Pascoal. Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties. *Int. J. Robust Nonlinear Control*, 16:485–503, 2006.
- [4] M. M. Michałek, T. Gawron. VFO path following control with guarantees of positionally constrained transients for unicycle-like robots with constrained control input. *Journal of Intelligent & Robotic Systems*, 89(1-2):191–210, 2018.
- [5] M. M. Michałek, R. M. Sobański, M. Defoort. Fixed-time VFO control for a unicycle. A. Mazur, C. Zieliński, redaktorzy, *Prace Naukowe. Elektronika, Tom I (197)*, strony 191–200. Oficyna Wydawnicza Politechniki Warszawskiej, 2022.
- [6] A. Morro, A. Sgorbissa, R. Zaccaria. Path following for unicycle robots with arbitrary path curvature. *IEEE Trans. Robotics*, 27(5):1016–1023, 2011.
- [7] E. Moulay, W. Perruquetti. Finite-time stability and stabilization: state of the art. C. Edwards et al., redaktor, *Advances in Variable Structure*, wolumen 334 serii *LNCIS*, strony 23–41. Springer-Verlag, 2006.
- [8] S. Parsegov, A. Polyakov, P. Shcherbakov. Nonlinear fixed-time control protocol for uniform allocation of agents on a segment. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, strony 7732–7737, 2012.
- [9] J. Płaskonka. Different kinematic path following controllers for a wheeled mobile robot of (2,0) type. *J. Intell. Robot. Syst.*, 77:481–498, 2015.
- [10] A. Polyakov. Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Trans. on Automatic Control*, 57(8):2106–2110, 2012.
- [11] W. Yao, H. Garcia de Marina, B. Lin, M. Cao. Singularity-free guiding vector field for robot navigation. *IEEE Trans. on Robotics*, 37(4):1206–1221, 2021.
- [12] W. Yao, B. Lina, B. D. O. Anderson, M. Cao. Topological analysis of vector-field guided path following on manifolds. *IEEE Trans. on Automatic Control*, 68(3):1353–1368, 2023.

Fixed-time controller for a unicycle in the positional path-following task

This paper presents the design and stability and robustness analysis of a position control system for a selected guidance point of a class (2,0) mobile robot in a non-parameterised path reconstruction task. The proposed controller uses a fixed-time control methodology, where the convergence of the path reconstruction error converges to zero over a finite time horizon, the upper limit of which does not depend on the choice of the initial condition of the robot configuration with respect to the path. Additional analysis reveals that the closed-loop system with the proposed steady-state time controller exhibits greater robustness to disturbances present in the main path compared to the system with an exponential controller. The theoretical results have been verified by simulation. (Translated with DeepL.com, free version)

Organizacja przerywania działań robotów

Cezary Zieliński¹

Streszczenie

Artykuł przedstawia sposób organizacji systemu robotycznego, którego działania mogą być przerywane. Powodem przerywania może być albo nowe polecenie albo wystąpienie nietypowej sytuacji. Obsługa przerywania może nie zostać zakończona, gdy nadejdzie kolejne. Zaproponowano formalną specyfikację sterownika rozwiązującego tak postawiony problem.

1. Wstęp

Motyacją do podjęcia zawartych tu rozważań były sytuacje, z którymi ludzie spotykają się na co dzień, a więc i roboty współpracujące z nimi także będą musiały sobie z nimi radzić. Załóżmy, że robot otrzymuje polecenie podania książki. Gdy wykonuje to polecenie kurier dzwoni do drzwi, więc musi odłożyć książkę, aby obsłużyć kuriera. Idąc do drzwi zauważa, że z zatkanego zlewu przelewa się woda, więc musi szybko zakręcić kran. Ten scenariusz można kontynuować w nieskończoność, ale kluczowe pytanie brzmi: w jaki sposób robot powinien przerywać swoje dotychczasowe działania, aby obsłużyć bardziej pilne żądanie? Jeśli robot niesie książkę, nie powinien jej upuszczać tylko dlatego, że zauważył wylewającą się wodę ze zlewu. Potrzebne jest dodatkowe działanie kończące dotychczasowe, by móc podjąć to nowe. W przeciwieństwie do obsługi przerw w komputerach tu powrót do przerywanych działań może nie być możliwy [20]. Osoba czekająca na książkę może sama sobie ją wziąć z półki, co oznacza, że zadanie zostało zakończone, ale nie przez robota, lecz przez człowieka. Nie ma zatem potrzeby wznowiania tego zadania.

Rozwiązanie podobnego problemu zaproponowano w [5, 6, 7]. W tym przypadku system sterowania robota składał się z: wielu agentów zgłaszających zadania, wielu agentów dynamicznych odpowiedzialnych za realizację określonych zadań, agenta podstawowego sterującego sprzętem robota, agenta zarządzającego repozytorium plików zawierających agenty dynamiczne, agenta platformy zapewniającego usługi obliczeniowe, wielu agentów chmurowych wspierających agenty dynamiczne w realizacji obliczeń związanych z przydzielonymi im zadaniami, agenta szeregującego zadania, odpowiedzialnego za podejmowanie decyzji dotyczących tego, który agent dynamiczny powinien sterować agentem podstawowym (a tym samym, które zadanie będzie realizowane). Architektura RAPP [22] była tu wzorcem. Rozróżniono zadania na możliwe do przerywania i takie, których nie wolno przerywać. Zakładano, że przerywane zadanie może zostać wznowione — niekoniecznie od tego samego stanu, w którym zostało przerywane. Przy zawieszaniu i wznowianiu wykonywane były pewne zdefiniowane wcześniej czynności. Agent szeregujący zadania decydował, które zadanie powinno być podjęte, na podstawie określonych parametrów związanych

¹ Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Instytut Automatyki i Informatyki Stosowanej, 00-665 Warszawa, ul. Nowowiejska 15/19, cezary.zielinski@pw.edu.pl

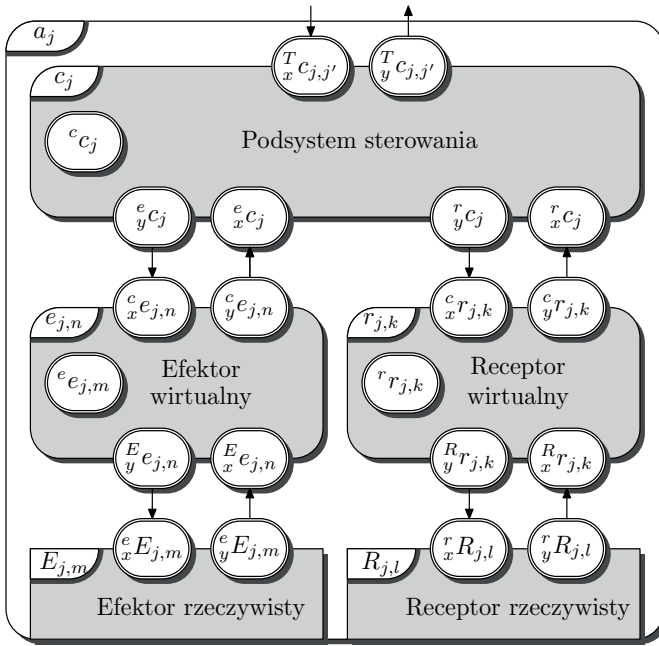
z aktywnymi zadaniami, tj. aktywnymi agentami dynamicznymi. Parametry te mogły ulec zmianie, co mogło doprowadzić nawet do porzucenia zadania, jeśli powód jego realizacji przestał być aktualny. Algorytm szeregowania był konfigurowalny, a priorytety zadań mogły się zmieniać w zależności od szacowanego czasu potrzebnego na zawieszenie bieżących działań robota oraz czasu potrzebnego na ukończenie zadania. Agent dynamiczny wykonujący zadanie odpowiadał za oszacowanie tych parametrów na podstawie heurystyki, uwzględniającej etap realizacji zadania, co w rzeczywistości nie zawsze było bezpośrednio powiązane z aktualnym i pożądanym stanem środowiska. Ponieważ system nie przewidywał zarządzania wiedzą ani planowania, te heurystyki musiały być niedokładne. Wznowienie działań agenta dynamicznego opierało się na danych przygotowanych w momencie jego zawieszenia, co zakładało, że dane te nadal będą aktualne w momencie ponownej aktywacji zadania.

Artykuł traktuje problem wielokrotnych przerw jako teoretyczny i przedstawia architektoniczne rozwiązanie z użyciem agentów upostaciowionych [2, 3]. Architektura systemu opiera się na sterownikach o zmiennej strukturze [22] i wykorzystuje specyficzną strukturę przedstawioną w [14, 16], co upraszcza architekturę, jednocześnie włączając do niej planowanie i zarządzanie wiedzą. Podobnie jak w [5, 6, 7], artykuł dotyczy zadań wykonywanych przez pojedynczego robota realizującego jedno zadanie na raz – Single-Robot Single-Task według klasyfikacji przedstawionej w [11, 15]. Nie zajmuje się alokacją zadań ani ich szeregowaniem [6, 7], bo ten problem powinien zostać rozwiązany przez planer. Zakłada, że może dochodzić do wielokrotnych przerw. Nie dzieli zachowań na zawieszalne i niezawieszalne. Brak konieczności zawieszenia zadania oznacza, że nie trzeba go wznawiać. Zachowania mogą być przerwane, bądź nie, co rozwiązywane jest poprzez odpowiednie sformułowanie ich warunków końcowych. Szeregowanie zadań opiera się na planowaniu, dzięki czemu można uwzględnić aktualny stan bazy wiedzy. Decyzja o kontynuowaniu przerwanej zadania podejmowana jest więc na podstawie bieżącego stanu środowiska. Nie wprowadzono zegara wyzwalającego planowanie zadań — system opiera się wyłącznie na szeregowaniu zdarzeniowym. Dla zwięzłości problem trybów komunikacji [21, 18, 8, 9] między podsystemami nie jest omawiany w tym artykule, ale może zostać łatwo uwzględniony przy użyciu rozwiązań zaprezentowanych w [19].

2. Agent upostaciowiony

Agent upostaciowiony (rys. 1) stanowi podstawowy element konstrukcyjny systemu robotycznego [19]. Każdy agent a_j (j – nazwa agenta) składa się z pojedynczego podsystemu sterowania c_j oraz zera lub więcej: efektorów rzeczywistych $E_{j,m}$, efektorów wirtualnych $e_{j,n}$, receptorów rzeczywistych $R_{j,l}$ i receptorów wirtualnych $r_{j,k}$, gdzie m, n, l, k są nazwami wymienionych podsystemów. Podsystem sterowania c_j realizuje zadanie przypisane agentowi a_j . Zazwyczaj zadanie jest wyrażone za pomocą pojęć różnych od tych wytwarzanych przez receptory $R_{j,l}$ i wymaganych przez efekторы $E_{j,m}$. W tym celu wirtualne efekторы $e_{j,n}$ i wirtualne receptory $r_{j,k}$ odpowiednio transformują dane. Agent a_j za pośrednictwem swojego podsystemu sterowania c_j może wymieniać dane z podsystemami sterowania $c_{j'}$ innych agentów $a_{j'}$. Do tego celu wykorzystuje bufory wejściowe i wyjściowe ${}^T_{xc_{j,j'}}$ i ${}^T_{yc_{j,j'}}$.

Struktura podsystemów jest podobna, więc można skorzystać z ogólnego symbolu



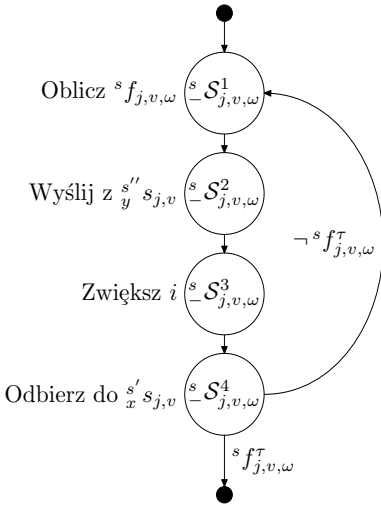
Rys. 1. Struktura agenta upostaciowionego

podsystemu $s_{j,v}$, gdzie $s \in \{c, r, e\}$. Podsystem $s_{j,v}$ uzyskuje dane od innych podsystemów s' poprzez swoje bufor wejściowe $s'_{x,j,v}$, wysyła dane do innych podsystemów s'' poprzez swoje bufor wyjściowe $s''_{y,j,v}$, posiada pamięć wewnętrzną $s^i_{j,v}$ i działa cyklicznie z czasem dyskretnym i , powodując zmianę zawartości buforów i pamięci z okresem próbkowania $\Delta i = 1$. Tak więc: $s', s'' \in \{c, r, e, E, R, T\}$, $s \neq s' \neq s''$.

Przetwarzaniem danych wprowadzonych do buforów wejściowych $s'_{x,j,v}$ w chwili i na dane wysyłane z buforów wyjściowych $s''_{y,j,v}$ w chwili $i+1$, zajmuje się funkcja przejścia $^s f_{j,v,\omega}$, gdzie ω jest jej nazwą. Ponadto funkcja ta wykorzystuje aktualną i tworzy nową zawartość pamięci wewnętrznej $s_{j,v}$.

$$[s''_{y,j,v}^{i+1}, s^i_{j,v}] := {}^s f_{j,v,\omega}(s'_{x,j,v}^i, s^i_{j,v}), \quad (1)$$

Zachowanie ${}^s \mathcal{B}_{j,v,\omega}$ iteracyjnie oblicza funkcję przejścia $^s f_{j,v,\omega}$, stanowiącą jego parametr, wysyła $s''_{y,j,v}$ do s'' , i pobiera nowe wartości wstawione przez s' do $s'_{x,j,v}$. Ten wzorzec zachowania podsystemu obrazuje graf automatu skończonego (rys. 2). Dalej automat skończony będzie nazywany w skrócie automatem. Stany ${}^s S_{j,v}^z$, $z = 1, \dots, 4$, automatu z rys. 2 wyróżniono indeksem '–', aby odróżnić je od stanów ${}^s S_{j,v}$ i superstanów ${}^+ S_{j,v}$ automatu hierarchicznego reprezentującego działanie podsystemu ${}^s s_{j,v}$. Iteracje zachowania ${}^s \mathcal{B}_{j,v,\omega}$ są kończone, gdy spełniony jest warunek końcowy reprezentowany predykatem ${}^s f_{j,v,\omega}^{\tau}$, który stanowi drugi parametr tego zachowania, czyli ${}^s \mathcal{B}_{j,v,\omega}({}^s f_{j,v,\omega}, {}^s f_{j,v,\omega}^{\tau})$. Gdy spełniony jest warunek ${}^s f_{j,v,\omega}^{\tau}$ podsystem



Rys. 2. Graf automatu realizującego zachowanie ${}^s\mathcal{B}_{j,v,\omega}$ podsystemu – wzorec

$s_{j,v}$ musi przełączyć się z zachowania ${}^s\mathcal{B}_{j,v,\omega}$ na inne zachowanie ${}^s\mathcal{B}_{j,v,\omega'}$. Tym zajmuje się hierarchiczny automat ${}^s\mathcal{F}_{j,v}$. Z każdym węzłem jego grafu skojarzone jest zachowanie ${}^s\mathcal{B}_{j,v,\omega}$ albo inny automat ${}^s\mathcal{F}_{j,v'}$. Przejścia między tymi węzłami uwarunkowane są warunkami początkowymi ${}^s f^{\sigma}_{j,v,\alpha}$. Warunki początkowe są predykatami ${}^s f^{\sigma}_{j,v,\alpha}(s'^i_{j,v}, s^i_{j,v})$. Formalną definicję wykorzystywanego tu hierarchicznego automatu skończonego typu Moore'a zamieszczono np. w [18, 19].

3. Przerywanie działań sytemu robotycznego

W systemach robotycznych istnieją trzy powody do przerywania aktualnie prowadzonych działań: 1) osiągnięcie celu realizowanego zachowania, 2) nowe pilniejsze polecenie od operatora, 3) wykrycie sytuacji awaryjnej. Wszystkie te sytuacje można wykryć za pomocą odpowiednio skonstruowanego warunku końcowego zachowania ${}^s f^{\tau}_{j,v,\omega}$. Zaletą tego rozwiązania jest szybkość reakcji, bo warunek końcowy sprawdzany jest w każdej iteracji zachowania, a więc z okresem próbkowania danego podsystemu $s_{j,v}$. Warunek końcowy ${}^s f^{\tau}_{j,v,\omega}$ zachowania ${}^s\mathcal{B}_{j,v,\omega}$ przyjmuje następującą postać:

$${}^s f^{\tau}_{j,v,\omega}(s'^i_{j,v}, s^i_{j,v}) = {}^s p_{j,v,\omega,\tau}(s'^i_{j,v}, s^i_{j,v}) \vee {}^s p_{j,v,\omega,l}(s'^i_{j,v}, s^i_{j,v}) \vee {}^s p_{j,v,\omega,\varepsilon}(s'^i_{j,v}, s^i_{j,v}) \quad (2)$$

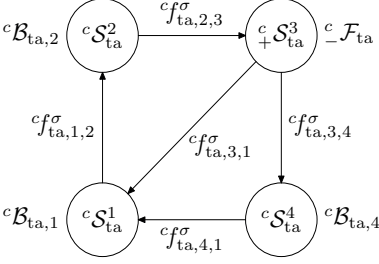
gdzie 1) predykat ${}^s p_{j,v,\omega,\tau}$ kończy zachowanie ${}^s\mathcal{B}_{j,v,\omega}$, gdy jego cel został osiągnięty i nie wykryto sytuacji nadzwyczajnej; 2) predykat ${}^s p_{j,v,\omega,l}$ informuje o pojawieniu się polecenia o wyższym priorytecie, wydanego przez operatora; 3) predykat ${}^s p_{j,v,\omega,\varepsilon}$ wykrywa sytuację awaryjną w systemie albo nietypową w środowisku. Wszystkie te predykaty korzystają z tych samych buforów wejściowych i pamięci wewnętrznej podsystemu, a więc mają dostęp do pełnej informacji. Jeżeli zachowanie ${}^s\mathcal{B}_{j,v,\omega}$ nie powinno być przerywane przez inne polecenie, to ${}^s p_{j,v,\omega,l} = \text{False}$. Polecenie przery-

4. System zakładający możliwość przerywania działań

Ponieważ każde działanie zmienia zarówno stan systemu, jak i środowiska, a ponad-



oraz stanie robota — dostarcza ją agent a_{km} . Agent a_{co} przekazuje elementy planu otrzymanego od agenta a_{pl} agentowi a_{ex} do wykonania.



Rys. 4. Graf automatu ${}^c\mathcal{F}_{ta}$ reprezentującego działanie agenta zadania a_{ta}

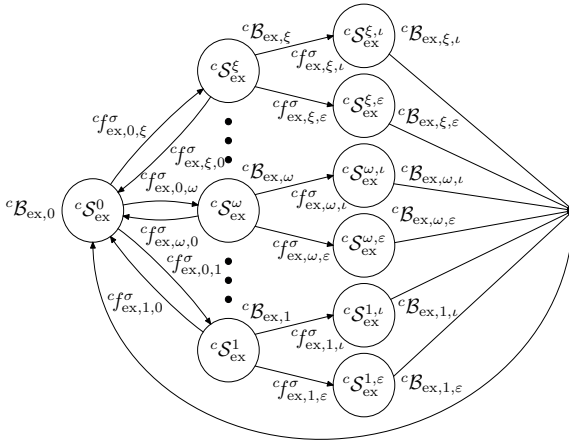
4.1. Agent zadania a_{ta} (task agent)

Automat ${}^c\mathcal{F}_{ta}$ sterujący działaniami agenta a_{ta} ma trzy stany: ${}^cS_{ta}^1$ (początkowy), ${}^cS_{ta}^2$, ${}^cS_{ta}^4$ oraz jeden superstan ${}^c_+\mathcal{S}_{ta}^3$ (rys. 4). Zachowanie ${}^c\mathcal{B}_{ta,1}$ wykonywane w ${}^cS_{ta}^1$ oczekuje na kolejne zadanie (plan). Pobierane jest w postaci automatu ${}^c\mathcal{F}_{ta}$ od a_{co} w stanie ${}^cS_{ta}^2$ przez zachowanie ${}^c\mathcal{B}_{ta,2}$. Po zintegrowaniu ${}^c\mathcal{F}_{ta}$ z ${}^c\mathcal{F}_{ta}$, ten ostatni przechodzi do superstanu ${}^c_+\mathcal{S}_{ta}^3$, inicjując tym samym wykonanie planu zapisanego jako ${}^c\mathcal{F}_{ta}$.

Warunki początkowe przyporządkowane łukom skierowanym ${}^c\mathcal{F}_{ta}$, czyli: ${}^cf_{ta,1,2}^\sigma$, ${}^cf_{ta,2,3}^\sigma$, ${}^cf_{ta,3,1}^\sigma$, ${}^cf_{ta,3,4}^\sigma$ i ${}^cf_{ta,4,1}^\sigma$, muszą spełniać warunki rozłączności oraz kompletności [18]. Rozłączność wymaga, aby żadne dwa predykaty przypisane łukom wychodzącym z danego węzła nie były jednocześnie prawdziwe, a warunek kompletności wymaga, aby jeden z tych predykatów był prawdziwy. Warunki te muszą być spełnione w chwili i , w której kończy się zachowanie powiązane z danym węzłem. Ponieważ z każdego stanu (rys. 4) wychodzi tylko jeden łuk skierowany, wszystkie muszą spełniać: ${}^cf_{ta,1,2}^\sigma = {}^cf_{ta,2,3}^\sigma = {}^cf_{ta,4,1}^\sigma = \text{True}$. Z superstanu ${}^c_+\mathcal{S}_{ta}^3$ wychodzą dwa łuki oznaczone: ${}^cf_{ta,3,1}^\sigma$ i ${}^cf_{ta,3,4}^\sigma$. ${}^cf_{ta,3,1}^\sigma$ powoduje wznowienie działania po normalnym zakończeniu zadania, natomiast ${}^cf_{ta,3,4}^\sigma$ prowadzi do wykonania zachowania ${}^c\mathcal{B}_{ta,4}$, które odpowiada za łagodne zakończenie przerwane zadania. Z (2) wynika, że ${}^cf_{ta,3,1}^\sigma = {}^sp_{j,v,\omega,\tau}(s_{j,v}^i, s_{j,v}^i)$, natomiast ${}^cf_{ta,3,4}^\sigma = {}^sp_{j,v,\omega,t}(s_{j,v}^i, s_{j,v}^i) \vee {}^sp_{j,v,\omega,\varepsilon}(s_{j,v}^i, s_{j,v}^i)$.

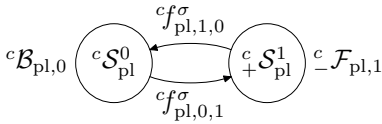
4.2. Agent wykonawczy a_{ex} (execution agent)

Każde zachowanie wchodzące w skład planu reprezentowanego automatem ${}^c\mathcal{F}_{ta}$ wysyła polecenia do agenta a_{ex} , a dokładniej do ${}^c\mathcal{F}_{ex}$, który zawiera stan decyzyjny ${}^cS_{ex}^0$, ze skojarzonym zachowaniem interpretującym otrzymane polecenie (rys. 5). Otrzymanie polecenia ω , $\omega = 1, \dots, \xi$, powoduje przejście ${}^c\mathcal{F}_{ex}$ do stanu ${}^cS_{ex}^\omega$, w którym zachowanie ${}^c\mathcal{B}_{ex,\omega}$ wykona to polecenie (dla uproszczenia przyjęto, że polecenie, stan i zachowanie noszą tę samą nazwę ω). Jeżeli w trakcie wykonywania zachowania ${}^c\mathcal{B}_{ex,\omega}$ pojawi się polecenie przerywające, co będzie wykryte przez warunek końcowy ${}^cf_{j,v,\omega}^\tau$ (2), to automat przejdzie do stanu ${}^cS_{ex}^{\omega,1}$ i rozpocznie wykonanie zachowania ${}^c\mathcal{B}_{ex,\omega,1}$, natomiast wykrycie sytuacji awaryjnej spowoduje przejście do stanu ${}^cS_{ex}^{\omega,\varepsilon}$.



Rys. 5. Graf automatu ${}^c\mathcal{F}_{\text{ex}}$ reprezentującego działanie agenta wykonawczego a_{ex}

i wykonanie zachowania ${}^c\mathcal{B}_{\text{ex},\omega,\epsilon}$. Zachowania ${}^c\mathcal{B}_{\text{ex},\omega,1}$ i ${}^c\mathcal{B}_{\text{ex},\omega,\epsilon}$ powodują łagodne zakończenie przerwanej zachowania ${}^c\mathcal{B}_{\text{ex},\omega}$ oraz informują agenta a_{ta} o zaistniałej sytuacji. Po zakończeniu tych zachowań automat ${}^c\mathcal{F}_{\text{ex}}$ wraca do stanu początkowego ${}^c\mathcal{S}_{\text{ex}}^0$ i oczekuje na decyzję a_{ta} . Należy przyjąć, że w przypadku pojawienia się kolejnych sytuacji nadzwyczajnych ${}^c\mathcal{B}_{\text{ex},\omega,1}$ i ${}^c\mathcal{B}_{\text{ex},\omega,\epsilon}$ jedynie informują agenta a_{ta} o awariach lub poleceniach przerywających, ale nie podejmują dodatkowych działań. Warunki początkowe przyjmują następującą postać: ${}^c f_{\text{ex},\omega,1}^{\sigma} = {}^s p_{\text{ta},\omega,1}$, ${}^c f_{\text{ex},\omega,\tau}^{\sigma} = {}^s p_{\text{ta},\omega,\tau}$ oraz ${}^c f_{\text{ex},\omega,\epsilon}^{\sigma} = {}^s p_{\text{ta},\omega,\epsilon}$, gdzie $\omega = 1, \dots, \xi$. Warunki początkowe ${}^c f_{\text{ex},0,\omega}^{\sigma}$ odpowiedzialne są za wskazanie zachowania, które wykona polecenie. Dla łuków skierowanych pozbawionych etykiet przyjmuje się, że ${}^c f_{\text{ex},\dots}^{\sigma} = \text{True}$.

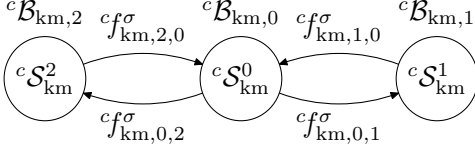


Rys. 6. Graf automatu ${}^c\mathcal{F}_{\text{pl}}$ reprezentującego działanie agenta planującego a_{pl}

4.3. Agent planujący a_{pl} (planning agent)

Agent planujący a_{pl} składa się jedynie z podsystemu sterowania c_{pl} reprezentowanego automatem ${}^c\mathcal{F}_{\text{pl}}$, który ma stan ${}^c\mathcal{S}_{\text{pl}}^0$ i superstan ${}^+_c\mathcal{S}_{\text{pl}}^1$, a ${}^c f_{\text{pl},1,0}^{\sigma} = {}^c f_{\text{pl},0,1}^{\sigma} = \text{True}$. (rys. 6). Zachowanie ${}^c\mathcal{B}_{\text{pl},0}$, wykonywane w stanie ${}^c\mathcal{S}_{\text{pl}}^0$, czeka na polecenia od a_{co} . Polecenie to dotyczy żądania planu dla wybranego przez a_{co} zadania. Polecenie powoduje przejście ${}^c\mathcal{F}_{\text{pl}}$ do superstanu ${}^+_c\mathcal{S}_{\text{pl}}^1$, w którym aktywowany jest automat ${}^c\mathcal{F}_{\text{pl},1}$, odpowiedzialny za stworzenie planu. Podczas planowania a_{pl} komunikuje się z a_{km} w celu uzyskania wiedzy o aktualnym stanie systemu i środowiska oraz o wiedzy zdroworozsądkowej. W wyniku planowania powstaje nie tylko wymagana sekwencja

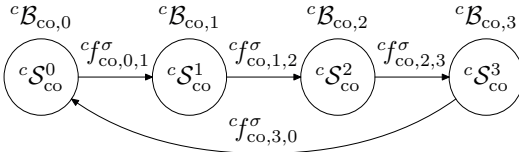
działań, ale także bezpieczne trajektorie ruchów. Stosowane są algorytmy TMP (Task Motion Planning), np. [4]. W [16] skorzystano ze struktury systemu przedstawionej na rys. 2 oraz hierarchicznych sieci zadań [12], a planowanie polegało na: 1) dekompozycji złożonych zadań na prostsze, co w rezultacie prowadzi do powstania akcji; 2) nadania wartości parametrom zadań; i 3) wykonania akcji, co powoduje zmiany stanu. Wykonanie akcji wymaga upewnienia się, że ruch robota będzie bezkolizyjny. W tym celu skorzystano z generatora PDDLStream [10]. Plan jest tworzony w formie grafu $_F_{ta}$. Akcje reprezentowane są jako zachowania. Plan w tej postaci przekazywany jest agentowi a_{ta} . Planowania można zrealizować też innymi metodami, np. [1].



Rys. 7. Graf automatu $_F_{km}$ zarządzającego bazą wiedzy (a_{km})

4.4. Agent zarządzający bazą wiedzy a_{km} (knowledge management agent)

Agent zarządzający bazą wiedzy a_{km} (rys. 7) odpowiada na żądania agenta planującego a_{pl} , dostarczając mu niezbędnej wiedzy, oraz zdobywa informacje o agenta wykonawczego a_{ex} o zachodzących zmianach w środowisku i stanie sprzętu. W związku z tym automat $_F_{km}$ w stanie $_cS^0_{km}$ oczekuje albo na żądania a_{pl} albo na nowe informacje od a_{ex} . Zależnie od tego, który z tych agentów się zgłosi $_cF_{km}$ przechodzi albo do stanu $_cS^1_{km}$ albo do $_cS^2_{km}$, wykonując albo zachowanie $_cB_{km,1}$ dostarczające wiedzę a_{pl} albo zachowanie $_cB_{km,2}$ pobierające informacje od a_{ex} . $_cf^{\sigma}_{km,1,0} = _cf^{\sigma}_{km,2,0} = \text{True}$.



Rys. 8. Graf automatu $_F_{co}$ reprezentującego działanie agenta komunikacyjnego a_{co}

4.5. Agent komunikacyjny a_{co} (communication agent)

Agent komunikacyjny a_{co} (rys. 8) umożliwia interakcję systemu z operatorem oraz harmonogramowanie zadań zleczanych do wykonania. W stanie $_cS^0_{co}$ wykonuje zachowanie $_cB_{co,0}$, które oczekuje na polecenie operatora bądź na informację od agenta a_{ta} wskazującą, że powstała sytuacja nadzwyczajna. Nowe polecenie bądź wykrzykie sytuacji nadzwyczajnej powoduje przejście do stanu $_cS^1_{co}$ i rozpoczęcie wykonania procedury harmonogramowania przez $_cB_{co,1}$. Wybranie zadania do wykonania powoduje przejście do stanu $_cS^2_{co}$, w którym $_cB_{co,2}$ żąda od agenta a_{pl} dostarczenia planu jego realizacji. Plan przekazywany jest w formie automatu $_F_{ta}$. W stanie $_cS^3_{co}$ zacho-

wanie ${}^c\mathcal{B}_{co,3}$ przekazuje plan agentowi a_{ta} do wykonania. Łuki ${}^c\mathcal{F}_{co}$: ${}^cf_{co,0,1}^\sigma = {}^cf_{co,1,2}^\sigma = {}^cf_{co,2,3}^\sigma = {}^cf_{co,3,4}^\sigma = \text{True}$. Najbardziej złożonym zachowaniem jest ${}^c\mathcal{B}_{co,1}$, które realizuje harmonogramowanie zadań. Można to zrobić podobnie jak w [7, 5], ale wyzwalanie zegarem nie jest potrzebne, bo tutaj harmonogramowanie uaktywniane jest przez nadchodzące polecenia lub sytuacje nadzwyczajne. Przypisanie priorytetów do zadań może być zastąpione planowaniem przez a_{pl} .

5. Wnioski

Biorąc pod uwagę złożoność problemu związanego z przerywaniem realizowanych przez robota zadań, zastosowano względnie prostą strukturę systemu składającą się z pięciu agentów (rys. 3). Dla każdego z nich określono hierarchiczny automat skończony reprezentujący działania jego podsystemu sterowania. Przerwania są inicjowane albo przez agenta wykonawczego a_{ex} (sytuacje nadzwyczajne) albo agenta komunikacyjnego a_{co} (polecenia operatora). Postać warunku końcowego (2) umożliwia zachowanie standardowej postaci zachowania (rys. 2), a co więcej gwarantuje najszybszą reakcję na sytuację nadzwyczajną. Postuluje się wybór zadania spośród zestawu zadań oczekujących na wykonanie opierający się na aktualnym stanie środowiska i planowaniu. Przyszłe prace będą dotyczyć adaptacji algorytmu planowania opisanego w [16], aby dostarczyć niezbędne informacje do harmonogramowania zadań. Następnie algorytm ten zostanie zintegrowany z dwurękiem robotem [17], wyposażonym w ruchomy tułów oraz ruchomą szyję, na której zamontowano kamery [13].

Literatura

- [1] M. Beetz, R. Chatila, J. Hertzberg, and F. Pecora, *The Handbook of Robotics*. Springer, 2016, ch. AI Reasoning Methods for Robotics, pp. 329–356.
- [2] R. A. Brooks, “Intelligence without reason,” *Artificial intelligence: critical concepts*, vol. 3, pp. 107–163, 1991.
- [3] R. Brooks, “Elephants don’t play chess,” *Robotics and autonomous systems*, vol. 6, no. 1–2, pp. 3–15, 1990.
- [4] N. Castaman, E. Pagello, E. Menegatti, and A. Pretto, “Receding horizon task and motion planning in changing environments,” *Robotics and Autonomous Systems*, p. 103863, 2021.
- [5] W. Dudek, *Prudent Management of Interruptible Tasks Executed by a Service Robot*. Politechnika Warszawska, Rozprawa doktorska, 2022.
- [6] W. Dudek, M. Węgierek, J. Karwowski, W. Szykiewicz, and T. Winiarski, “Task harmonisation for a single-task robot controller,” in *12th Int. Workshop on Robot Motion and Control (RoMoCo)*, K. Kozłowski, Ed. IEEE, 2019, pp. 86–91.
- [7] W. Dudek and T. Winiarski, “Scheduling of a robot’s tasks with the tasker framework,” *IEEE Access*, vol. 8, pp. 161 449–161 471, 2020.
- [8] M. Figat and C. Zieliński, “Parameterised robotic system meta-model expressed by Hierarchical Petri nets,” *Robotics and Autonomous Systems*, vol. 150, pp. 103987, April 2022.
- [9] M. Figat and C. Zieliński, “Synthesis of robotic system controllers using robotic system specification language,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 688–695, 2023.

- [10] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, “Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning,” in *Proceedings of the Int. Conf. on Automated Planning and Scheduling*, vol. 30, 2020, pp. 440–448.
- [11] B. Gerkey and M. J. Matarić, “A formal analysis and taxonomy of task allocation in multi-robot systems,” *Int. J. of Robotics Research*, vol. 23, no. 9, pp. 939–954, September 2004.
- [12] L. P. Kaelbling and T. Lozano-Pérez, “Hierarchical task and motion planning in the now,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 1470–1477.
- [13] R. Nebeluk, K. Zarzycki, D. Seredyński, P. Chaber, M. Figat, P. Domański, and C. Zieliński, “Predictive tracking of an object by a pan-tilt camera of a robot,” *Nonlinear Dynamics*, vol. 111, p. 8383–8395, 2023.
- [14] P. Pałka, C. Zieliński, W. Dudek, D. Seredyński, and W. Szykiewicz, “Communication-focused top-down design of robotic systems based on binary decomposition,” *MDPI Energies*, vol. 15, no. 21, article no 7983, pp. 1187–1205, 2022.
- [15] L. Parker, D. Rus, and S. G. Sukhatme, “Multiple mobile robot systems,” in *Springer Handbook of Robotics*, O. Khatib and B. Siciliano, Eds. Springer, June 2016, pp. 1335–1384.
- [16] D. Seredynski, *Planning and Realization of Service Robot Manipulation Tasks at Multiple Levels of Abstraction*. Politechnika Warszawska, Rozprawa doktorska, 2023.
- [17] D. Seredyński, T. Winiarski, and C. Zieliński, “FABRIC: Framework for Agent-Based Robot Control Systems,” in *12th Int. Workshop on Robot Motion and Control (RoMoCo’2019)*, IEEE Xplore, 2019, pp. 215–222.
- [18] C. Zieliński, “Specification of agent based robotic systems using hierarchical finite state automata,” in *Advanced, Contemporary Control*, A. Bartoszewicz, J. Kabziński, and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2020, pp. 465–476.
- [19] C. Zieliński, ser. *Advances in Intelligent Systems and Computing*. Springer, 2021, vol. 296, rozdz. “Robotic System Design Methodology Utilising Embodied Agents”, pp. 523–561.
- [20] C. Zieliński, “Interrupting Companion Robot Activities,” in *13th Int. Workshop on Robot, Motion and Control (RoMoCo’2024)*, Poznań, Poland, July 2–4, 2024. IEEE Xplore 979-8-3503-9396-5/24/\$31.00, pp. 49–53.
- [21] C. Zieliński, M. Figat, and R. Hexel, “Communication within multi-FSM based robotic systems,” *Journal of Intelligent & Robotic Systems*, June 2018.
- [22] C. Zieliński i inni, “Variable structure robot control systems: The RAPP approach,” *Robotics and Autonomous Systems*, vol. 94, pp. 226–244, 2017.

Interruption of robot activities

The paper presents how to organise a robotic system able to cope with interruption of its current activities. Interruption is either due to a new request or an abnormal situation. Moreover, the remedy for the interrupt might not be completed when a new interrupt arrives, and so on. A formal specification of a controller solving the thus posed problem is suggested.

Sterowanie dwuosiowym manipulatorem maksymalizujące siłę sygnału w systemie komunikacji laserowej*

Patryk Bartkowiak¹, Dariusz Pazderski¹

Streszczenie

Praca dotyczy sterowania ekstremalnego dla jednostki pozycjonującej, w celu maksymalizacji intensywności odbieranego sygnału w systemie komunikacji laserowej. Zakłada się, że położenie wiązki laserowej nie jest mierzone bezpośrednio. Kluczową ideą proponowanego mechanizmu sprzężenia zwrotnego jest pośrednie wykorzystanie gradientu funkcji intensywności wiązki w otoczeniu sensora. Gradient ten jest aproksymowany za pomocą metody uśredniania wzdłuż trajektorii cykloidalnej. W referacie przedstawiono wyniki eksperymentów uzyskane przy użyciu dwuosiowego manipulatora – montażu astronomicznego i zestawu urządzeń do komunikacji laserowej. Wyniki badań demonstrują skuteczność proponowanego algorytmu sterowania i potwierdzają wysoką dokładność pozycjonowania w scenariuszach czasu rzeczywistego.

1. Wstęp

Ostatnio obserwuje się wzrost znaczenia systemów komunikacji laserowej [1, 2, 3], które mogą zapewniać większą przepustowość i niższe opóźnienia transmisji w porównaniu do tradycyjnych systemów wykorzystujących fale radiowe [4, 5]. Technologia laserowa umożliwia przesyłanie dużych ilości danych, w tym plików multimedialnych i danych pomiarowych, z większą efektywnością i niezawodnością. Systemy komunikacji laserowej są szczególnie cenne w misjach kosmicznych, gdzie konieczna jest komunikacja o wysokiej przepustowości między statkami kosmicznymi, satelitami i stacjami naziemnymi [6, 7]. Znajdują także zastosowania wojskowe, zapewniając bezpieczne i odporne na zakłócenia kanały komunikacyjne, niezbędne do wymiany danych w czasie rzeczywistym [8, 9].

W wielu przypadkach poprawne działanie laserowych systemów transmisyjnych zależy od dokładnego pozycjonowania urządzeń nadawczo-odbiorczych, w celu optymalnego nakierowania wiązki laserowej w obszar aktywny detektora. Pozwala to maksymalizować stosunek sygnału do szumu, poprawiając jakość detekcji i niezawodność transmisji, w tym jej ciągłości oraz ograniczenie błędów. Stąd m.in. wynikają badania nad metodami precyzyjnego pozycjonowania wiązki laserowej. Zazwyczaj zadanie to realizuje się przy użyciu kamer lub detektorów czterokwadrantowych [10], co umożliwia uzyskanie dużej dynamiki odpowiedzi układu zamkniętego. Bardziej

*Praca finansowana ze środków subwencji badawczej 0211/SBAD/0124 oraz z Programu „PhDBoost” dla doktorantów Szkoły Doktorskiej Politechniki Poznańskiej (w roku 2024) z subwencji Uczelni, pochodzącej ze środków Ministerstwa Nauki i Szkolnictwa Wyższego.

¹Instytut Automatyki i Robotyki, Politechnika Poznańska, ul. Piotrowo 3A, Poznań, patryk.bartkowiak@put.poznan.pl, dariusz.pazderski@put.poznan.pl

złożony jest przypadek, w którym współrzędne wiązki nie są mierzone bezpośrednio, a dostępna jest tylko informacja o jej natężeniu.

Niniejsza praca rozważa problem sterowania ekstremalnego w kontekście pozycjonowania urządzenia nadawczego w celu maksymalizacji sygnału odbieranego przez detektor. Proponowany algorytm opiera się na technice gradientowej, która jest fundamentalnym narzędziem w zadaniach optymalizacji i sterowania, wykorzystywanym powszechnie w systemach automatyki, robotyce i technologiach kosmicznych [11, 12].

W rozważanym zagadnieniu gradient funkcji natężenia sygnału nie jest jawnie określony – do jego rekonstrukcji wykorzystano metodę uśredniania. Zaproponowany algorytm sterowania dwuosowego manipulatora jest sformułowany w dziedzinie czasu ciągłego i wykorzystuje periodyczną trajektorię, umożliwiającą aproksymację gradientu w otoczeniu punktu w przestrzeni konfiguracyjnej.

Głównym wkładem niniejszej pracy jest przedstawienie formalnego opisu algorytmu sterowania opartego na sprzężeniu zwrotnym oraz wyników jego praktycznej implementacji w warunkach laboratoryjnych. W celu przeprowadzenia badań eksperymentalnych przygotowano dedykowane stanowisko. Wykorzystuje ono m.in. układ sterowania i nadzoru montażu astronomicznego, opracowany i zbudowany w Instytucie Automatyki i Robotyki Politechniki Poznańskiej [13, 14].

2. Algorytm sterowania

Rozważmy system transmisyjny przedstawiony na rys. 1, w którym detektor odbiera plamkę wiązki laserowej wysyłaną przez nadajnik. Natężenie odbieranej wiązki, opisane przez nieujemną funkcję $\mu \geq 0$, zależy od położenia plamki względem detektora. Maksymalne natężenie osiągane jest w przypadku, gdy środek wiązki pokrywa się ze środkiem detektora. Uzyskanie takiego ustawienia pomiędzy nadajnikiem a odbiornikiem stanowi główne zadanie sterowania, które jest rozważane w tej pracy.

Zakładamy, że orientacja nadajnika może być zmieniana w dwóch osiach, a jego konfigurację można opisać jako: $q = [q_1 \ q_2]^T \in Q = S^1 \times S^1$. Kinematyka układu pozycjonującego dana jest następującym równaniem:

$$\dot{q} = v, \quad (1)$$

gdzie $v \in \mathbb{R}^2$ jest wejściem sterującym. Główny problem sterowania rozważany w tej pracy można formułować następująco.

Problem 1. *Należy znaleźć prawo sterowania $v = v(\cdot)$, które zapewnia spełnienie zależności granicznej*

$$\lim_{t \rightarrow \infty} q(t) = q_m, \quad (2)$$

przy czym $q_m \in Q$ oznacza nieznaną a priori konfigurację, dla której natężenie μ osiąga maksymalną wartość μ_m .

Z perspektywy rozwiązania tego zadania sterowania kluczowe jest rozróżnienie dwóch przypadków uwzględniających możliwość detekcji wiązki. W tym celu wprowadźmy zbiór Ω w Q , dla którego odbiór wiązki przez odbiornik jest prawidłowy,

$$\Omega = \{q \in Q: \mu \geq \bar{\mu}\}, \quad (3)$$

gdzie $\bar{\mu} > 0$ jest pewnym minimalnym progiem natężenia sygnału, umożliwiającym detekcję. W przypadku, gdy $q \in Q \setminus \Omega$, wiązka laserowa znajduje się poza obszarem detekcji odbiornika, co oznacza, że natężenie μ jest bliskie zeru. Wówczas wymagane jest uzyskanie tranzycji z punktu $q \in Q \setminus \Omega$ do pewnego punktu $q \in \Omega$, dla którego wiązka laserowa oświetla sensor odbiornika. W tym drugim przypadku można zaprojektować sprzężenie zwrotne od wyjścia μ w celu poszukiwania maksymalnej wartości natężenia. Oba przypadki są rozważane kolejno w punktach 2.1 i 2.2.

2.1. Sterowanie w pętli otwartej podczas fazy pokrywania obszaru

Uwzględniając topologię przestrzeni konfiguracyjnej wynikającą z obrotów wokół niezależnych osi, proponujemy zastosowanie prostej strategii pokrywania przestrzeni konfiguracyjnej za pomocą następującej trajektorii spiralnej:

$$q_o(t) = rt \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix} + q_o^0, \quad (4)$$

gdzie $q_o^0 \in Q$ jest przesunięciem, $r \in \mathbb{R}_+$ oraz $\omega \in \mathbb{R} \setminus \{0\}$ są parametrami, kolejno odpowiadającymi amplitudzie oraz pulsacji. Sygnał sterujący v w równaniu (1) można łatwo wyznaczyć, różniczkując po czasie wyrażenie (4). Faza pokrywania obszaru trwa dopóki natężenie $\mu < \bar{\mu}$, co oznacza, że $q \in Q \setminus \Omega$.

2.2. Sterowanie w pętli zamkniętej podczas fazy poszukiwania maksimum

Dla uproszczenia zakładamy, że funkcja μ nie ma minimów lokalnych, a jedynym punktem krytycznym jest maksimum. Możliwe jest wówczas poszukiwanie maksimum tej funkcji na podstawie gradientu $\nabla_q \mu = \frac{\partial \mu}{\partial q}$. Określmy pole wektorowe na Ω , które lokalnie opisuje prędkość w dowolnym punkcie $q \in \Omega$ według następującej formuły:

$$v = k(\nabla_q \mu)^\top, \quad (5)$$

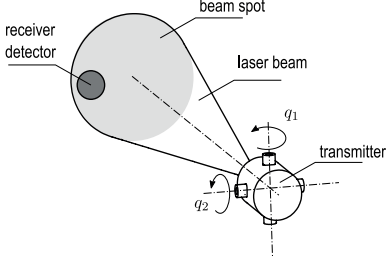
gdzie $k > 0$ jest wzmocnieniem. Równanie (5) opisuje sprzężenie od stanu, które zapewnia osiągnięcie celu sterowania (2). Należy jednak zwrócić uwagę, że podana formuła nie może być bezpośrednio zastosowana, gdy tak jak w rozpatrywanym przypadku, jawna postać gradientu nie jest znana. Dlatego istotne jest określenie metody, pozwalającej wyznaczać gradient w trakcie procesu sterowania.

2.3. Estymacja gradientu

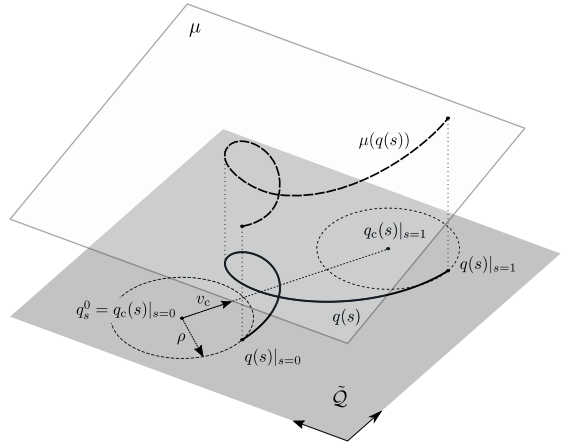
Zakładamy, że jedyną dostępną informacją jest wartość funkcji μ w punkcie q i na tej podstawie należy odtworzyć gradient $\nabla_q \mu$. Aby wprowadzić metodę obliczania gradientu, rozważmy najpierw przypadek szczególny, w którym μ jest funkcją liniową – z definicji pozbawioną ekstremów.

Na podzbiorze $\tilde{Q} \subset \mathbb{R}^2$ określamy funkcję $\mu: \tilde{Q} \rightarrow \mathbb{R}$, która spełnia: $\forall q \in \tilde{Q}, \mu = g^\top q$, gdzie $g \in \mathbb{R}^2$ oznacza stałe parametry. Niech $q_s(s) \in \tilde{Q}$ będzie cykloidą spełniającą równanie

$$q_s(s) = v_c s + \rho \begin{bmatrix} \cos(2\pi s) \\ \sin(2\pi s) \end{bmatrix} + q_s^0, \quad (6)$$



Rys. 1. Natężenie wiązki laserowej odbieranej przez detektor



Rys. 2. Ilustracja graficzna metody wyznaczania gradientu rozważanej w artykule

gdzie $v_c \in \mathbb{R}^2$ i $q_s^0 \in \tilde{Q}$, $\rho \in \mathbb{R}_+$ są stałymi, a $s \in \mathbb{R}$ oznacza zmienną parametryzującą.

Propozycja 1. Gradient $\nabla_q \mu$ na $\tilde{Q}$ można obliczyć wzdłuż krzywej (6) dla $s \in [0, 1]$ według następującej zależności

$$\nabla_q \mu = (\pi \rho)^{-1} \int_0^1 \mu'(s) [-\sin(2\pi s) \cos(2\pi s)] ds, \quad (7)$$

gdzie $\mu'(s) = \frac{\partial \mu}{\partial s}$.

Dowód: Obliczając pochodną $\mu'(s)$ zapisujemy: $\frac{\partial \mu}{\partial s} = \frac{\partial \mu}{\partial q} \frac{\partial q}{\partial s}$. Wyrażenie $\frac{\partial q}{\partial s}$ obliczamy na podstawie (6) uwzględniając, że $q = q_s$ i otrzymujemy

$$q'(s) = v_c + 2\pi \rho \begin{bmatrix} -\sin(2\pi s) \\ \cos(2\pi s) \end{bmatrix}. \quad (8)$$

Dla założonego liniowego modelu μ zachodzi: $\nabla_q \mu = \frac{\partial \mu}{\partial q} = g^\top$. W związku z tym formułę (7) można przepisać jako

$$\nabla_q \mu = (\pi \rho)^{-1} g^\top \int_0^1 (v_c + 2\pi \rho w(s)) w^\top(s) ds = (\pi \rho)^{-1} g^\top \chi, \quad (9)$$

gdzie $w^\top(s) = [-\sin(2\pi s) \cos(2\pi s)]$. Wyznaczając całkę w równaniu (9), otrzymamy $\chi = \rho \pi I$, gdzie I to macierz jednostkowa. W ten sposób można wykazać, że

$$\nabla_q \mu = (\pi \rho)^{-1} g^\top \rho \pi = g^\top. \quad (10)$$

■

Proponowaną metodę można zinterpretować geometrycznie, jak pokazano na rys. 2. Trajektoria odpowiadająca krzywej (6) wynika ze złożenia ruchu po okręgu o promieniu ρ oraz ruchu liniowego o stałej prędkości v_c środka okręgu, o współrzędnych q_c . W ruchomym układzie współrzędnych przywiązanym do środka okręgu, gradient $\nabla_q \mu$ jest rekonstruowany w trakcie pełnego obrotu na okręgu.

Należy podkreślić, że w ogólnym przypadku funkcja μ jest nieliniowa, co stanowi warunek konieczny istnienia maksimum, a v_c jest zmienne w czasie. Wówczas proponowana metoda obliczania gradientu może być postrzegana jako metoda aproksymująca gradient, przy założeniu, że promień ρ jest wystarczająco mały, a prędkość v_c nie zmienia się zbyt szybko, tzn. norma $|v'_c(s)|$ nie jest znacząca. Na podstawie (9) można zatem rozważyć następujące przybliżenie gradientu $\nabla_q \mu$ na $\tilde{Q}$:

$$\tilde{\nabla}_q \mu = (\pi \rho)^{-1} \int_0^1 \mu'(s) w^\top(s) ds. \quad (11)$$

2.4. Projektowanie sprzężenia zwrotnego

Wracając do projektowania sprzężenia zwrotnego od stanu na podstawie nominalnej zależności (5), czynnik parametryzujący zmienną s uzależniamy od czasu, redefiniując $s = vt$, gdzie v jest częstotliwością oscylacji. W związku z tym, obliczając prędkość $\dot{q}_s$ na podstawie (8), otrzymujemy: $\dot{q}_s = v v_c + 2\pi v \rho w(vt)$. Podobnie, gradient można oszacować w sposób ciągły na podstawie (11) jako

$$\tilde{\nabla}_q \mu(t) = (\pi \rho)^{-1} \int_{vt-1}^{vt} \mu'(s) w^\top(s) ds. \quad (12)$$

Następnie przywołując sprzężenie zwrotne (5), można sformułować następującą propozycję.

Propozycja 2. *Prawo sterowania*

$$v(t) = v v_c(t) + 2\pi v \rho w^\top(vt), \quad (13)$$

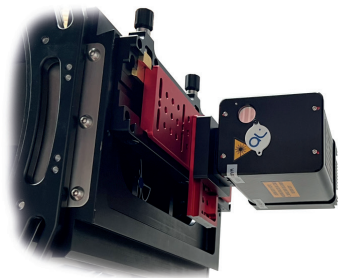
$$v_c(t) = k \tilde{\nabla}_q \mu(t), \quad (14)$$

gdzie $k \in (0, \bar{k})$ i $\rho \in (0, \rho_m > 0)$, $v > \underline{v} > 0$ są współczynnikami, $\bar{k}$, $\bar{\rho}$ i $\underline{v}$ są pewnymi stałymi, zapewnia, że dla $q(0) \in \Omega$, $\lim_{t \rightarrow \infty} q_c(t) = q_m$, przy czym q_c oznacza pozycję środka wirtualnego okręgu o promieniu ρ .

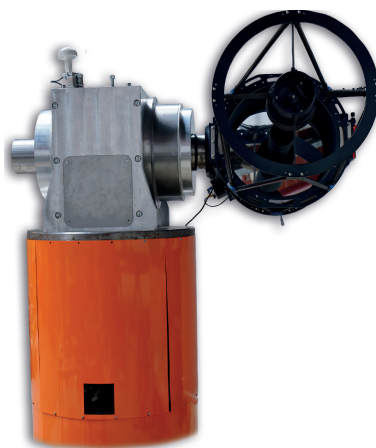
Z punktu widzenia zastosowań istotne jest ograniczenie wartości prędkości v_c , co można uzyskać wprowadzając następującą procedurę skalowania:

$$v_c := \left(\max\{1, |v_{c1} \bar{v}_c^{-1}|, |v_{c2} \bar{v}_c^{-1}|\} \right)^{-1} v_c, \quad (15)$$

gdzie $\bar{v}_c > 0$ oznacza założoną maksymalną bezwzględną wartość składowych v_c .



Rys. 3. Nadajnik i odbiornik zestawu do komunikacji laserowej *AstroLight*



Rys. 4. Montaż astronomiczny *MDSO2* opracowany w Instytucie Automatyki i Robotyki (Politechnika Poznańska)

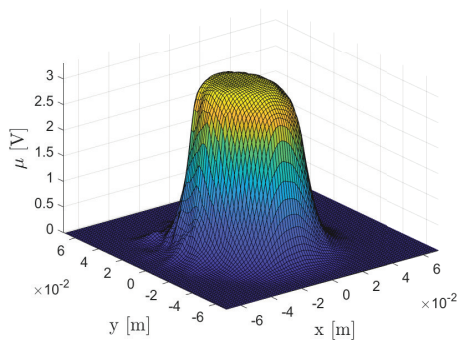
3. Eksperyment

3.1. Konfiguracja sprzętowo-programowa

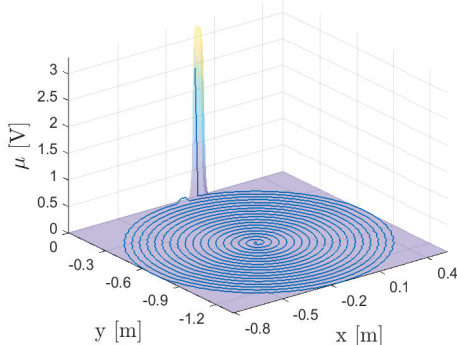
W badaniach eksperymentalnych zastosowano zestaw do komunikacji laserowej *AstroLight*, który umożliwia transmisję jednokierunkową z przepustowością 10 Mbps przy użyciu lasera emitującego falę w pasmie podczerwieni o długości $1,5\mu\text{m}$. Odbiornik został zamontowany nieruchomo na statywie, natomiast nadajnik przymocowano do osi poziomej horyzontalnego montażu astronomicznego – por. rys. 3 i 4. Odbiornik i nadajnik zostały umieszczone w odległości $d = 13\text{ m}$, uwzględniając dostępną przestrzeń w laboratorium. Aby ułatwić interpretację geometryczną, można zauważyć, że z uwagi na niewielkie odchylenia kątowe, trajektoria kołowa o promieniu p w przestrzeni konfiguracyjnej odpowiada w przestrzeni zadania trajektorii kołowej o promieniu $a = pd$ rzutowanej na płaszczyznę detektora odbiornika.

Algorytm opisany w punkcie 2 został zaimplementowany pośrednio, wykorzystując strukturę kaskadową układu sterowania zaprojektowaną dla użytego montażu astronomicznego – tj. sterowanie na poziomie prędkości zastąpione zostało sterowaniem pozycyjnym. W tym celu na bieżąco obliczana była trajektoria referencyjna na podstawie wzorów (4) oraz (13)-(14), a jej precyzyjne śledzenie zapewniał sterownik ruchu niskiego poziomu, wykorzystujący strukturę ADRC [15].

Logika sterownika wysokiego poziomu została zintegrowana z aplikacją internetową *Robot Supervisor Panel* (RSP), zaprojektowaną w celu wspomagania projektowania, testowania i utrzymania optycznych systemów obserwacyjnych, przy użyciu standardowych narzędzi programowania rozwiązań internetowych. Aplikacja wykorzystuje interfejs *Web Serial API* do ustanawiania komunikacji z urządzeniem pomiarowym, pozwalając na odczyt napięcia odpowiadającego intensywności wiązki laserowej μ padającej na sensor. Urządzeniem do pomiaru napięcia był multimetr *Fluke* 289, który dostarczał dane z częstotliwością około 15 Hz.



Rys. 5. Rozkład intensywności wiązki określony przy użyciu trajektorii spiralnej dla $r = 1,54e-5$ rad/s, $\omega = 0,3$ rad/s.



Rys. 6. Trajektoria spiralna w fazie pokrywania obszaru dla $r = 3,84e-5$ rad/s i $\omega = 0,5$ rad/s

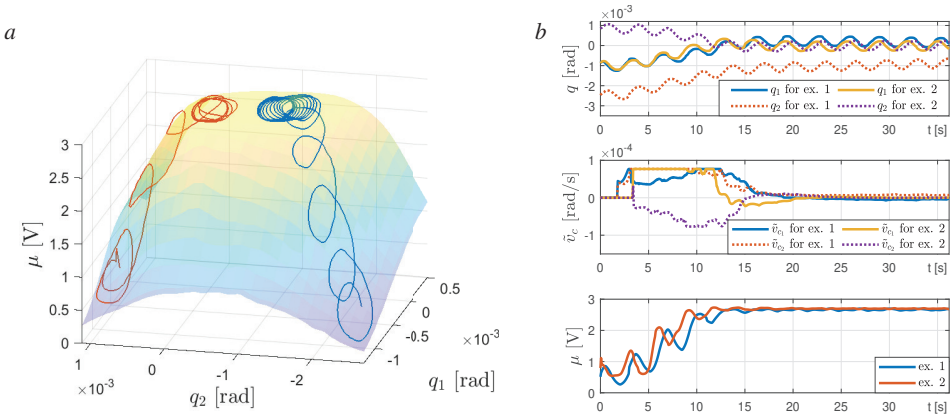
3.2. Wyniki

Wstępny eksperyment polegał na identyfikacji funkcji intensywności sygnału laserowego poprzez rejestrację napięcia dla ruchu spiralnego (4). Jego wynik przedstawia rys. 5. Analizując otrzymany wykres można zauważyć, że dla w badanym scenariuszu wyjściowe napięcie odbiornika osiąga nasycenie, co objawia się pewnym wypłaszczeniem funkcji w obszarze większej intensywności, a w rezultacie brakiem wyraźnego maksimum. Trajektoria (4) została również zastosowana w fazie pokrywania algorytmu pozycjonowania, choć z innymi wartościami parametrów. Przykładowe wyniki tej fazy przedstawiono na rys. 6.

Aby ułatwić interpretację uzyskanych danych, odpowiednio przesunięto wartości konfiguracji q , aby zapewnić, że dla $q_m = 0$ osiągnięta zostanie maksymalna intensywność $\mu = \mu_m$. Faza pokrywania przestrzeni kończy się, gdy zmierzony sygnał napięcia przekroczy wartość progową $\bar{\mu}$, tzn. q osiąga Ω ; następnie uruchamiana jest kolejna faza algorytmu – poszukiwania maksimum. Dla wszystkich przedstawionych eksperymentów próg przełączenia fazy algorytmu był równy $\bar{\mu} = 0,3$ V.

W kolejnych przypadkach badano właściwości sterowania z sprzężeniem zwrotnym. We wszystkich przedstawionych eksperymentach wartość wzmocnienia i nasycenie sygnału wejściowego ustawiono odpowiednio: $k = 1e-7$ oraz $\bar{v}_c = 7,69e-5$ rad/s. Na rysunkach 7a oraz 7b przedstawiono wyniki uzyskane dla dwóch różnych warunków początkowych. Pierwszy rysunek to wykres trójwymiarowy, który przedstawia ścieżkę $q \in Q$ nałożoną na wykres powierzchniowy ilustrujący (zidentyfikowany wcześniej) rozkład intensywności wiązki laserowej. Drugi rysunek pokazuje przebiegi konfiguracji q , sygnału sterującego v_c i intensywności μ . W tym przypadku użyto następujących wartości parametrów: $\rho = 2,3e-4$ rad ($a = 3$ mm w przestrzeni zadania) oraz $v = 0,2$ Hz. Jak można zaobserwować, dla obu stanów początkowych $q(0)$ funkcja intensywności μ zbliża się do wartości μ_m . Z powodu nasycenia sygnału μ w pobliżu szczytu rozkładu intensywności wiązki laserowej, wartość gradientu maleje, co prowadzi do małych wartości sygnału sterującego.

Dodatkowo przeprowadzono eksperymenty dla dwóch różnych promieni trajektorii cykloidalnej $\rho \in \{3,84e-4; 7,69e-5\}$ rad ($a \in \{5; 1\}$ mm) oraz stałej wartości $v = 0,1$ Hz, aby zweryfikować jak promień okręgu wpływa na jakość sterowania ze



Rys. 7. Sterowanie podczas fazy poszukiwania maksimum dla $\rho = 2,3e-4$ rad, $\nu = 0,2$ Hz
a – ścieżki w przestrzeni konfiguracyjnej, b – wykresy czasowe konfiguracji q , sygnału wejściowego v_c oraz intensywności wiązki laserowej μ

sprężeniem zwrotnym. Uzyskane wyniki przedstawiono na wykresach trójwymiarowych 8a oraz 8b. Badano również wpływ częstotliwości ν na działanie algorytmu. W tym celu uzyskano wyniki dla $\rho = 3,84e-4$ rad (5 mm) oraz zwiększonej wartości $\nu = 0,5$ Hz, które przedstawiono na rysunku 8c. Kolejny eksperyment przeprowadzono w celu sprawdzenia zdolności algorytmu sterowania do śledzenia ruchomego odbiornika. W tym scenariuszu, po osiągnięciu wartości μ bliskiej μ_m , odbiornik został ręcznie przesunięty, co wpłynęło na zmiany w pozycji docelowej głównie dla q_1 (które w przybliżeniu odpowiada współrzędnej x w przestrzeni zadania). Wyniki uzyskane dla $\rho = 2,3e-4$ rad ($a = 3$ mm), $\nu = 0,2$ Hz zostały przedstawione na rys. 8d, które pozwalają potwierdzić poprawne zachowanie układu sterowania dla zadania śledzenia w tym scenariuszu.

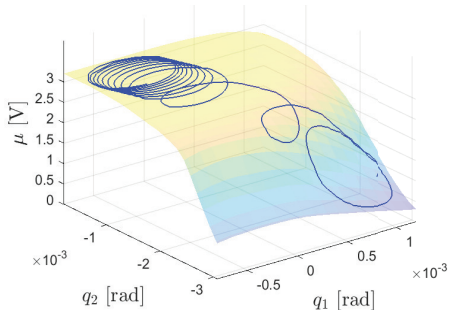
4. Podsumowanie

W referacie zaproponowano algorytm sterowania ekstremalnego, mający na celu optymalizację orientacji nadajnika laserowego w celu maksymalizacji współczynnika sygnału do szumu, zakładając brak dostępu do pomiaru pozycji wiązki. Algorytm wykorzystuje etap pokrywania przestrzeni oraz sterowanie z sprzężeniem zwrotnym oparte na ciągłych przybliżeniach gradientu w dwuwymiarowej przestrzeni konfiguracyjnej. Formalnie pokazano, że algorytm zapewnia zbieżność konfiguracji systemu do zbioru, w którym intensywność odbieranego sygnału jest bliska maksymalnej.

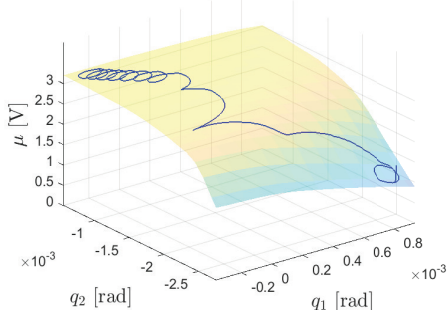
Badania eksperymentalne potwierdziły, że metoda może być skutecznie zaimplementowana i zapewnia wszystkie zakładane cechy w szerokim zakresie wartości parametrów. W szczególności przeprowadzone eksperymenty potwierdzają wysoką precyzję pozycjonowania uzyskiwaną za pomocą autorskiego dwuosiowego montażu astronomicznego, dedykowanego do obserwacji astronomicznych i satelitarnych. System został również wstępnie przetestowany w trybie śledzenia, w którym występuje zmiana położenia odbiornika w czasie.

Planuje się, że przyszłe prace będą dotyczyć pogłębionej analizy teoretycznej stabilności układu zamkniętego w celu znalezienia dokładniejszych warunków dla pra-

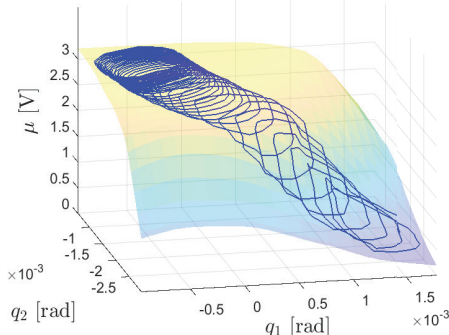
a



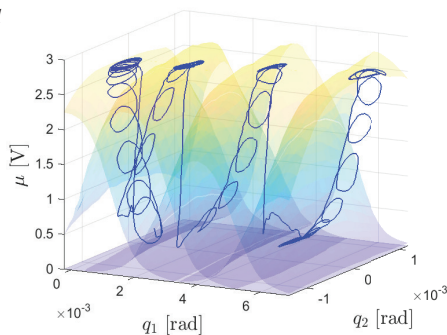
b



c



d



Rys. 8. Ścieżki w przestrzeni konfiguracyjnej podczas fazy poszukiwania maksimum
a – dla $\rho = 3,84\text{e-}4$ rad, $\nu = 0,1$ Hz, b – dla $\rho = 7,69\text{e-}5$ rad, $\nu = 0,1$ Hz,
c – dla $\rho = 3,84\text{e-}4$ rad, $\nu = 0,5$ Hz, d – dla $\rho = 2,3\text{e-}4$ rad, $\nu = 0,2$ Hz (śledzenie)

widłowego strojenia sprzężenia zwrotnego, biorąc pod uwagę model intensywności μ . Planuje się dalszą poprawę wydajności systemu w scenariuszach dynamicznych poprzez zmniejszenie opóźnień czasowych i przyspieszenie akwizycji sygnału intensywności. Przeprowadzone zostaną również kolejne testy w bardziej wymagających warunkach w otwartej przestrzeni na większych odległościach.

Literatura

- [1] J. Ye, X. Yu, P. Lin, S. Chen, Z. Zhang, and S. Tong, "Research on quadrant detector multi-spot position detection based on orthogonal frequency division multiplexing," *Applied Sciences*, vol. 13, no. 23, 2023.
- [2] W. Zhou, K. Jin, and R. Zhang, "A fast-speed gmppt method for pv array under gaussian laser beam condition in wireless power transfer application," *IEEE Transactions on Power Electronics*, vol. 37, no. 8, pp. 10 016–10 028, 2022.
- [3] K. Watanabe, T. Koyama, H. Koga, K. Izumi, and T. Tsujimura, "Robotic alignment technique of search control for laser beam communication," in *2022 61st Annual Conference of the Society of Instrument and Control Engineers (SICE)*, 2022, pp. 258–263.
- [4] S. Affan Ahmed, M. Mohsin, and S. M. Zubair Ali, "Survey and technological analysis of laser and its defense applications," *Defence Technology*, vol. 17, no. 2, pp. 583–592, 2021.

- [5] V. Nikulin, R. Khandekar, and J. Sofka, “Agile acousto-optic tracking system for free-space optical communications,” *Optical Engineering*, vol. 47, p. 064301, 2008.
- [6] M. Laas-Bourez, D. Coward, A. Klotz, and M. Boër, “A robotic telescope network for space debris identification and tracking,” *Advances in Space Research*, vol. 47, no. 3, pp. 402–410, 2011.
- [7] V. V. Nikulin, M. Bouzoubaa, V. A. Skormin, and T. E. Busch, “Modeling of an acousto-optic laser veam steering system intended for satellite communication,” *Optical Engineering*, vol. 40, no. 10, pp. 2208 – 2214, 2001.
- [8] H. Kaushal and G. Kaddoum, “Applications of lasers for tactical military operations,” *IEEE Access*, vol. 5, pp. 20 736–20 753, 2017.
- [9] R. Miller, J. Y. Gallegos, A. Lee, K. Chow, V. Tran, M. W. Bissonnette, and J. Denham, “Implementation of laser communications acquisition function using MEMS FSM and quad detector,” in *Free-Space Laser Communications XXXIV*, H. Hemmati and B. S. Robinson, Eds., vol. 11993, International Society for Optics and Photonics. SPIE, 2022, p. 1199302.
- [10] E. Frączek, W. Frączek, and A. Popiołek-Masajada, “Laser beam positioning by using a broken-down optical vortex marker,” *Applied Sciences*, vol. 11, no. 16, 2021.
- [11] E. Rimón and D. Koditschek, “Exact robot navigation using artificial potential fields,” *IEEE Trans. Robot. and Autom.*, vol. 8, no. 5, pp. 501–518, 1992.
- [12] S. Ma, P. Yang, B. Lai, C. Su, W. Zhao, K. Yang, R. Jin, T. Cheng, and B. Xu, “Adaptive gradient estimation stochastic parallel gradient descent algorithm for laser beam cleanup,” *Photonics*, vol. 8, no. 5, 2021.
- [13] K. Kozłowski, D. Pazderski, B. Krysiak, T. Jedwabny, J. Piasek, S. Kozłowski, S. Brock, D. Janiszewski, and K. Nowopolski, “High precision automated astronomical mount,” in *Automation 2019*. Cham: Springer International Publishing, 2020, pp. 299–315.
- [14] P. Bartkowiak, R. Patelski, M. Kwiatkowska, and D. Pazderski, “System architecture for development and supervision of robotic astronomical telescope,” *Pomiary Automatyka Robotyka*, vol. 26, no. 246, pp. 43–51, 4 2022.
- [15] R. Patelski and D. Pazderski, “Tracking control for a cascade perturbed control system using the active disturbance rejection paradigm,” *Archives of Control Sciences*, vol. 29, no. 2, p. 387–408, 2019.

Control of a Two-Axis Manipulator for Maximising Signal Strength in a Laser Communication System

This paper concerns extremum control for a positioning unit, aimed at maximising the intensity of the received signal in a laser communication system. It is assumed that the position of the laser beam is not measured directly. The key idea of the proposed feedback mechanism is the indirect use of the gradient of the beam intensity function in the vicinity of the sensor. This gradient is approximated using a method based on averaging along a cycloidal trajectory. The paper presents experimental results obtained using a two-axis manipulator – an astronomical mount – and a set of laser communication devices. The research results demonstrate the effectiveness of the proposed control algorithm and confirm high positioning accuracy in real-time scenarios.

Sterowanie z ograniczonym momentem w zadaniu śledzenia ścieżki manipulatora

Filip Dyba¹, Alicja Mazur¹

Streszczenie

W artykule omówiono zagadnienie śledzenia ścieżki przez manipulator, w obecności ograniczeń na dopuszczalny moment siły generowany przez napędy. Do opisu robota względem ścieżki zastosowano nieortogonalną parametryzację Serreta–Freneta oraz w pełni znany model dynamiki manipulatora. W rozwiązaniu zastosowano dwie, znane z literatury, metody do realizacji sformułowanego zadania. Badania symulacyjne przeprowadzone na modelu manipulatora RTR śledzącego krzywą 3D potwierdziły poprawność uzyskanych wyników.

1. Wprowadzenie

Manipulatory stacjonarne to urządzenia pracujące w przemyśle. Wykonują operacje technologiczne wymagające dużej precyzji i powtarzalności, a także służą do przenoszenia i operacji transportowych. Można wyróżnić trzy podstawowe typy zadań realizowanych przez roboty: śledzenie trajektorii, zadanie sterowania do punktu oraz śledzenie ścieżki [11].

Zadana trajektoria $q_d(t)$ jest krzywą sparametryzowaną czasem t . Zadanie śledzenia trajektorii polega na odtwarzaniu przez konfigurację robota $q(t)$ żądanej konfiguracji $q_d(t)$ w każdej chwili czasu t . Wynika stąd, że śledzenie trajektorii nakłada na sterowanego robota reżimy czasowe, których spełnienie może nie być możliwe.

Zadanie stabilizacji w punkcie (sterowanie do punktu, P-T-P) dla manipulatorów może być potraktowane jako szczególny przypadek śledzenia trajektorii, gdy trajektoria zadana q_d jest stała w czasie. Realizacja zadania polega na sprowadzeniu konfiguracji robota $q(t)$ do punktu q_d . Takie zadanie jest najczęściej spotykane w praktyce, mimo że pozwala tylko na zgrubne odtwarzanie kształtów przez chwytak robota.

Z kolei, zadanie śledzenia ścieżki polega na śledzeniu krzywej sparametryzowanej odległością krzywoliniową mierzoną od jej początku. W przeciwieństwie do zadania śledzenia trajektorii, podczas realizacji śledzenia ścieżki nie nakłada się na sterowanego robota reżimów czasowych. Prędkość poruszania się wzdłuż ścieżki może być tak dobrana, aby sterowany robot był w stanie ją zrealizować. Wynika stąd, że podczas realizacji śledzenia ścieżki można zażądać, aby sterowanie zaplanowane przez algorytm nie przekraczało fizycznych ograniczeń w napędach. Takie właśnie założenie przyjęto w niniejszej pracy. Pozwoli to na zaplanowanie sterowania, które nie zostanie znieskształcone poprzez ograniczenie wartości do maksymalnych wielkości uzyskanych z napędów (takie „przycięcie” amplitudy sterowań jest nazywane *projekcją*).

Jak widać, zadanie śledzenia ścieżki jest najogólniejszym zadaniem spośród zadań

¹Katedra Cybernetyki i Robotyki, Wydział Elektroniki, Fotoniki i Mikrosystemów, Politechnika Wrocławska, ul. Janiszewskiego 11/17, 50-372 Wrocław,
e-mail: filip.dyba@pwr.edu.pl, alicja.mazur@pwr.edu.pl

realizowanych przez manipulator, choć nie jest standardowym zadaniem implementowanym w robotach przemysłowych. Może być wykonane za pomocą metod nieparametrycznych [8, 12] lub z wykorzystaniem różnych parametryzacji [2, 4]. Najczęściej spotykane w literaturze są metody parametryczne, zwłaszcza oparte na parametryzacji Serreta–Freneta [1, 6], która jest reprezentacją najbardziej intuicyjną geometrycznie. Z tego powodu w artykule wykorzystano właśnie tę parametryzację 3D. Aby opisać ruch chwytaka względem układu współrzędnych przemieszczającego się wzdłuż pewnej krzywej w przestrzeni, użyto parametryzacji w wersji nieortogonalnej.

Celem uniknięcia bezpośredniej projekcji sterowań zaproponowano 2 metody skalujące parametry sterowników. Wykorzystano prostą ideę reprezentacji prawa sterowania w postaci liniowej zależności od skalowanego parametru. Dzięki temu współczynniki sygnału sterującego można dostroić *online* na podstawie układu nierówności liniowych. To proste podejście znane z literatury [10] oraz praktyki inżynierskiej zostało wpisane w zaproponowaną reprezentację zadania śledzenia ścieżki. W szczególności, wykorzystano kaskadowe połączenie sterowników na poziomach dynamicznym oraz kinematycznym. Skorzystano także z możliwości swobodnego doboru profilu prędkościowego wzdłuż ścieżki, a więc cechy typowej dla parametryzacji nieortogonalnej.

2. Matematyczny opis problemu

2.1. Parametryzacja Serreta–Freneta

Parametryzacje krzywoliniowe to jedno z podstawowych narzędzi do opisu geometrii krzywych. Pozwalają na stowarzyszenie z krzywą lokalnego układu współrzędnych, którego ewolucja zależy wprost od geometrii krzywej. Najbardziej powszechną metodą jest parametryzacja Serreta–Freneta [9]. Definiuje ona układ, który składa się z trzech wersorów rozpinających bazę ortonormalną w przestrzeni $\mathbb{R}^3$. Składowymi tego układu są: wektor styczny do krzywej T , wektor normalny do krzywej N , wektor binormalny B . Definicje poszczególnych wersorów są następujące

$$T(s) = \frac{dr(s)}{ds}, \quad N(s) = \frac{dT(s)}{\kappa}, \quad B(s) = T(s) \times N(s), \quad (1)$$

gdzie r oznacza równanie krzywej, $\kappa = \left\| \frac{dT(s)}{ds} \right\|$ definiuje krzywiznę, natomiast s jest odległością krzywoliniową opisującą dystans między początkiem krzywej, a rozważanym punktem.

Parametry w pełni charakteryzujące geometrię krzywej to krzywizna κ oraz torsja τ [7, 9]. Stąd, równania ewolucji układu Serreta–Freneta wzdłuż krzywej można zapisać w formie macierzowej jako

$$\begin{bmatrix} \frac{dT(s)}{ds} & \frac{dN(s)}{ds} & \frac{dB(s)}{ds} \end{bmatrix} = \begin{bmatrix} T^T(s) \\ N^T(s) \\ B^T(s) \end{bmatrix}^T \begin{bmatrix} 0 & -\kappa(s) & 0 \\ \kappa(s) & 0 & -\tau(s) \\ 0 & \tau(s) & 0 \end{bmatrix} = S(s)W(s), \quad (2)$$

gdzie $S(s) \in \mathbb{SO}(3)$ jest macierzą ortogonalną, której kolumny stanowią wersory układu Serreta–Freneta, $S(s) = [T(s) \ N(s) \ B(s)]$, natomiast $W(s)$ jest macierzą antysymetryczną, której elementami są parametry geometryczne krzywej.

2.2. Model manipulatora holonomicznego

Obiektem rozważanym w niniejszej pracy jest manipulator holonomiczny. Położenie końcówki chwytaka $\mathbf{p} = (x \ y \ z)^T$ względem układu bazy $X_0Y_0Z_0$ można uzyskać, stosując np. parametryzację Denavita–Hartenberga [11]. Położenie $\mathbf{p}$ można wyrazić jako funkcję konfiguracji robota, $\mathbf{p} = \mathbf{k}(\mathbf{q})$, przy czym $\mathbf{q} \in \mathbb{R}^n$ jest wektorem pozycji konfiguracyjnych manipulatora o n stopniach swobody. Na podstawie tej funkcji można uzyskać zależności prędkościowe

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{k}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad (3)$$

gdzie $\mathbf{J}(\mathbf{q})$ jest jacobianem analitycznym zależnym od konfiguracji robota $\mathbf{q}$.

Model dynamiki robota wynika z formalizmu Eulera-Lagrange'a i w ogólnej postaci dany jest równaniem [11]

$$\mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \quad (4)$$

gdzie $\mathbf{Q} \in \mathbb{R}^{n \times n}$ oznacza macierz bezwładności, $\mathbf{C} \in \mathbb{R}^{n \times n}$ opisuje macierz sił odśrodkowych i Coriolisa, $\mathbf{g} \in \mathbb{R}^n$ jest wektorem grawitacji, natomiast $\mathbf{u} \in \mathbb{R}^n$ oznacza momenty sterujące w silnikach robota. W rozważanym modelu pominięto siły tarcia.

2.3. Równania robota względem ścieżki

Do opisu robota względem ścieżki wykorzystano parametryzację nieortogonalną, którą opisano m. in. w pracy [6]. Niech punkt prowadzenia robota będzie zdefiniowany przez wektor $\mathbf{p}$. Położenie tego punktu w układzie Serreta–Freneta stowarzyszonego ze ścieżką $\mathbf{r}$ jest dane zależnością

$$\mathbf{d} = \mathbf{S}^T(\mathbf{p} - \mathbf{r}) = (d_1 \ d_2 \ d_3)^T. \quad (5)$$

Położenie punktu prowadzenia robota $\mathbf{p}$ w układzie Serreta–Freneta stowarzyszonym z krzywą $\mathbf{r}$ zobrazowano na rys. 1a. Warto zauważyć, że parametryzacja nieortogonalna nie wymaga przyjęcia żadnych dodatkowych założeń dotyczących względnego położenia punktu prowadzenia robota. Stąd, proponowany opis jest prawdziwy w każdym punkcie przestrzeni i nie występują osobliwości charakterystyczne dla parametryzacji ortogonalnej [3].

Aby opisać zachowanie robota względem ścieżki, należy zróżniczkować równanie (5) względem czasu

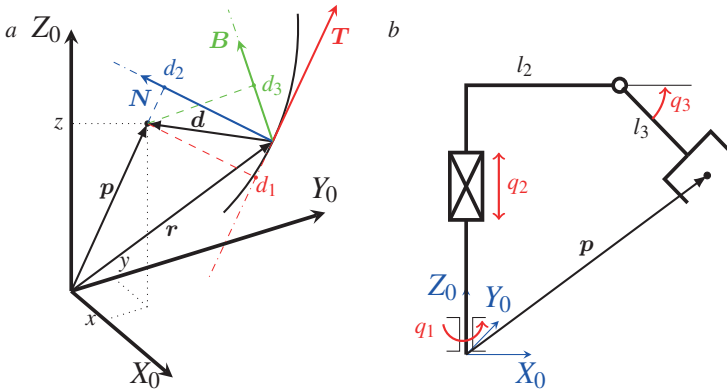
$$\dot{\mathbf{d}} = \mathbf{S}^T \dot{\mathbf{p}} - \mathbf{S}^T \dot{\mathbf{r}} + \dot{\mathbf{S}}^T(\mathbf{p} - \mathbf{r}). \quad (6)$$

Z zależności (2) oraz (5) wynika, że zachodzi

$$\dot{\mathbf{S}}^T(\mathbf{p} - \mathbf{r}) = \dot{s} \frac{d\mathbf{S}^T}{ds}(\mathbf{p} - \mathbf{r}) = \dot{s} \mathbf{W}^T \mathbf{S}^T(\mathbf{p} - \mathbf{r}) = -\dot{s} \mathbf{W} \mathbf{d}. \quad (7)$$

Ponadto, uwzględniając definicję (1) oraz ortonormalność wersorów bazowych układu Serreta–Freneta, można uzyskać

$$\mathbf{S}^T \dot{\mathbf{r}} = \begin{pmatrix} \langle \mathbf{T}, \frac{d\mathbf{r}}{ds} \dot{s} \rangle \\ \langle \mathbf{N}, \frac{d\mathbf{r}}{ds} \dot{s} \rangle \\ \langle \mathbf{B}, \frac{d\mathbf{r}}{ds} \dot{s} \rangle \end{pmatrix} = \begin{pmatrix} \langle \mathbf{T}, \mathbf{T} \rangle \\ \langle \mathbf{N}, \mathbf{T} \rangle \\ \langle \mathbf{B}, \mathbf{T} \rangle \end{pmatrix} \dot{s} = \dot{s} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \dot{s} \mathbf{i}. \quad (8)$$



Rys. 1. Położenie punktu prowadzenia robota

a – realizacja parametryzacji nieortogonalnej w układzie Serreta–Freneta, b – przykładowa struktura manipulatora holonomicznego wraz z punktem prowadzenia robota

Korzystając z definicji składników (7) oraz (8), a także uwzględniając definicję (3) w równaniu (6), można uzyskać równania kinematyki robota względem ścieżki

$$\dot{d} = S^T J \dot{q} - \dot{s}i - \dot{s}Wd = \underbrace{S^T J}_{L} \dot{q} + \underbrace{\dot{s}(-i - Wd)}_f = L\dot{q} + sf. \quad (9)$$

3. Postawienie problemu sterowania

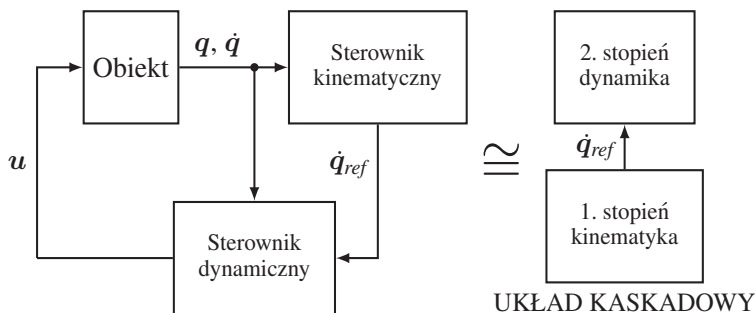
W niniejszej pracy sformułowano następujący problem sterowania:

Należy znaleźć algorytm sterowania dla manipulatora o całkowicie znanej kinematyce i dynamice, który będzie realizował zadanie śledzenia trójwymiarowej ścieżki co najmniej klasy C^3 (wynika to z definicji układu Serreta–Freneta [6]), a generowany przez niego moment sterujący nie będzie przekraczał fizycznych ograniczeń nałożonych przez napędy.

W rozdziale 2 pokazano, że matematyczny opis ruchu manipulatora przemieszczającego się wzdłuż ścieżki składa się z dwóch grup równań: opisu względem krzywej sformułowanego z wykorzystaniem parametryzacji nieortogonalnej 3D Serreta–Freneta, spełniającego rolę ograniczeń różniczkowych pierwszego rzędu, oraz równań dynamiki. Ze względu na swoją postać, opis ten ma strukturę kaskadową, co wymaga zastosowania algorytmu sterowania dedykowanego układom kaskadowym, czyli algorytmu całkowania wstecznego (z ang. *backstepping integrator*) [5].

Problem sterowania może być rozwiązany poprzez zastosowanie procedury całkowania wstecznego, która sprowadza się do rozwiązania dwóch osobnych zadań:

1. sterowanie kinematyczne – znalezienie prędkości pomocniczych $\dot{q}_{ref}(t)$, które rozwiązują problem śledzenia ścieżki dla równania kinematyki, czyli ograniczeń prędkościowych (I stopień kaskady),



Rys. 2. Schemat układu sterowania dla manipulatora śledzącego zadaną ścieżkę

2. sterowanie dynamiczne – sterowanie dynamiką z uwzględnieniem rozwiązania problemu uzyskanego ze sterownika kinematycznego (II stopień kaskady). W tym algorytmie wprowadzono modyfikacje pozwalające na ograniczenie amplitud sterowań, aby nie przekraczały ograniczeń na dopuszczalne momenty w napędach.

Schemat układu sterowania dla systemu kaskadowego, jakim jest manipulator śledzący zadaną ścieżkę krzywoliniową, został przedstawiony na rys. 2.

Ponadto, rozważono ograniczenia na dopuszczalne momenty w napędach, wynikające z limitów fizycznych, zdefiniowane w następującej formie

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}, \quad (10)$$

przy czym ograniczenia w każdym wymiarze są definiowane niezależnie, tzn. zachodzi $\mathbf{u}_{min} = (u_1^{min} \dots u_n^{min})^T$ oraz $\mathbf{u}_{max} = (u_1^{max} \dots u_n^{max})^T$.

4. Prawo sterowania

4.1. Realizacja zadania śledzenia ścieżki

Aby rozwiązać problem sformułowany w rozdziale 3, należy zaproponować prawo sterowania na poziomie kinematycznym, a także na poziomie dynamicznym, które zagwarantują zbieżność do zera odpowiednio błędów śledzenia ścieżki $\mathbf{e}_d$ oraz błędów śledzenia profili prędkościowych $\dot{\mathbf{e}}_q$.

Celem spełnienia ograniczeń kinematycznych wynikających z definicji zadania śledzenia ścieżki (9), można zastosować następujące prawo sterowania

$$\dot{\mathbf{q}}_{ref} = \mathbf{L}^{-1}(\dot{\mathbf{d}}_d - \mathbf{K}_k \mathbf{e}_d - \dot{\mathbf{s}} \mathbf{f}), \quad (11)$$

gdzie $\mathbf{d}_d$ oznacza żądane położenie robota względem ścieżki, $\mathbf{e}_d = \mathbf{d} - \mathbf{d}_d$ definiuje błędy śledzenia ścieżki, $\mathbf{K}_k$ jest dodatnio określoną macierzą wzmocnień, natomiast $\dot{\mathbf{s}}$ jest zadanym profilem prędkościowym wzdłuż ścieżki. Warto podkreślić, że dla parametryzacji nieortogonalnej profil prędkościowy wzdłuż ścieżki $\dot{\mathbf{s}}$ jest niezależnym stopniem swobody wpływającym na ewolucję układu lokalnego wzdłuż krzywej.

Z kolei, za realizację profili prędkościowych wygenerowanych przez sterownik kinematyczny (11) odpowiada prawo sterowania bazujące na pełnej znajomości modelu dynamiki (4)

$$\mathbf{u} = \mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}}_{ref} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}}_{ref} + \mathbf{g}(\mathbf{q}) - \mathbf{K}_d \dot{\mathbf{e}}_q, \quad (12)$$

przy czym $\dot{e}_q = \dot{q} - \dot{q}_{ref}$ definiuje błąd śledzenia profili prędkościowych, a K_d jest dodatnio określoną macierzą wzmocnień.

Prawo sterowania zdefiniowane dla obu poziomów kaskady układu sterowania gwarantuje zbieżność do zera błędów śledzenia profili prędkościowych $\dot{e}_q$ oraz błędów śledzenia ścieżki e_d . Analiza stabilności algorytmu jest analogiczna do wnioskowania przedstawionego w pracach [3, 7].

4.2. Ograniczenia na sterowanie

Aby uwzględnić ograniczenia na sterowanie u , należy odpowiednio zmodyfikować sterownik dynamiczny zaproponowany w rozdziale 4.1. W tym celu rozważono następujące metody:

- skalowanie trajektorii robota,
- skalowanie profilu prędkościowego wzdłuż ścieżki.

Obie metody są oparte na idei uporządkowania dynamicznego prawa sterowania (12) do postaci funkcji liniowej zależnej od skalowanego parametru

$$u = x_1 \psi + x_2, \quad (13)$$

gdzie ψ oznacza skalowany parametr, odmienny dla każdego podejścia. Dla takiej postaci sterowania problem ograniczeń (10) sprowadza się do układu nierówności liniowych. Na ich podstawie można określić wartość skalowanego parametru ψ tak, by spełniała zależność

$$\psi_{min} \leq \psi \leq \psi_{max}, \quad \text{gdzie} \quad \psi_{max} = \min_{i \in \{1,2,\dots,n\}} \psi_i^+, \quad \psi_{min} = \max_{i \in \{1,2,\dots,n\}} \psi_i^-, \quad (14)$$

przy czym warunki brzegowe ψ_i^+ oraz ψ_i^- dla każdego wymiaru wektora u są definiowane zgodnie z regułami

$$\psi_i^+ = \begin{cases} (u_i^{max} - x_{2i})/x_{1i}, & x_{1i} > 0, \\ \infty, & x_{1i} = 0, \\ (u_i^{min} - x_{2i})/x_{1i}, & x_{1i} < 0, \end{cases} \quad \psi_i^- = \begin{cases} (u_i^{min} - x_{2i})/x_{1i}, & x_{1i} > 0, \\ -\infty, & x_{1i} = 0, \\ (u_i^{max} - x_{2i})/x_{1i}, & x_{1i} < 0. \end{cases} \quad (15)$$

Odpowiedni dobór skalowanego parametru $\psi \in [\psi_{min}; \psi_{max}]$ pozwala zachować ograniczenia momentów sterujących. Należy podkreślić, że obie metody nie wpływają na stabilność algorytmu sterowania. Jednakże, warunkiem koniecznym istnienia rozwiązania jest $\psi_{min} \leq \psi_{max}$.

4.2.1. Metoda skalowania trajektorii robota

Metoda skalowania trajektorii robota (MSTR) polega na modyfikacji trajektorii realizowanej przez robota w czasie na podstawie zadanej geometrycznie ścieżki. MSTR jest oparta o metodę sterowania wirtualnego, która została przedstawiona w pracy [10]. Proponowana metoda polega na wyróżnieniu elementów dynamicznego prawa sterowania, które zależą od wzmocnień sterowników z obu poziomów kaskady $a_1(K_d, K_k)$, a także elementów niezależnych od wzmocnień a_2 . Następnie należy zdefiniować współczynnik skalujący $\psi = k(t)$. Wtedy równanie (13) przyjmuje postać

$$u = a_1(K_k, K_d) \cdot k(t) + a_2, \quad (16)$$

Jeśli sterownik dynamiczny (12) generuje momenty sterujące w dopuszczalnym zakresie, to należy przyjąć $k(t) = 1$. W przeciwnym razie należy zredukować wzmocnienia poprzez dobór wartości $k(t)$ zgodnie z regułą (14). Należy podkreślić, że dla zachowania stabilności algorytmu sterowania, należy zagwarantować, że $k(t) \in [0; 1]$.

Warto zauważyć, że zaproponowana metoda pozwala skalować wartości wyrazów zależnych od błędów na obu poziomach kaskady. W związku z tym MSTR jest przydatna w fazie przejściowej, gdy robot znajduje się poza ścieżką. Dzięki skalowaniu wartości wzmocnień można ograniczyć gwałtowność reakcji układu sterowania.

4.2.2. Metoda skalowania profilu prędkościowego wzdłuż ścieżki

Metoda skalowania profilu prędkościowego wzdłuż ścieżki (MSPP) polega na skalowaniu zadanego profilu prędkościowego wzdłuż ścieżki $\dot{s}$, który stymuluje ewolucję układu Serreta–Freneta. Szybkość ewolucji układu na ścieżce ma bowiem bezpośredni wpływ na prędkość ruchu robota. Idea MSPP polega na modyfikowaniu przyspieszenia krzywoliniowego $\ddot{s}$, aby zachować fizyczne ograniczenia napędów.

Aby zrealizować MSPP, należy wyodrębnić w dynamicznym prawie sterowania (12) elementy zależne liniowo od drugiej pochodnej odległości krzywoliniowej $b_1(s, \dot{s})$ oraz składniki niezależne od przyspieszenia krzywoliniowego $b_2(s, \dot{s})$. Wówczas równanie (13) przyjmuje postać liniowo zależną od skalowanego parametru $\psi = \dot{s}$

$$u = b_1(s) \cdot \dot{s} + b_2(s, \dot{s}). \quad (17)$$

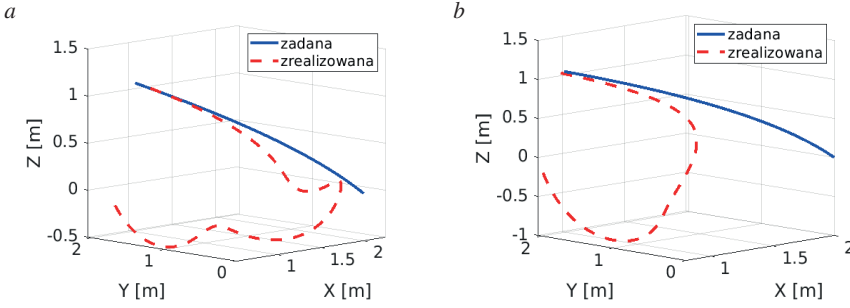
Dla równania (17) należy zastosować zasadę (14), aby dobrać wartość $\dot{s}$, dla której ograniczenia momentów sterujących są zachowane. Warto zauważyć, że MSPP skutkuje także redukcją referencyjnych profili prędkościowych (11) poprzez ograniczanie zadanego profilu prędkościowego wzdłuż ścieżki.

5. Badania symulacyjne

Przedstawione metody ograniczania momentów sterujących zostały zweryfikowane symulacyjnie. Każde rozwiązanie analizowano w osobnym studium przypadku. Rezultaty zestawiono z przypadkami, w których zastosowano identyczne parametry zadania, lecz pominięto ograniczenia na sterowanie. Takie scenariusze oznaczono mianem *przypadków referencyjnych*.

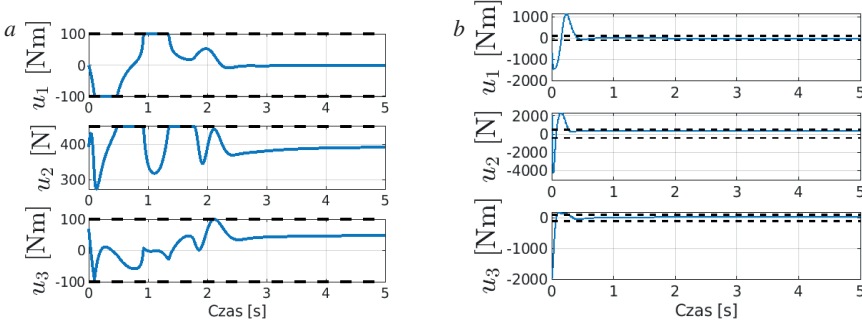
Obiektem sterowania w przeprowadzonych badaniach symulacyjnych był manipulator holonomiczny o 3 stopniach swobody, którego strukturę kinematyczną przedstawiono na rys. 1b. Przyjęto, że obowiązują następujące ograniczenia na sterowania: $u_{max} = (100 [\text{Nm}] \quad 450 [\text{N}] \quad 100 [\text{Nm}])^T$, $u_{min} = -u_{max}$.

W pierwszym studium analizowano MSTR. Na rys. 3 przedstawiono zadaną ścieżkę oraz ścieżkę zrealizowaną przez manipulator. Widać wyraźnie, że ograniczenie momentów sterujących w istotny sposób zmodyfikowało trajektorię zrealizowaną przez robota. Przebyta droga jest dłuższa, a kierunek ruchu końcówki chwytaka ulegał częstym zmianom. Wynikało to ze skutecznego ograniczenia wartości momentów sterujących, których przebiegi przedstawiono na rys. 4. W przypadku referencyjnym można zauważyć, że w początkowej fazie ruchu wartości sterowań przekraczają dopuszczalne limity, co mogłoby skutkować uszkodzeniami mechanicznymi. MSTR skutecznie pozwoliła zmniejszyć gwałtowność reakcji układu sterowania. Gdy błędy śle-



Rys. 3. Ścieżka zadana i zrealizowana – MSTR

a – wykorzystanie metody, b – przypadek referencyjny (nie uwzględniono ograniczeń)



Rys. 4. Sterowania wraz z ograniczeniami – MSTR

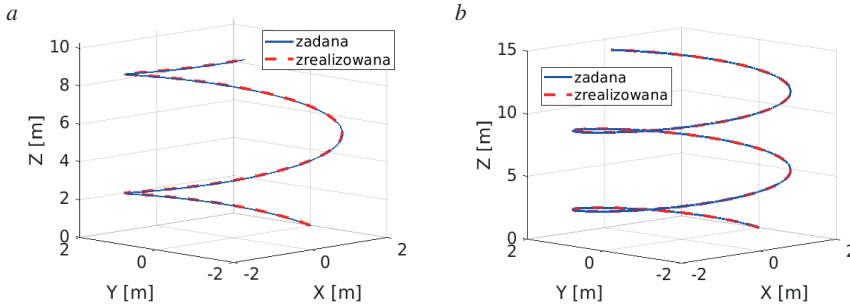
a – wykorzystanie metody, b – przypadek referencyjny (nie uwzględniono ograniczeń)

dzenia były znaczące, wzmocnienia sterownika zostały zredukowane, dzięki czemu uzyskano momenty sterujące w dopuszczalnym zakresie.

W drugim studium badano MSPP. Zmieniono konfigurację początkową manipulatora tak, aby punkt prowadzenia robota znajdował się na ścieżce, gdyż MSPP nie umożliwia redukcji momentów wynikających z błędów śledzenia. Natomiast profil prędkościowy wzdłuż ścieżki zadano jako $\dot{s}(t) = \frac{6}{1+t^2} \left[\frac{\text{m}}{\text{s}^2} \right]$. Na rys. 5 porównano ścieżki zadaną i zrealizowaną przez robota. W obu scenariuszach ścieżki pokrywają się, co świadczy o poprawnej realizacji zadania. MSPP nie wpływa na kształt krzywej realizowanej przez manipulator. Skalowanie profilu prędkościowego spowodowało wprawdzie, że w tym samym czasie zrealizowano krótszy fragment ścieżki, jednak czas wykonania zadania zgodnie z definicją jest dowolny. Z kolei, na rys. 6 zestawiono wygenerowane sterowania. Można zauważyć, że zadany profil prędkościowy skutkowało przekroczeniem dopuszczalnych limitów najpierw w przegubach 1 oraz 2, a w trakcie manewru – również w przegubie 3. MSPP pozwoliła na redukcję profilu prędkościowego tak, że momenty sterujące we wszystkich przegubach zostały zachowane.

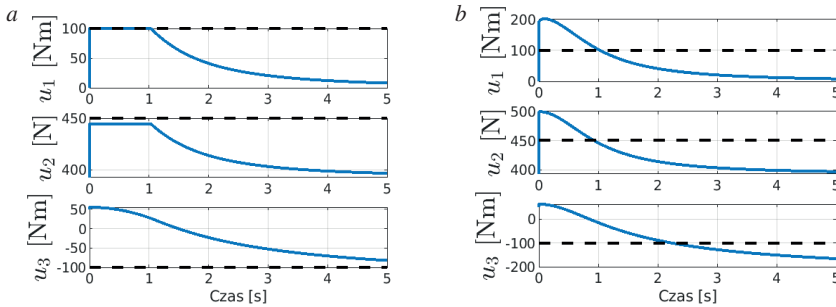
6. Podsumowanie

W niniejszej pracy przedstawiono dwie metody pozwalające na ograniczenie momentów sterujących w zadaniu śledzenia ścieżki. MSTR pozwala redukować wzmocnie-



Rys. 5. Ścieżka zadana i zrealizowana – MSPP

a – wykorzystanie metody, b – przypadek referencyjny (nie uwzględniono ograniczeń)



Rys. 6. Sterowania wraz z ograniczeniami – MSPP

a – wykorzystanie metody, b – przypadek referencyjny (nie uwzględniono ograniczeń)

nia sterowników tak, aby zmniejszyć gwałtowność reakcji w stanie przejściowym. Z kolei MSPP pozwala redukować zadany profil prędkościowy układu Serreta–Freneta, aby zwolnić ruch robota wzdłuż ścieżki. Badania symulacyjne wykazały, że oba sposoby pozwalają na poprawną realizację zadanej ścieżki przy zachowaniu ograniczeń w postaci (10). Zalety i wady rozważanych podejść zestawiono w tab. 1. W przyszłych badaniach należałoby porównać skuteczność zaproponowanych metod z innymi znanymi rozwiązaniami, np. nowoczesnymi algorytmami typu *anti-windup* generującymi rozwiązania suboptymalne. Warto również rozważyć połączenie MSTR oraz MSPP celem uzyskania rozwiązania skutecznego zarówno w fazie dojścia do ścieżki, jak i w stanie ustalonym. Ponadto, można rozszerzyć MSPP o ograniczenia prędkościowe na poziomie kinematycznym.

Literatura

- [1] W. Donski, A. Mazur. Path tracking with orthogonal parametrization for a satellite with partial state information. W: Proc. 15th ICINCO, 2018. wolumen 2, s. 262–267.
- [2] F. Dyba, M. Frego. Non-Orthogonal Serret–Frenet Parametrization Applied to Path Following of B-Spline Curves by a Mobile Manipulator. Robotics, 2024, wolumen 13, numer 9.
- [3] F. Dyba, A. Mazur. Comparison of curvilinear parametrization methods and avoidance of orthogonal singularities in the path following task. JAMRIS, 2024,

Tab. 1. Porównanie metod ograniczających momenty sterujące w zadaniu śledzenia ścieżki

Metoda	Zalety	Wady
MSTR	+ Zachowanie ograniczeń na sterowanie w postaci (10). + Skuteczność w fazie dojścia robota do ścieżki. + Zachowanie stabilności algorytmu dla $k(t) \in [0; 1]$.	– Brak wpływu na sterowania w stanie ustalonym. – Warunek istnienia rozwiązania $k_{min} \leq k_{max}$. – Nieciągłe momenty sterujące dla nieciągłych zmian $k(t)$.
MSPP	+ Zachowanie ograniczeń na sterowanie w postaci (10). + Skuteczność w stanie ustalonym, gdy robot porusza się wzdłuż ścieżki. + Nie wpływa na analizę stabilności algorytmu.	– Przyspieszenia krzywoliniowe jako element definicji zadania. – Brak odporności na duży błąd w stanie początkowym. – Warunek istnienia rozwiązania $\ddot{s}_{min} \leq \ddot{s}_{max}$.

wolumen 17, numer 3, s. 46–64.

- [4] N. Hung et al. A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments. *Journal of Field Robotics*, 2023, wolumen 40, numer 3, s. 747–779.
- [5] M. Krstić, I. Kanellakopoulos, P. V. Kokotovic. *Nonlinear and Adaptive Control Design*. New York, Wiley 1995.
- [6] A. Mazur, F. Dyba. The Non-orthogonal Serret–Frenet Parametrization Applied to the Path Following Problem of a Manipulator with Partially Known Dynamics. *ACS*, 2023, wolumen 33, numer 2, s. 339–370.
- [7] A. Mazur, J. Płaskonka, M. Kaczmarek. Following 3D paths by a manipulator. *ACS*, 2015, wolumen 25, numer 1, s. 117–133.
- [8] M. M. Michałek, T. Gawron. VFO path following control with guarantees of positionally constrained transients for unicycle-like robots with constrained control input. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2018, wolumen 89, numer 1-2, s. 191 – 210.
- [9] J. Oprea. *Differential Geometry and Its Applications*. Cleveland State University, Prentice Hall, Washington 2007.
- [10] G. Pająk. *Manipulatory mobilne. Zadania z ograniczeniami na stan i sterowanie*. Zielona Góra, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego 2019.
- [11] K. Tchoń et al. *Manipulatory i roboty mobilne: Modele, Planowanie Ruchu, Sterowanie*. Warszawa, Akademicka Oficyna Wydawnicza PLJ 2000.
- [12] W. Yao. *Guiding Vector Fields for Robot Motion Control*. Springer Tracts in Advanced Robotics. Springer 2023.

Limited-torque control in the path following task for a manipulator

This paper considers a problem of a manipulator performing the path following task in the presence of constraints on the allowable control torques generated by actuators. In order to describe a robot with respect to the given path, the non-orthogonal Serret–Frenet parametrization and the fully known model of the manipulator dynamics were used. In the solution two methods, which are known from the literature, were adopted to accomplish the formulated task. Simulation studies conducted on the RTR manipulator model following a 3D curve confirmed correctness of the obtained results.

INTERAKCJE ROBOT-CZŁOWIEK

Selecting the number of key postures for an unsupervised method of human action recognition

Esther Ekemeyong¹, Teresa Zielinska¹, Vibekananda Dutta²

Abstract

This paper discusses the problem of selecting the appropriate number of clusters for recognizing key human body postures using an unsupervised learning method. Commonly used indices such as Silhouette, Dunn, Davies-Bouldin, and Calinski-Harabasz, used to select the appropriate number of clusters, shed light on the overall separation and compactness of clusters but do not directly determine how strongly each data vector belongs to its assigned cluster. This work aims to evaluate how effectively data are clustered. A novel Discriminant Score is proposed. The Self-Organizing Map algorithm is used as an unsupervised technique. The problem is demonstrated using experimental validation.

1. Introduction and Problem Statement

Several fields have demonstrated the growing need for robots to collaborate with humans to carry out important tasks. Recognising human actions is needed for many reasons [1]. The examples are surveillance systems, assistive technologies in assembly lines[2], elderly care [3, 4], search and rescue [5]. Relevant research fields are self-learning robots. In 'learning by observation', the robot learns what sequence of body postures is needed to perform a given action. Then, it retrieves the motion primitives (position specifications) from its database to implement the observed postures. There are many approaches used for human action recognition and prediction. Often, actions are realized by adopting appropriate sequences of body postures. During observation combined with posture recognition, the hypothesis regarding the performed action becomes narrower as the sequence of postures becomes longer. At a certain point, this sequence is exclusively specific to the given action. Therefore, it becomes possible to formulate a conclusion about the ongoing action. The validity of such an approach was tested in [6]. Most works on human action prediction using neural networks or deep learning techniques apply classification (supervised) approaches [7]. When focusing on postures, the goal is to recognize whether a person has adopted one of several predefined postures. Simple methods include SVM, K-means algorithms, Perceptron, or Bayesian learning models [8]. Popular supervised deep learning methods include convolutional neural networks, long-term memory networks, recurrent

¹Warsaw University of Technology, Faculty of Power and Aeronautical Engineering, Nowowiejska 24, 00-665 Warsaw (ekemeyong.lisane_esther.dokt@pw.edu.pl, teresa.zielinska@pw.edu.pl).

²Warsaw University of Technology, Faculty of Mechatronics, Sw. Andrzeja Boboli 8, 02-525 Warsaw, (vibekananda.dutta@pw.edu.pl).

neural networks, graph neural networks, graph neural networks, and Transformers [9, 10, 11, 12]. The pre-processed human motion data collected are divided by a human expert into sequences representing the key postures. This labelling process is time-consuming and possesses a component of subjectivity. Contrary to supervised approaches, the clustering techniques (unsupervised learning) do not require data labelling, i.e. assigning postures to the recorded sequences of body positions listed for training. The human body positions are automatically grouped into clusters representing the postures. Hence, the 'resolution' of posture recognition is uncontrolled. This can be a disadvantage in some applications. However, such an approach avoids time-consuming and subjective labelling and makes them easier to use in the case of new data and new postures. This paper discusses the problem of selecting the appropriate number of clusters for unsupervised clustering applied to human posture recognition. An unsupervised Self-Organizing Map (SOM) is used for clustering. The advantage of unsupervised SOM is its simplicity, transparent data processing mechanism, and computational efficiency. Moreover, it can be modified for training in a supervised or unsupervised manner, as described in our other work [13]. The proposed method for selecting the number of clusters can also be used in any other clustering approach.

2. Self Organizing Map

The SOM architecture, utilized for human posture classification in this work, is an unsupervised neural network that maps high-dimensional data into a lower-dimensional grid while preserving the topological relationships of the data. This approach makes SOMs particularly suitable for tasks like human posture classification, where complex spatial relationships and transitions must be captured effectively. The architecture comprises an input layer and a 1D grid of neurons. The number of neurons in the grid reflects the number of clusters in which the data will be categorized. Each neuron in the grid is associated with a weight vector of the same dimensionality as the input data, initialized randomly. In our architecture, each neuron consists of weights of dimensionality 1×57 . SOM identifies the neuron in the grid, known as the Best Matching Unit (BMU), whose weight vector is closest to the input vector. We measure this closeness using the cosine and Euclidean distances.

In the SOM, we have the fixed number of neurons denoted by M . To find the best match for each input vector $\mathbf{f}_j$ ($j = 1, 2, \dots, N$), the SOM identifies among the neurons, such neuron m_{BMU} , which weights are nearest to $\mathbf{f}_j$ in terms of assumed distance metrics. BMU means Best Matching Unit, and $m_{BMU} = \arg \min_m \text{DIST}(\mathbf{f}_j - \mathbf{c}_m)$, where $\mathbf{c}_m$ denotes the weight vector of m -th neuron ($m = 1, \dots, M$). The Euclidean norm $\|\bullet\|$ or the cosine distance is often used as the distance measure. The cosine distance is expressed by $d_{\cos}(\mathbf{f}_j, \mathbf{c}_m) = 1 - \frac{\mathbf{f}_j \cdot \mathbf{c}_m}{\|\mathbf{f}_j\| \|\mathbf{c}_m\|}$. After determining the BMU, the weights of the BMU and its neighbouring neurons are updated to bring them closer to the input vector using the formula $\mathbf{c}_m(t+1) = \mathbf{c}_m(t) + \eta(t) \cdot h_{BMU}(t) \cdot (\mathbf{f}_j - \mathbf{c}_{BMU}(t))$, where t represents the iteration index, $\eta(t)$ is the learning rate that decreases over iterations, $h_{BMU}(t)$ is the neighbourhood function and index m in this case denotes only the BMU and neighbouring neurons. As the neighbourhood function, often Gaussian is used, it determines the influence of the BMU on its neighbours as given by $h_m(t) = \exp\left(-\frac{\|c_{BMU} - c_m\|^2}{2\sigma(t)^2}\right)$, where c_{BMU} and c_m are the coordinates of the BMU

and its neighbours in the neurons grid. Here, $\sigma(t)$ denotes the neighbourhood radius, which decreases with iterations. During training, the SOM network generates cluster centroids (BMUs). A given cluster includes postures located closer to a given BMU than others in the sense of the adopted distance measure. The determination of BMUs and the update of their weights is performed repeatedly for the training data until a stopping condition is reached (defined as a negligible value of the weight update).

3. Clustering Quality Scores

Determining the most appropriate number of clusters is essential to know how effectively our SOM captures the underlying structure of posture data. The common measures used to assess clustering quality are the Silhouette Coefficient, Dunn Index, Davies-Bouldin Index, and Calinski-Harabasz Index. Each of these measures considers cluster separation and compactness from slightly different perspectives. In our previous studies, we have shown that these measures suggest divergent numbers of clusters. Therefore, we proposed a dedicated measure, namely the Discriminant Score (DS) [14]. In this paper, we present an improved Discriminant Score. Based on experimental studies, we describe selecting the appropriate number of clusters for the posture recognition task. Below, we briefly describe other scores for a better illustration of the specific features of our DS score.

Let N be the total number of data points for the vector $\mathbf{f}_j$. The number of data points assigned to cluster m is K_m , the k -th vector belonging to cluster m is denoted by $\mathbf{f}_{m,k}$, where $k = 1, 2, \dots, K_m$. We represent by $DIST(\mathbf{x}, \mathbf{y})$ the chosen distance function (e.g. Euclidean or cosine) between the two vectors $\mathbf{x}$ and $\mathbf{y}$.

The **Silhouette Coefficient** SC measures how close each data point $\mathbf{f}_{m,k}$ lies to other points in the same cluster m compared to points in different clusters. This means that it evaluates how well each data point fits within its assigned cluster compared to others. To do this, two quantities are computed: (1) the mean distance between $\mathbf{f}_{m,k}$ and every other point $\mathbf{f}_{m,k'}$ in the same cluster, namely $\bar{a}_m(k) = \frac{1}{K_m-1} \sum_{k' \neq k}^{K_m} DIST(\mathbf{f}_{m,k}, \mathbf{f}_{m,k'})$, called the average intra-cluster distance and (2) the smallest average distance between $\mathbf{f}_{m,k}$ and the data points assigned to any other cluster m' :

$$\bar{b}_{\min}(m, k) = \min_{m'} \left(\frac{1}{K_{m'}} \sum_{k'=1}^{K_{m'}} DIST(\mathbf{f}_{m,k}, \mathbf{f}_{m',k'}) \right), \quad (m' = 1, \dots, M \text{ and } m' \neq m), \text{ which}$$

is called the minimum inter-cluster distance. The Silhouette Coefficient for $\mathbf{f}_{m,k}$ is defined as:

$$SC = \frac{\bar{b}_{\min}(m, k) - \bar{a}_m(k)}{\max(\bar{a}_m(k), \bar{b}_{\min}(m, k))}. \quad (1)$$

This score ranges from -1 to $+1$. A higher (positive) value implies that $\mathbf{f}_{m,k}$ is closer to its own cluster than to any other cluster, indicating good separability. A negative score, conversely, signifies potential misclassification.

The **Dunn Index** (DI) focuses on maximizing the distance between distinct clusters and minimizing the maximum intra-cluster distance:

$$d_{\max}(m) = \max_{k \neq k'} DIST(\mathbf{f}_{m,k}, \mathbf{f}_{m,k'}) \text{ is the largest distance between any two points in}$$

cluster m . The global maximum across all clusters is: $d_{glob}^{max} = \max_{m=1, \dots, M} \{d_{max}(m)\}$. The minimum inter-cluster distance is given by $d_{c_{min}}^{glob} = \min_{\substack{m \neq m' \\ m, m' = 1, \dots, M}} DIST(\mathbf{cc}_m, \mathbf{cc}_{m'})$. The

Dunn Index is expressed by:

$$DI = \frac{d_{min}^{glob}}{d_{max}^{glob}}. \quad (2)$$

A larger DI indicates more distinct (better-separated) clusters with smaller intra-cluster spread.

The **Davies-Bouldin Index** (DBI) combines the notion of average intra-cluster distance and the distance between cluster centroids. First, let the average distance of points belonging to cluster m to its centroid $\mathbf{cc}_m$ is obtained:

$\bar{a}(m) = \frac{1}{K_m} \sum_{k=1}^{K_m} DIST(\mathbf{f}_{m,k}, \mathbf{cc}_m)$. Then, for each pair of clusters (m, m') is calculated: $d(m, m') = DIST(\mathbf{cc}_m, \mathbf{cc}_{m'})$ and next the average: $d_{av}(m, m') = \frac{\bar{a}(m) + \bar{a}(m')}{d(m, m')}$. The maximum value of $d_{av}(m, m')$ over is obtained over all M clusters: $d_{av, max}(m) = \max_{m' \neq m} \{d_{av}(m, m')\}$, $(m = 1, \dots, M)$. The Davies–Bouldin Index is then the average of these maxima:

$$DBI = \frac{1}{M} \sum_{m=1}^M d_{av, max}(m). \quad (3)$$

A smaller DBI implies a better clustering structure with well-separated and internally coherent clusters.

Sometimes referred to as the *Variance Ratio Criterion*, the **Calinski–Harabasz Index** (CH) compares the *between-cluster* dispersion to the *within-cluster* dispersion. Higher CH values imply more distinct partitions, where clusters are well-separated and internally compact. Let $\mathbf{cc} = \frac{1}{N} \sum_{m=1}^M \sum_{k=1}^{K_m} \mathbf{f}_{m,k}$ be the global centroid of all N data vectors. The between-cluster dispersion B is typically defined as:

$B = \sum_{m=1}^M K_m d(\mathbf{cc}_m, \mathbf{cc})^2$, while the within-cluster dispersion W is given by $W = \sum_{m=1}^M \sum_{k=1}^{K_m} d(\mathbf{f}_{m,k}, \mathbf{cc}_m)^2$. The Calinski-Harabasz Index is expressed by:

$$CH = \frac{B/(M-1)}{W/(N-M)}. \quad (4)$$

The numerator captures how far each cluster centroid $\mathbf{cc}_m$ is from the global centroid $\mathbf{cc}$, weighted by K_m – the number of data points belonging to cluster m . The denominator measures how scattered each point $\mathbf{f}_{m,k}$ is around its own cluster centroid. Dividing by $(M-1)$ and $(N-M)$ normalizes these terms by the number of clusters and the dataset size. A larger CH indicates more separable and internally cohesive clusters.

Although global indices such as Silhouette, Dunn, Davies-Bouldin, and Calinski-Harabasz shed light on the overall separation and compactness of clusters, they do not

directly quantify *how strongly* each data vector $\mathbf{f}_j$ belonging to its assigned cluster. To address this limitation, we defined the **Discriminant Score**, denoted by $DS_m(\mathbf{f}_j)$, which assesses the local relationship between $\mathbf{f}_j$ and the centroid $\mathbf{cc}_m$ of cluster m [14].

Let $DIST(\mathbf{f}_j, \mathbf{cc}_m)$ be the distance function chosen. If $DIST(\mathbf{f}_j, \mathbf{cc}_m) \neq 0$, then $a(\mathbf{f}, \mathbf{cc}_m) = \frac{1}{|DIST(\mathbf{f}_j, \mathbf{cc}_m)|}$, the DS score is expressed as:

$$DS_m(\mathbf{f}_j) = 1 - \frac{a(\mathbf{f}_j, \mathbf{cc}_m)}{\sum_{m'=1}^M DIST(\mathbf{f}_j, \mathbf{cc}_{m'})}. \quad (5)$$

$DS_m(\mathbf{f}_j) = 1$ when $DIST(\mathbf{f}_j, \mathbf{cc}_m) = 0$. If $\mathbf{f}_j$ is significantly closer to $\mathbf{cc}_m$ than to other centroids, $DIST(\mathbf{f}_j, \mathbf{cc}_m)$ is small, so $DS_m(\mathbf{f}_j)$ approaches 1. If $\mathbf{f}_j$ is relatively far from $\mathbf{cc}_m$, then $DS_m(\mathbf{f}_j)$ decreases toward 0. The value of $DS_m(\mathbf{f}_j)$ spans $\langle 0, 1 \rangle$. A score near 1 suggests that $\mathbf{f}_j$ strongly belongs to cluster m , while a value near 0 indicates that $\mathbf{f}_j$ is better aligned with a different cluster centroid. Unlike global indices, which evaluate the overall structure of the clustering, the DS score provides a *fine-grained* measure of how well each individual data point $\mathbf{f}_j$ fits into its assigned cluster. Combining these local scores with global validity indices provides a comprehensive framework for selecting the optimal number of clusters and verifying the consistency of posture classification results.

4. The Dataset and Data Preprocessing

In this article, the term key posture addresses a range of body positions, such as walking, sitting down, standing up, etc. The clusters should represent the key postures indicative of specific actions serving as the foundation for human action recognition and prediction. The data describing the human postures during different actions were recorded by RGB-D cameras. The software associated with these sensors automatically performed data pre-processing with data reliability validation. The data describe the positions of relevant human body points by 3D coordinates, expressing each body position as a skeleton. The data used in this study are coming from two datasets recorded by our team [13]: the WUT-20 dataset and the Waseda University (Waseda) dataset. We utilized these datasets to train and test a SOM network, first with a controlled subset of recordings and later with the full combined dataset to broaden the range of postures and assess the most appropriate number of clusters. The first (limited) studies are given in [14] with more data preprocessing information. In this paper, we consider 30 actions. Examples of additional actions not previously considered in [14] include: *cleaning the wall, answering a phone call, packing boxes, etc.* For every experiment, i.e. the initial training action (TA) and the dataset of actions used to train the SOM, the data were partitioned in the ratio 7:2:1 for training, testing and validation samples accordingly. The data were expressed in the reference frame attached to the specific point in the human spine and normalised considering the motion ranges.

In the presented study, each posture is represented by 57-dimensional feature vectors, corresponding to 19 body key points (each body key point is represented by 3D coordinate). These points include: the head, neck, spine-chest, spine-naval, pelvis,

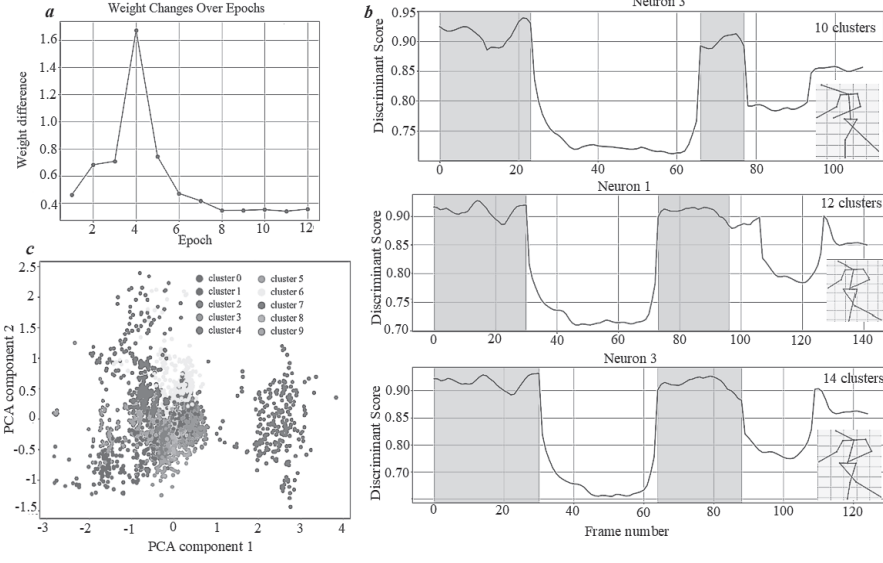


Fig. 1. Results for 10 clusters SOM: a – weight changes over training epochs, b – comparison of DS Score for 10, 12 and 14 cluster for a recognized posture, c – distribution of samples belonging to clusters for an example set of postures from the test actions set

shoulder-L/R (L/R: Left/Right), clavicle-L/R, elbow-L/R, wrist-L/R, hip-L/R, knee-L/R, and foot-L/R. Each cluster is visualized by projecting the high-dimensional data (57-dimensional feature) into a two-dimensional space using Principal Component Analysis (PCA). This dimensionality reduction retains most data variance, enabling a clearer visualization of clustering patterns. To compute the principal components of a dataset $\mathbf{X} \in \mathbb{R}^{n \times D}$, each row $\mathbf{x}_i$ is first mean-centred by subtracting the average feature vector $\bar{\mathbf{x}}$. From the mean-centered data matrix $\mathbf{X}'$, the covariance matrix $\mathbf{S}$ is calculated as $\mathbf{S} = \frac{1}{n-1} (\mathbf{X}')^\top \mathbf{X}'$. The eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_D$ of $\mathbf{S}$ (sorted by descending eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$) are the principal components. Selecting the top k components, one forms the matrix $\mathbf{U}_k = [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_k]$. Any mean-centred sample $\mathbf{x}'$ is then projected onto these components to obtain its low-dimensional representation $\mathbf{z} \in \mathbb{R}^k$ via $\mathbf{z} = \mathbf{U}_k^\top \mathbf{x}'$. This procedure captures the data's most important directions of variance, enabling dimensionality reduction and simpler cluster visualization.

5. Adjusting the Discriminant Score

Considering the equation (5), it is obvious that the DS value depends on the total number of clusters. A larger number of clusters causes a larger decrease in the DS of the BMU. To solve this problem, we proposed a modified formula in which the denominator is a constant factor independent of the number of neurons in the SOM configurations. Having a database with a larger number of actions, we selected several SOM configurations with different numbers of clusters that we considered suitable

for our dataset. For each SOM configuration $conf = \{1, 2, \dots\}$ that comprises M^{conf} neurons, we define the cosine distance between an input $\mathbf{f}_j$ and the m -th neuron centroid $\mathbf{cc}_m$ as $DIST(\mathbf{f}_j, \mathbf{cc}_m)$. The cosine distance was chosen in our study because it considers the orientation of body segments relative to each other. In human postures, the relative orientation of body parts (determined by joint angles) allows us to identify similar postures, even if the geometric dimensions of the body segments differ due to personal differences. We determine the maximum distance for each SOM configuration over the entire test dataset $d_{\max} = \max(DIST(\mathbf{f}_j), \mathbf{cc}_m)$. Next we compute the sum of these maximum distances across all SOM configurations: $\sum_{conf} d_{\max}$. The adjusted DS score is given by:

$$DS_m(\mathbf{f}_j) = 1 - \frac{a(\mathbf{f}_j, \mathbf{cc}_m)}{\sum_{conf} d_{\max}}. \quad (6)$$

To further validate the choice of the appropriate number of clusters for the full dataset, we proposed not only to apply the adjusted DS Score, which exhibits the highest magnitude for the BMU, but also to evaluate the second-best matching unit (2^{nd} BMU). Incorporating the 2^{nd} BMU allows for evaluating clusters' separability between the BMU and 2^{nd} BMU. A significant gap in their DS Scores indicates strong cluster separability, hence better clustering quality. Conversely, small differences suggest potential overlaps between clusters, highlighting areas where the SOM may need further refinement. Evaluating the DS scores of the BMU and the 2^{nd} BMU can also help assess the robustness of the SOM configuration on selecting the BMU. Additionally, the motivation for evaluating the 2^{nd} BMU extends beyond validating the BMU. By analyzing the 2^{nd} BMU, we gain insights into the proximity of data points to neighbouring clusters. This is particularly useful for identifying ambiguous points that may straddle multiple clusters, offering valuable feedback for refining SOM parameters such as the learning rate, neighbourhood radius, or initialization strategy. The BMU c_1 and 2^{nd} BMU c_2 can be obtained using their respective cosine distances as: $c_1 = \arg \min d(\mathbf{f}_j, \mathbf{cc}_m)$ and $c_2 = \arg \min_{m \neq c_1} d(\mathbf{f}_j, \mathbf{cc}_m)$.

6. Results and Evaluation of Clustering Quality

The 10, 12, and 14 clusters SOM configurations were trained using the training dataset until there was negligible weights update (see Fig. 1a). The obtained results demonstrated clear inter-cluster separability within each configuration based on their DS score (see Fig. 1b). A distinct boundary is observable, marking transitions between clusters, which is characterized by a steep gradient in the DS score plot. This indicates that the clustering process effectively distinguishes between different postures in an ongoing action. Furthermore, postures were recognized at two distinct instances for this specific action and for all SOM configurations. Although these postures were not ideally identical, they characterized the same semantic meaning, illustrating that all configurations successfully learned generalizable representations while preserving the topology of the data. When comparing the various configurations, it is evident that the 10-cluster configuration demonstrates the highest DS score, indicating that it achieved the most distinct cluster separation, as shown in Fig. 1c. This observation is further

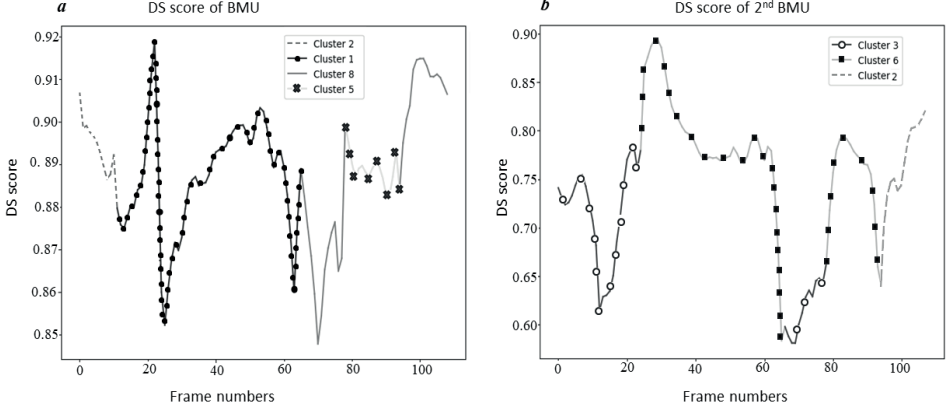


Fig. 2. Comparison of the modified DS score
a – The BMU vs b – the 2nd BMU for the 10-cluster SOM configuration for a tested human action

supported by Fig. 1a, where the training process for the 10-cluster configuration shows negligible weight updates, indicating good convergence and training stability. Additionally, Fig. 1b illustrates the clusters obtained for the 10-cluster configuration. After normalisation, the original data, which consisted of 57-dimensional feature vectors ranging from -1 to 1 , underwent PCA to obtain a reduced 2D representation. The 2D representation of the clustered postures shows their spatial distribution. Clusters were colour-coded for better visibility. The trained SOM successfully grouped related postures while maintaining the dataset's structure. When comparing the DS scores obtained from the 2nd BMU neuron against those from the BMU (see Fig. 2), several insights are emerging. First, because the SOM naturally classifies data based on a 'best match' principle, using the second-best match to compute scores yielded lower discriminant values, reflecting that the 2nd BMU captures postures that are less likely aligned with that cluster. This discrepancy highlights subtle ambiguities in the data, where certain samples may lie near the boundary between two closely related classes, leading to occasional misclassifications or situations in which a posture could be deemed equally similar to both the first and second winners. By exploring these overlaps, we gain a deeper understanding of how well our SOM-based clustering separates postures: a significant difference between the BMU and 2nd BMU DS score indicates robust, distinct clusters, whereas smaller gaps suggest borderline cases that may require a closer look or perhaps additional features to improve class separability. Ultimately, analyzing 2nd BMU DS scores along with BMU enriches our knowledge of how postures within the actions are distributed in the feature space, shows potential misclassification patterns, and hence provides opportunities for refining the clustering approach to achieve more precise and reliable representations of the underlying human action data.

7. Selection of the Number of Clusters

Introducing a constant denominator that integrates the maximum distances from all SOM configurations ensures a fair comparison among all configurations and that the relative quality of clustering is unbiased, i.e. the DS is independent of each configuration's specific structure. The commonly used clustering quality scores like the Silhouette coefficient, Dunn index, Davies-Bouldin index, and Calinski-Harabasz index provide a valuable method of cluster selection, but they usually focus on specific data features that can be important or not depending on the type of data and intended applications. Furthermore, they do not offer insights into posture similarities, cluster characteristics, boundaries and transitions. However, unlike these scores, the proposed DS provide valuable insights into these analyses. They continuously track how well each posture fits its assigned cluster relative to all others (expressed by their magnitudes), effectively showing how clustering confidence evolves over time and highlighting cluster boundaries, which other scores may overlook. This work thus supports the feasibility of an unsupervised SOM approach, where the DS score can be applied to refine decisions about the proper number of clusters and ensure that key postures of interest, possibly rare or complex, are not marginalized. This perspective is especially relevant for posture data, which often exhibit gradual, overlapping transitions rather than abrupt class separations. The comparison of BMU and 2^nd BMU DS scores is important for such situations because it further highlights subtle overlaps among posture clusters, where borderline postures may be nearly indistinguishable in high-dimensional space. This efficient method of cluster selection further reduces reliance on labelling, hence making the posture recognition methods more adaptive to new actions, more efficient, and easily scalable. This methodology offers a balanced solution regarding computational efficiency, recognition accuracy, and extensibility by reducing reliance on manual labelling and making the classification model more adaptive to new actions.

8. Conclusions

A key contribution is the introduction of a modified Discriminant Score (DS). The key improvement involves implementing a global normalization factor causing the DS score to be independent of the number of clusters, which ensures comparability of DS values across different SOMs configurations. Furthermore, the DS score not only evaluates clustering quality but also illustrates postural transitions and inter-cluster similarities, thereby adding a continuous perspective on how posture data evolve over time. This additional insight revealed borderline cases where postures were positioned near the boundaries between clusters. These transitional postures were further validated by analysing the difference between the first- and second-best matching neurons, providing a deeper understanding of cluster separability and posture continuity. The DS score further gives insights into the clustering quality, and with the help of this, we were able to decide on the most appropriate number of clusters, hence improving posture clustering performance while avoiding manual labelling, which requires domain expertise and is time-consuming. The results presented in this work highlighted our methodology's adaptability to unseen actions and laid the foundation for future work in which larger datasets and real-time implementations will extend the practical

applications of human posture recognition.

Acknowledgments. The authors express their sincere gratitude to Prof. Jacek Malec for supporting the research internship of the first author of this paper, as well as for his valuable discussions and advice on the work.

Literature

- [1] C. M. Huang and B. Mutlu, “Anticipatory robot control for efficient human-robot collaboration,” in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 83–90, 2016.
- [2] M. Kheirabadi, S. Keivanpour, Y. A. Chinniah, and J.-M. Frayret, “Human-robot collaboration in assembly line balancing problems: Review and research gaps,” *Computers Industrial Engineering*, vol. 186, p. 109737, 2023.
- [3] H. Buxbaum, S. Sen, and L. Kremer, “An investigation into the implication of human-robot collaboration in the health care sector,” *IFAC-PapersOnLine*, vol. 52, no. 19, pp. 217–222, 2019. 14th IFAC Symposium on Analysis, Design, and Evaluation of Human Machine Systems HMS 2019.
- [4] I. N. Weeraratna, D. Raymond, and A. Luharia, “Human-robot collaboration for healthcare: A narrative review,” *Cureus*, vol. 15, no. 11, 2023.
- [5] H. Khambhaita and R. Alami, “Assessing the social criteria for human-robot collaborative navigation: A comparison of human-aware navigation planners,” in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1140–1145, 2017.
- [6] V. Dutta, J. Cydejko, and T. Zielinska, “Improved competitive neural network for classification of human postures based on data from rgb-d sensors,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 17, pp. 15–28, Feb. 2024.
- [7] J. Xiaoyan *et al.*, “A survey on artificial intelligence in posture recognition,” *Comput Model Eng. Sci.*, vol. 137, no. 1, pp. 35–82, 2023.
- [8] G. Saleem, U. I. Bajwa, and R. H. Raza, “Toward human activity recognition: a survey,” *Neural Computing & Applications*, vol. 35, no. 5, pp. 4145–4182, 2023.
- [9] N. Ma *et al.*, “A survey of human action recognition and posture prediction,” *Tsinghua Science and Technology*, vol. 27, no. 6, pp. 973–1001, 2022.
- [10] Z. Liu *et al.*, “Stagg: Spatio-temporal adaptive graph pooling network for pedestrian trajectory prediction,” *IEEE Robotics & Automation Letters*, vol. 9, no. 3, pp. 2001–2007, 2024.
- [11] M.-F. R. Lee, Y.-C. Chen, and C.-Y. Tsai, “Deep learning-based human body posture recognition and tracking for unmanned aerial vehicles,” *Processes*, vol. 10, no. 11, p. 2295, 2022.
- [12] A. Caliskan, “Detecting human activity types from 3d posture data using deep learning models,” *Biomed. Signal Processing & Control*, vol. 81, 2023.
- [13] X. He, T. Zielinska, V. Dutta, T. Matsumaru, and R. Sitnik, “From seeing to recognising – an extended self-organizing map for human postures identification,” *IEEE Robotics and Automation Letters*, vol. 9, pp. 7899–7906, sep 2024.
- [14] L. E. Ekemeyong Awong and T. Zielinska, “Comparative analysis of the clustering quality in self-organizing maps for human posture classification,” *Sensors (Basel)*, vol. 23(18), 2023.

Recognition of Human Actions by Analyzing the Trajectories of Moved Objects

Jakub Kaminski¹, Vibekananda Dutta¹, Teresa Zielinska²

Abstract

Predicting object position changes during human action is useful for motion planning in robotics, safe human-robot interaction, and smart homes. The aim of this work is to analyze the motion trajectories of objects moved by humans and recorded as RGB images. These trajectories are associated with the corresponding human activities through AI models. The research contribution is: (1) designing a K-multidimensional clustering algorithm used to group normalized object motion trajectories and assign them to their corresponding clusters representing activities; (2) proposing a classifier-based convolutional neural network (CNN) that jointly learns state changes together with the corresponding object states. The classifier evaluates the transformations: initial object state – motion trajectories – final object state. On this basis, it is possible to predict subsequent object positions (states) based on previous states. A new dataset was created. The dataset consists of trajectories recorded for 30 human actions. A comparative study on the usefulness of popular clustering quality scores such as silhouette Score (SS) and Sum of Squared Errors (SSE) is presented, as well as a comparison of the proposed classification method with popular state-of-the-art methods using a new dataset. The results indicate that both developed approaches (clustering and classification models) perform well for the given task.

1. Introduction, State-of-the-Art and Problem Statement

People naturally understand events and the character of interactions based on the objects involved when observing human-object interactions and object manipulations. Human actions often influence changes in an object's state, such as drinking coffee, which changes the state of the cup's location. People can recognize these actions and the resulting changes in object states [1, 2]. The ability to anticipate object state-modifying transitions driven by human actions is crucial for various applications, including vision-based scene understanding, automated monitoring systems, action planning in robotics, and effective human-robot collaboration [4, 5, 6]. Consequently, a longstanding goal in human-robot interaction, particularly in assistive robotics research, is to develop methodologies and intelligent agents capable of understanding and structuring the world regarding its fundamental building blocks. It enables encoding scenes into latent space and reasoning about the large variation in object geometries [3]. However, the task is challenging due to the large amount and variability of existing object-action pairs and the subjective annotation of recorded data. Some pioneering work in action anticipation has combined motion and spatial representations

¹Faculty of Mechatronics, Warsaw University of Technology, św. Andrzeja Boboli 8, 02-525 Warsaw, (jakub.kaminski4.stud@pw.edu.pl, vibekananda.dutta@pw.edu.pl).

²Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, Nowowiejska 24, 00-665 Warsaw (teresa.zielinska@pw.edu.pl).

with object features. Several studies have explored action anticipation, often focusing on predicting the next possible human action in a sequence. Early approaches often relied on understanding the ongoing action and potentially the object affordances (i.e. human-object interactions) to infer future actions. For instance, methods like Conditional Random Fields (CRFs) [7] and probabilistic methods [8] have been used to predict future human actions based on object affordances and positions. However, these methods often relied on hand-crafted features and struggled with scalability to new tasks. More recent methods have acknowledged the strength of deep learning in learning more robust and generalizable features for action anticipation. Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTMs), have been widely used to model the temporal evolution of actions and interactions [9]. Some works have combined RNNs with Convolutional Neural Networks (CNNs) to process visual information and aggregate action sequence information over observed frames. Temporal aggregation using non-local blocks has also been explored [10]. These works emphasize the crucial role of human-object interactions (HOIs) in understanding and anticipating actions, especially those involving objects. Another related area is the understanding and anticipation of object position transitions. This involves recognizing object position transformations due to human actions (and manipulations). Researchers have started to focus on representing and interpreting object position (state) changes in videos considering the state-modifying actions [11, 12]. The problem of object state change anticipation has been recently introduced, focusing on predicting whether an object position change will occur and, if so, its possible characteristics. Approaches to object state transition anticipation often integrate learned visual features representing recent visual information with natural language features describing past object state changes and actions. Considering an object’s motion trajectories for understanding and predicting human-object interactions is also relevant. Capturing and analyzing the motion of objects during manipulation can provide valuable cues about ongoing and future actions [6]. The problem of Object State Classification (OSC) is defined as the multi-class recognition of the initial and final object states in video data that demonstrate state-modifying actions.

With the aforementioned ideas in mind, the contribution of our work is four-fold. First, comprehensive data processing is introduced, which delivers the pre-processing of objects’ motion trajectories by extracting 2D coordinates and applying a K-multi-dimensional clustering algorithm to segregate object motion trajectories corresponding to distinct human action scenarios. Second, an extended disjoint CNN model [13] is proposed to accommodate multi-variable object trajectory data. This generalization uses information from multiple object trajectories for prognosis. Third, a new dataset, consisting of 720 video records of distinct human activities, is introduced. Finally, comparative studies on the usefulness of popular clustering quality scores, such as the silhouette score (SS) and the sum of squared error (SSE), are presented, together with a comparison of our method with state-of-the-art methods on the newly recorded dataset.

2. The Dataset and Data Preprocessing

The dataset [15] used in this study comprises 30 different human activities involving object transfer and 16 types of objects involved in these activities. Each recorded

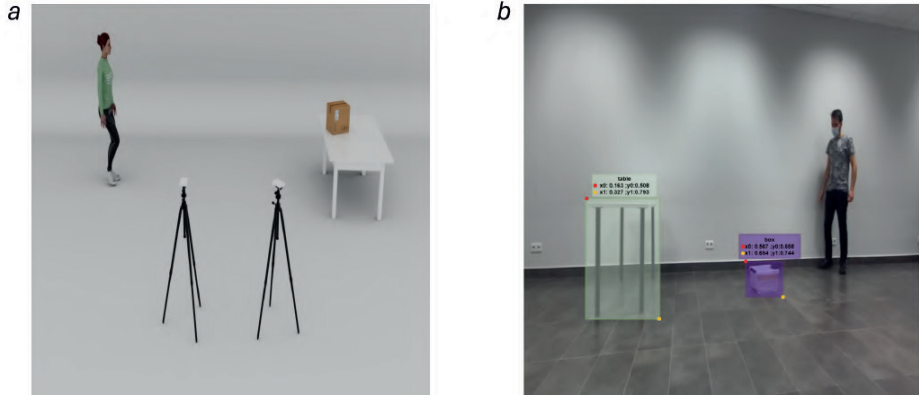


Fig. 1. Recording setup and example of extraction of object locations: a – two camera perspectives were used for each video recording (frontal view, 50-degree angle view), b – extracted 2D coordinates of recognized objects

data sample was analyzed to extract the motion paths (trajectories) of the objects transferred by humans and visible in the scene. In the presented research, the trajectories are represented in the planar view and described by the pixel coordinates in the recorded RGB image. Each extracted object trajectory should be assigned to a specific class, representing a scenario associated with a specific action and specific objects. Representative trajectories were extracted for each object type to visually inspect how objects are moved for each group class. The recordings for the dataset were conducted indoors under artificial lighting conditions, and the activities involved domestic tasks. To ensure a more diverse data collection, two camera perspectives were used for recording [15], as shown in Fig. 1a.

The initial data preparation stage involved annotating a subset of object instances within selected video data using the open-source platform Label Studio[†]. This manual annotation process focused on building the bounding boxes around objects that were involved in human activities. Subsequently, to automate the extraction of object locations across the entire dataset, a YOLOv7 model [14] was trained to learn robust visual features for detecting the target object categories and storing the bounding box coordinates (in 2D coordinate space) for the objects in each video, as shown in Fig. 1b. The semantic labels associated with each detected object were recorded with its spatial localization.

The motion trajectories and objects mapping. A trajectory τ^{o_j} of a detected object o_j represented in the 2D space is described as a time-ordered sequence of discrete positions across consecutive image frames in a video:

$$\tau^{o_j} = (p_1^{o_j}, \dots, p_N^{o_j}), \quad (1)$$

where $p_n^{o_j} = (x_n^{o_j}, y_n^{o_j})$ represents the o_j object's coordinates in frame n , and N is the total number of observations (frames) for this trajectory. Each trajectory τ^{o_j} thus

[†]Label Studio - <https://labelstud.io/>

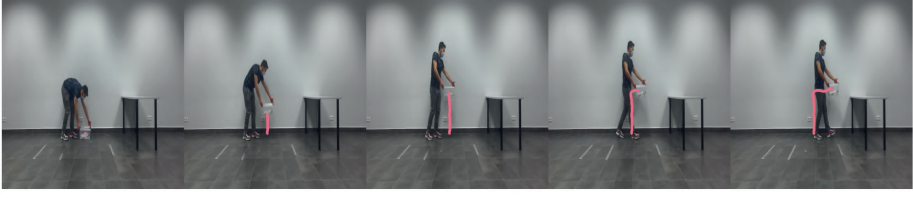


Fig. 2. An example of trajectory creation as the action progresses

forms a 2D time-series sequence describing the object's motion. Since multiple objects of the same type may be present in a frame, it is necessary to establish correct associations between the objects detected in consecutive frames. It is achieved by matching the objects according to the minimal distances of objects' positions identified in neighbouring frames. Let $\mathbf{p}_{n-1}^{o_j}$ represent the o_j ($j = 1, \dots, J$) object position in the previous frame. Of course, the number of objects in consecutive frames is not obviously equal. We denote the objects observed in the frame n by o_k ($k = 1, \dots, K$). For precision of considerations, it must be emphasized that the same value of lower superscript by the object in the frame $n - 1$ and n does not mean that there are the same objects. The aim of matching is to find such pairs of j and k from $\mathbf{p}_{n-1}^{o_j}$, ($j = 1, \dots, J$) and $\mathbf{p}_n^{o_k}$, ($k = 1, \dots, K$) that denote the same object. It is addressed as object mapping and is obtained considering the minimal distances. For each pair of $n - 1$, n frames, the pairs of j_{min} and k_{min} indexes denoting the same object are obtained by applying the minimum distance:

$$(j_{min}, k_{min}) = \arg \min_{j=1, \dots, J, k=1, \dots, K} d_{Euc}(\mathbf{p}_{n-1}^{o_j}, \mathbf{p}_n^{o_k}), \quad (2)$$

where d_{Euc} is denoting the squared Euclidean distance. As mentioned, the values of J and K can differ.

3. Methodology

Human action recognition can be done using either clustering or classification approaches. In the first case, each cluster represents a set of typical trajectories for a given action. In the training stage, the motion trajectories represent each action. After the training stage, each cluster ultimately represents a trajectory pattern typical for a given action. In the testing stage, testing trajectories are assigned to the created clusters (i.e. activities) based on the minimum distance metric between the pattern representing the cluster and the tested trajectory. This method is simple and refers to the intuitive understanding of similarity. However, the problem is the data heterogeneity because different people can perform the same activities at different speeds. Therefore, the normalization process is applied as it is described later. The next stage is forecasting the future part of the trajectory using the clustering results. For this purpose, the classifier is applied. Here, the learning stage uses labelled trajectories, i.e. trajectories assigned by earlier clustering to specific activities. In the clustering, the pattern trajectories are created. In the training phase, the classifier requires arbitrary labelled data.

3.1. Clustering process

The dataset for this study comprises human action recordings for various individuals, leading to inherent inter-subject variability in the realisation of actions. Therefore, the extracted objects' trajectories exhibit heterogeneous temporal lengths. To enable comparative analysis, **linear interpolation** was used to normalize the time dimension of the trajectories and produce the motion paths. Each index n of data point $\mathbf{p}_n^{oj}$ of each data trajectory was rescaled from natural number n to the real number r using the normalization $r = (n - 1)/(N - 1)$, where $N > 1$ and $r \in \langle 0, 1 \rangle$. The next stage was adjusting the density of data point indexes in this real domain to match the highest density of indexes obtained for the trajectory consisting of the greatest number of points N_{max} . Let us denote by r_{max} the indexes obtained for such trajectory ($r_{max} = \frac{n-1}{N_{max}-1}$, $n = 1, \dots, N_{max}$). For each recorded trajectory with $N < N_{max}$ points, the density of real indexes also means the density of re-indexed data points is lower. Therefore, the new, more dense data points indexed by r_{max} must be obtained. The formula for the data representing the new data point indexed by r_{max} ($\frac{n-1}{N-1} \leq r_{max} \leq \frac{n}{N-1}$) applies piecewise linear interpolation:

$$\mathbf{p}_{r_{max}}^{oj} = \mathbf{p}_n^{oj} + \Delta r_{max} \frac{\mathbf{p}_{n+1}^{oj} - \mathbf{p}_n^{oj}}{\Delta r}, \quad (3)$$

where $\Delta r = \frac{1}{N-1}$ and $\Delta r_{max} = r_{max} - \frac{n-1}{N-1}$. After normalizing the trajectories (converting them to the motion paths) to ensure uniform length and comparability, we applied a clustering approach to identify the path clusters. By grouping similar paths, we established a set of representative path scenarios. To achieve this, we employed the K-Multi-Dimensional Time Series Clustering (K-MDTSC) algorithm [16], chosen for its effectiveness in clustering multidimensional serial data. Two task-oriented modifications were introduced: a) instead of selecting initial centroids randomly, we select them directly from the dataset to minimize the likelihood of empty clusters, b) if an empty cluster occurred during the clustering process, the algorithm identifies the path farthest from the centroid within the largest cluster and assigns it as the centroid of the new cluster. A cluster-representative path is defined, besides establishing the cluster centroid. The representative path is computed as the mean of the data point coordinates of all normalized motion trajectories (paths) assigned to that cluster. This representative path provides an interpretable summary of the assigned to each cluster motion pattern, facilitating visualization and further analysis (see Fig. 3).

3.2. Classifier applied for future path prediction

This task uses a classification approach for assessing future object positions. The developed classifier extends the baseline Disjoint-CNN model [13]. Two task-oriented modifications were made – the extended classifier supports multiple entities simultaneously, and a cross-entity convolutional component was introduced along with temporal and spatial components, making the extended network. The network supports cross-object interactions. We address it as **Multi-Entity Disjoint-CNN** (ME D-CNN) model. ME D-CNN incorporates 3D convolution operations, allowing the network to process intra-trajectory features and inter-trajectory dependencies.

Input Tensor Construction. Each detected object oj belongs to a set of object types

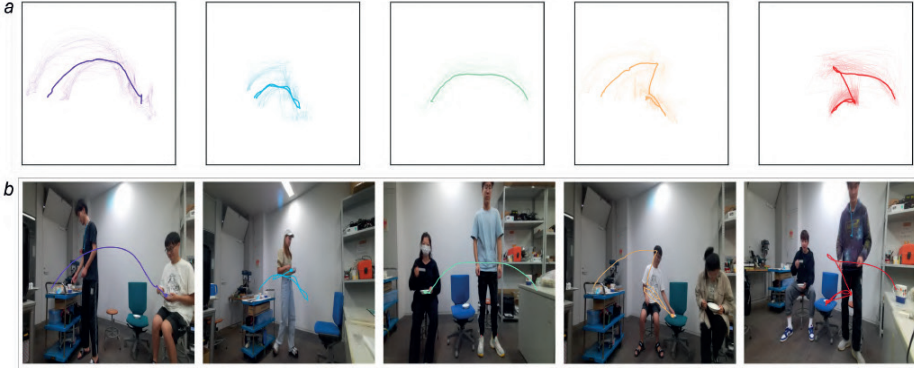


Fig. 3. Steps of grouping trajectories and assigning clusters: a – examples of mean trajectory belong to different clusters, b – illustration of representative trajectory of various activities

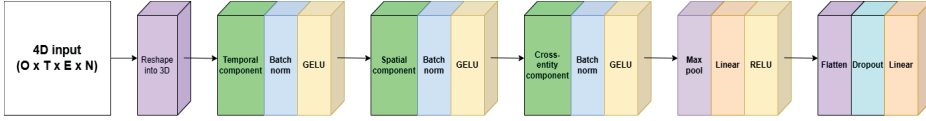


Fig. 4. Architecture of the proposed ME D-CNN

O , such that $o_j \in O$. Each object type is characterised by E distinct instances (entities), where each instance is a motion trajectory represented as a sequence of object positions in 2D space over N time steps. It means that for recorded action k , the input tensor is represented by $\mathbf{x}_k \in \mathbb{R}^{O \times T \times E \times N}$, where $T = 2$ denotes the planar dimension of the data. To support 3D convolutional operations in the proposed ME D-CNN architecture, we then reshape the 4D tensor by merging the spatial and entity dimensions - $\mathbf{x}_k \in \mathbb{R}^{O \times (E \cdot T) \times N}$. Such reshaped input allows the model to convolve across both spatial and entity dimensions simultaneously, facilitating the learning of intra- and inter-object temporal patterns. Our ME D-CNN model processes the reshaped data to predict object state transitions. The classifier function is expressed by:

$$\hat{\mathbf{y}}_k = f(\theta, \mathbf{x}_k), \quad (4)$$

where f denotes the ME D-CNN model, θ represents all learnable parameters of the model, $\hat{\mathbf{y}}_k$ is the predicted output tensor containing class probabilities. In our implementation, $\hat{\mathbf{y}}_k$ typically has the shape $\mathbb{R}^{O' \times E \times C}$, where O' is the number of object types for which predictions are made and C is the number of target classes. This formulation allows the network to anticipate object state transitions by effectively capturing spatiotemporal interactions across multiple object trajectories.

Model parameters. The architecture employs temporal convolutional components that capture trajectory dynamics by applying a convolution kernel of size $[1 \times 1 \times t]$, where t is a hyperparameter representing the temporal filter length. The spatial components capture the object-specific trajectory patterns using a kernel of size $[1 \times 2 \times 1]$

Tab. 1. Silhouette scores (SS) and Sum of Squared Errors (SSE) for k-shape, K-MDTSC, and KernelKMeans algorithms on different objects. **Bold font** indicates the best results

Object	No. Clusters	Silhouette Score (SS)			Sum of Squared Errors (SSE)		
		k-shape	K-MDTSC	KernelKMeans	k-shape	K-MDTSC	KernelKMeans
box	14	-0.079	-0.023	-0.083	21707	17474	21172
trolley	3	0.415	-0.075	0.032	5079	9075	9951
cup	6	-0.103	0.446	-0.034	6558	2088	7970
glass	3	0.000	0.133	0.046	3149	2353	2592
package	2	0.000	0.315	0.150	9729	7293	8673
kettle	2	0.000	0.713	0.283	375	94	341
phone	2	0.000	0.240	0.192	1214	822	1054
Average	-	0.033	0.250	0.083	6830	5599	7393

with a stride of 2. Finally, to extract cross-entity dependencies, a cross-entity component is introduced, which applies convolution with a kernel of size $[O \times E \times 1]$, where O and E correspond to the number of object types and entities, respectively. Batch normalization and Gaussian Error Linear Units (GELU) activation functions are used after each convolutional block to stabilize and improve learning. The architecture consists of three stacked convolutional modules, followed by a max pooling layer, a fully connected (linear) layer with ReLU activation, a flattening operation, and dropout regularization. Due to its multivariate nature, the model employs the softmax activation function in the output layer and is trained using the cross-entropy loss function to support multi-class probability distributions. The architecture is shown in Figure 4.

4. Experiments and Results

A comparative evaluation was conducted to validate our network’s design and assess the object’s trajectory recognition quality when moving different objects. We also examined the clustering quality for object trajectories.

Clustering approach. The clustering quality was evaluated separately for the trajectories corresponding to each object type. For this purpose, the results delivered by K-MDTSC were compared with the results of two other clustering methods: KernelKMeans [18] and k-shape [19]. Each algorithm was set to perform 200 iterations with randomly initialized centroids. Each algorithm was run up to 10 times to mitigate the influence of random initializations. Following the ground truth, the same number of clusters was used in all cases. We compared the results using the clustering quality scores: the silhouette score (SS), which measures the coherence of the clusters, and the sum of squared errors (SSE), which quantifies the compactness of the clusters [17]. The results of the SS and SSE scores are presented in Table 1.

K-MDTSC performed the best on most objects, achieving the best average silhouette score across all objects, which is 0.250 (a higher value means better clustering). The k-shape algorithm does not provide more than one non-empty cluster for the objects: glass, package, kettle, and phone. The KernelKMeans algorithm performed worse than K-MDTSC across all objects except for the trolley, where the performance difference was negligible in the context of the overall silhouette score results. Similar to the silhouette scores, the K-MDTSC algorithm achieved the best performance across all objects except for the trolley, where it scored 4321, compared to 6743 for k-shape and 7253 for KernelKMeans, which exhibited the worst performance. It should

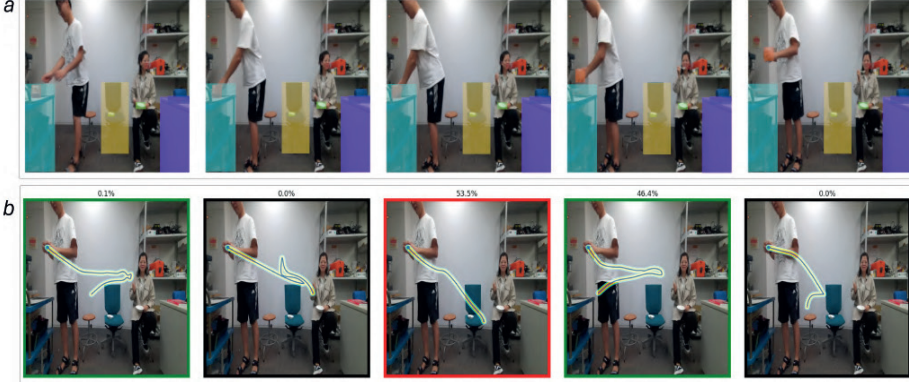


Fig. 5. All active object states and their possible state-modifying paths in an action scenario – ‘preparing food and eating together with a colleague’: a – object state, b – object paths.

be emphasized that Table 1 presents the results of 7 objects out of 16 objects as discussed in section 2. However, the experimental results presented in Table 2 were assessed considering all the possible objects (out of 16) involved in the test set.

Tab. 2. Comparison of evaluation metrics on chosen models on 20% variant of the Dataset

Dataset [20% used]	ResNet	LSTM-FCN	MVTS	Disjoint-CNN	ME D-CNN (ours)
Accuracy	0.435	0.697	0.688	0.759	0.931
Cohen’s kappa	0.300	0.613	0.616	0.698	0.917
Cross entropy loss	1.817	0.903	1.386	1.505	0.737
MPCE	0.142	0.105	0.136	0.093	0.005
Avg. inference time [ns]	343	200	21588	14054	12519

Prognosing results. To assess the proposed network capabilities, we also performed a benchmark comparison of the proposed ME D-CNN architecture against the baseline model and several popular classifiers that support multivariate data. The classifiers include ResNet [20], LSTM-FCN [21], and MVTS [22]. We used four key evaluation metrics: a) accuracy, b) Cohen’s Kappa, c) mean per class error (MPCE), and d) average inference time (in ns) for qualitative comparison.

The results presented in Table 2 demonstrate that the proposed ME D-CNN architecture achieved the highest accuracy (0.931), Cohen’s kappa (0.917), and the lowest MPCE (0.005) when only 20% of the test data were used for inference. The Disjoint-CNN model followed closely, achieving a second-highest accuracy of 0.759. LSTM-FCN model shows significant improvement (accuracy: 0.697, cross-entropy loss: 0.903), enhancing its competitiveness. MVTS also performed reasonably well (0.688 accuracy) but exhibited the highest inference time (21588 ns). ResNet achieved the lowest accuracy (0.435). We also visualised the test samples describing the anticipation of the object state and its state transitions. Figure 5 provides examples of these anticipation scenarios for the action ‘preparing food and eating together with a colleague’. The top row displays representative illustrations of the list of active objects

in this action, as depicted in sample frames in Figure 5a. The bottom row highlights the method’s capability, which delivers the motion scenario probabilities and future object positions in the form of trajectories (see Fig. 5b). Here, the ground truth trajectory and the most probable predicted scenario are typically highlighted (e.g. in a red frame), with other next highly probable scenarios also presented (e.g. in a green frame).

5. Conclusions

This paper presents human action recognition using a clustering approach with normalised motion trajectories (paths) and proposes a classification-augmented convolutional neural network for path prediction. This classification network predicts the state transitions of an object during human action. The basic idea is to learn the states (positions) of moving objects with respect to the sequential aspect and the transitions of these states. To achieve this, we first used the K-dimensional clustering algorithm to group the motion paths of objects based on object types, effectively categorising the possible path patterns for each object type, which is associated with the action. Then, we used a classification approach to estimate the future positions of objects. We developed a Multi-Entity Disjoint-CNN (ME D-CNN) architecture to extend the basic D-CNN. The results showed both good clustering quality (achieving the highest average silhouette score of 0.250) and high classification accuracy (0.931) compared to selected state-of-the-art methods. ME D-CNN also features a competitively short inference time, which makes it practical for real-world applications. Future work will focus on adaptive clustering techniques and on the use of heterogeneous data sets, such as spatial trajectories of different movement durations, without applying the normalisation. The research scenarios will be extended, e.g. by including hand-to-hand transfer of objects. Research is important for planning human-robot interactions and the actions of robots that assist humans.

References

- [1] B. M. Lake, et al. Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, 40:e253, 2017
- [2] J.B.Alayrac, et al., Joint Discovery of Object States and Manipulation Actions, in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2146–2155, 2017.
- [3] Wu, Z., et al. Slotformer: Unsupervised Visual Dynamics Simulation with Object-Centric Models. *arXiv preprint arXiv:2210.05861*, 2022.
- [4] Roy, D., & Fernando, B. Action Anticipation Using Pairwise Human Object Interactions and Transformers. *IEEE Transactions on Image Processing*, 30, 8579-859, 2021.
- [5] Wan, W., et al. Learn to Predict How Humans Manipulate Large-sized Objects from Interactive Motion. *IEEE Robotics and Automation Letters* 7.2 (2022): 4702-4709.
- [6] Ni, Z., et al. Human–Object Interaction Prediction in Videos Through Gaze Following. *Computer Vision and Image Understanding* 233 (2023): 103741.
- [7] Jiang, Y., and Saxena, A. Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent CRFs. *Robotics: Science and systems*.

2014.

- [8] Dutta, V., and Zielinska, T. Activities Prediction using Structured Database. 12th Int. Workshop on Robot Motion and Control (RoMoCo). IEEE, 2019.
- [9] Roy, D., & Fernando, B. Action Anticipation using Latent Goal Learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2745-2753), 2022.
- [10] Bach, P., et al. The Affordance-Matching Hypothesis: How Objects Guide Action Understanding and Prediction. *Frontiers in Human Neuroscience*, 8, 254, 2014.
- [11] Wang, T., et al. Learning Human-Object Interaction Detection Using Interaction Points. Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. 2020.
- [12] Manousaki, V., et al. Anticipating Object State Changes. arXiv preprint arXiv:2405.12789, 2024.
- [13] Foumani, S. N. M., et al. Disjoint-CNN for Multivariate Time Series Classification. In International Conf. on Data Mining Workshops (ICDMW) (pp. 760-769), 2021, IEEE.
- [14] Wang, C. Y., et al. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF conf. on Computer Vision and Pattern Recognition (pp. 7464-7475), 2023.
- [15] Dutta, V., Cydejko, J., & Zielinska, T. Improved Competitive Neural Network for Classification of Human Postures Based on Data from RGB-D Sensors. *J. of Automation, Mobile Robotics & Intelligent Systems*, 17(3), 2023.
- [16] Giordano, D., et al. K-MDTSC: K-Multi-Dimensional Time-Series Clustering Algorithm. *Electronics*, 10(10), 1166, 2021.
- [17] Thinsungnoena, T., et al. The Clustering Validity with Silhouette and Sum of Squared Errors. *Learning*, 3(7), 44-51, 2015.
- [18] Liu, X. Simplemkkm: Simple Multiple Kernel k-Means. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(4), 5174-5186, 2022.
- [19] Paparrizos, J., & Gravano, L. K-Shape: Efficient and Accurate Clustering of Time Series. In Proceedings of the 2015 ACM SIGMOD Int. Conf. on Management of Data (pp. 1855-1870), 2015.
- [20] Wu, Z., Shen, C., & Van Den Hengel, A. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *Pattern Recognition*, 90, 119-133, 2019.
- [21] Karim, F., et al. LSTM fully Convolutional Networks for Time Series Classification. *IEEE Access*, 6, 1662-1669, 2017.
- [22] Zerveas, G., et al. A Transformer-Based Framework for Multivariate Time Series Representation Learning. Proceedings of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining. 2021.

Badania potrzeb informacyjnych użytkowników systemów AMR*

Jakub Strzałka¹, Grzegorz Granosik¹

Streszczenie

Celem artykułu jest prezentacja wyników badań potrzeb informacyjnych użytkowników systemów zarządzania flotą autonomicznych robotów mobilnych (ang. Autonomous Mobile Robot – AMR). W badaniach wykorzystano metody jakościowe (wywiady indywidualne, definiowanie person) oraz ilościowe (sortowanie kart, testy A/B), co pozwoliło zidentyfikować kluczowe dane i optymalne formy ich wizualizacji. Wyniki badań wskazują na konieczność personalizacji interfejsów pod kątem roli użytkownika, integracji zaawansowanej diagnostyki w czasie rzeczywistym oraz wykorzystania wizualnych sygnałów statusu, co przekłada się na zwiększenie efektywności operacyjnej i satysfakcji użytkowników.

1. Wprowadzenie

Systemy zarządzania flotą autonomicznych robotów mobilnych (ang. Autonomous Mobile Robot – AMR) znajdują coraz szersze zastosowanie w przemyśle, jednak kwestie związane z User Experience (UX) i User Interface (UI) dla takich systemów są wciąż słabo zbadane. Dobre zaprojektowanie interfejsu może znacząco podnieść efektywność pracy operatorów oraz ich zadowolenie [1], [2].

Praca koncentruje się na zrozumieniu potrzeb informacyjnych użytkowników systemów zarządzania flotą AMR oraz określeniu sposobów prezentacji danych dla różnych grup użytkowników. Celem badań jest identyfikacja kluczowych informacji oraz ustalenie optymalnych metod ich wizualizacji, aby usprawnić kontrolę nad flotą robotów i poprawić doświadczenie użytkownika.

Aby osiągnąć ten cel, przeprowadzono pogłębione wywiady indywidualne (IDI) z głównymi użytkownikami systemów AMR i zdefiniowano na ich podstawie persony, zastosowano sortowanie kart do zaprojektowania architektury informacji oraz wykonano testy A/B porównujące istniejący interfejs z makietą zbudowaną na bazie zebranych danych. W kolejnych rozdziałach przedstawiono szczegółowo metody, przebieg eksperymentów, analizę wyników oraz wnioski z badań, ze szczególnym uwzględnieniem praktycznych implikacji dla projektowania interfejsów AMR [3].

2. Wywiady pogłębione z użytkownikami

Badania rozpoczęto od *pogłębionych wywiadów indywidualnych (In-Depth Interviews, IDI)* z kluczowymi użytkownikami systemu. Celem wywiadów było zdobycie dogłęb-

* Artykuł związany z wdrożeniem wyników projektu POIR.04.01.04-00-0017/17 finansowanego przez Narodowe Centrum Badań i Rozwoju.

¹ Instytut Automatyki, Politechnika Łódzka, ul. Stefanowskiego 18, 90-537 Łódź, Polska; Email: jakub-strzalkaa@gmail.com, granosik@p.lodz.pl

nej wiedzy o potrzebach informacyjnych użytkowników systemów AMR — **jakie informacje są dla nich najważniejsze, w jakiej formie chcą je otrzymywać, w jakich sytuacjach ich potrzebują** [3].

2.1. Uczestnicy wywiadów

Do wywiadów zaproszono **cztery osoby** pełniące różne role związane z obsługą i rozwojem systemów AMR. Wśród respondentów znaleźli się:

- (A) główny inżynier oprogramowania odpowiedzialny za uruchomienie i utrzymanie systemu (70 min rozmowy),
- (B) operator produkcji codziennie pracujący z robotami (40 min),
- (C) główna programistka systemu floty AMR (60 min),
- (D) menedżer firmy odpowiedzialny za sprzedaż i kontakt z klientami (65 min).

Taki dobór uczestników zapewnił zróżnicowane perspektywy: od użytkownika końcowego, przez twórcę technologii, po decydenta biznesowego. Wywiady przeprowadzono osobiście w siedzibie firmy wdrażającej system AMR, co sprzyjało swobodnej dyskusji w kontekście rzeczywistych warunków pracy. Wszystkie rozmowy nagrano (za zgodą uczestników) i szczegółowo przeanalizowano.

2.2. Analiza jakościowa

Analiza jakościowa zebranych wypowiedzi pozwoliła zidentyfikować powtarzające się tematy i potrzeby informacyjne wspólne dla różnych użytkowników. Respondenci zgodnie podkreślali znaczenie szybkiego dostępu do *aktualnych, szczegółowych danych* o stanie robotów i systemu. Jako **krytyczne informacje** wymieniano:

- stan naładowania baterii każdego robota,
- bieżący status zadań i aktywności robotów,
- wszelkie alarmy i ostrzeżenia o błędach,
- dane diagnostyczne (np. obciążenie CPU, logi) do rozwiązywania problemów.

Operatorzy akcentowali potrzebę prostego podglądu najważniejszych parametrów w czasie rzeczywistym, natomiast osoby techniczne chciały mieć możliwość **dostępu do szczegółowych danych** w razie potrzeby. Ponadto zauważono, że menedżerowie oczekują bardziej zagregowanych wskaźników wydajności floty, podczas gdy inżynierowie potrzebują surowych danych technicznych.

2.3. Persony użytkowników

Na podstawie wywiadów opracowano **persony użytkowników**, czyli archetypiczne profile reprezentujące kluczowe grupy odbiorców systemu. Persony pomagają uogólnić wyniki jakościowe i skupić projektowanie na konkretnych potrzebach i celach użytkowników. Przygotowano trzy persony odzwierciedlające różne role i wymagania: jedną *personę pierwszorzędną* oraz dwie *drugorzędne*. Persony zostały zdefiniowane w oparciu o rzeczywiste dane z wywiadów (wzorce zachowań, cele, bolączki użytkowników) oraz uzupełniające dane demograficzne [3].

Persona pierwszorzędna „Jan Nowak” (Rys. 1) reprezentuje doświadczonego *menedżera zakładu produkcyjnego*, który strategicznie nadzoruje flotę robotów AMR. Jego priorytetem są wysokopoziomowe wskaźniki efektywności, terminowe raporty oraz szybki wgląd w ogólny stan produkcji; ceni więc pulpit z kluczowymi statystykami, czytelne wizualizacje trendów i możliwość generowania raportów na żądanie.



WIEK
ZAWÓD
EDUKACJA
DOŚWIADCZENIE

45
Menedżer Zakładów Produkcyjnych
Magister Zarządzania i Inżynierii Produkcji
20 lat doświadczenia w zarządzaniu zakładami produkcyjnymi

Biografia
Janusz rozpoczął swoją karierę jako inżynier produkcji zaraz po ukończeniu studiów. Dzięki determinacji i umiejętnościom menedżerskim szybko awansował na stanowisko menedżera zakładu produkcyjnego.

Przyzwyczajenia

Codziennie rano przegląda raporty produkcyjne, regularnie uczestniczy w spotkaniach zespołu, śledzi wskaźniki wydajności produkcji.

Potrzeby

Narzędzia do monitorowania i optymalizacji procesów produkcyjnych, wsparcie w szybkim diagnozowaniu i rozwiązywaniu problemów, możliwość generowania raportów na żądanie.

Oczekiwania

Intuicyjny interfejs użytkownika, szybki dostęp do kluczowych wskaźników wydajności (KPI), alerty o problemach w czasie rzeczywistym.

Frustracje

Trudności w integracji nowych technologii z istniejącymi systemami, brak natychmiastowego dostępu do kluczowych informacji, opóźnienia w raportowaniu problemów technicznych.

Kompetencje techniczne i doświadczenie

-  Zaawansowane umiejętności korzystania z komputera i oprogramowania biurowego.
-  Biegły użytkownik, regularnie korzysta z internetu do śledzenia trendów w branży.
-  Używa smartfona do zarządzania i komunikacji w pracy, ma podstawowe umiejętności obsługi tabletu.

Osobowość

- Zdeterminowany
- Odpowiedzialny
- Otwarty na technologie

Hobby

- Gra w tenisa
- Wędrówki górskie

PERSONA
Jan Nowak
"Nie ma rzeczy niemożliwych, są tylko wyzwania, które czekają na rozwiązanie."

Rys. 1. Persona pierwszorzędna - Jan Nowak

Persony drugorzędne uzupełniają obraz użytkowników systemu: „*Michał Krawczyk*” odzwierciedla potrzeby *operatora systemu AMR*, skupionego na bieżącej kontroli zadań i natychmiastowych alertach z hali produkcyjnej, natomiast „*Anna Wysoczek*” reprezentuje *inżyniera/technika utrzymania*, dla którego najważniejsze są szczegółowe dane diagnostyczne, logi błędów i narzędzia do szybkiego uruchamiania testów serwisowych.

2.4. Wnioski

Wyniki wywiadów stworzyły bazę wiedzy o rzeczywistych potrzebach użytkowników. Ujawniono **kluczowe obszary informacyjne**, które system powinien obsługiwać (stany techniczne, zadania, alerty, statystyki) oraz wskazano wstępne preferencje co do form prezentacji (np. potrzeba prostego, „czytelnego na pierwszy rzut oka” statusu robotów vs. możliwości zagłębienia się w dane). Te ustalenia stały się punktem wyjścia do kolejnego etapu badań – strukturyzacji informacji i projektowania interfejsu przy użyciu metody sortowania kart.

3. Sortowanie kart

Metoda **sortowania kart (card sorting)** została wykorzystana, aby zaprojektować intuicyjną strukturę informacji interfejsu w oparciu o mentalny model użytkowników.

Pozwoliło to odpowiedzieć na pytanie, *jak użytkownicy grupują poszczególne elementy informacji* i które elementy wydają im się ze sobą powiązane lub ważne. Dodatkowo posłużyło zbadaniu *preferencji wizualnych* – jakiego typu reprezentacje (wykres, tabela, ikonka itp.) użytkownicy przypisaliby do danego rodzaju informacji [3].

3.1. Przygotowanie badania

Na podstawie wyników wywiadów zidentyfikowano kilkadziesiąt kluczowych elementów informacyjnych, które potencjalnie powinny znaleźć się w panelu zarządzania flotą (np. "Poziom baterii robota", "Bieżące zadanie", "Historia zadań", "Alert bezpieczeństwa", "Logi systemowe" itp.). Dla każdego elementu przygotowano kartę z nazwą i opisem. Dodatkowo przygotowano zestaw kart reprezentujących różne formy wizualizacji danych (np. wykres liniowy, tabela, mapa, ikona). Przed właściwym badaniem przeprowadzono test pilotażowy, aby upewnić się, że nazewnictwo kart jest zrozumiałe dla uczestników.

3.2. Przeprowadzenie badania

Rekrutację uczestników skierowano do osób zaznajomionych z systemem – zaproszono zarówno osoby biorące udział w wywiadach, jak i dodatkowych potencjalnych użytkowników systemów AMR. Ostatecznie do badania przystąpiło **6 uczestników** (spośród 10 zaproszonych). Sesje odbywały się **zdalnie**, z wykorzystaniem interaktywnego szablonu w narzędziu FigJam, co umożliwiło swobodne grupowanie kart metodą "przeciągnij i upuść". Każdy uczestnik samodzielnie grupował karty informacyjne w kategorii według własnego uznania, **nadawał nazwy utworzonym grupom**, a następnie przypisywał do każdej karty preferowaną formę wizualizacji (spośród przygotowanych). Cała sesja dla jednej osoby zajmowała około 30–45 minut i była moderowana w minimalnym stopniu – uczestnik na głos komentował swoje decyzje, co pozwoliło zebrać dodatkowe jakościowe uzasadnienia.

3.3. Wyniki sortowania kart

Zebrane dane przeanalizowano, identyfikując wspólne wzorce. Mimo indywidualnych różnic, analiza klastrowa pogrupowała karty w kilka **logicznych kategorii informacji**. Przykładowo, użytkownicy konsekwentnie łączyli ze sobą elementy dotyczące **stanu technicznego i wydajności robota** – tworząc grupę zawierającą m.in. poziom baterii, obciążenie podzespołów, statystyki wydajności i historię operacji. Osobno grupowano elementy związane z **bieżącymi zadaniami i trasami** robotów, a oddzielnie wszelkie **alerty, błędy i ostrzeżenia**. Wyniki te wskazują, że interfejs powinien być podzielony na moduły odpowiadające tym właśnie obszarom (np. osobny widok dla stanu i diagnostyki robotów, osobny dla zadań, osobny dla alertów itp.) – tak aby informacje powiązane ze sobą z punktu widzenia użytkownika były dostępne w jednym miejscu.

Drugim istotnym wynikiem była informacja o **preferowanych sposobach wizualizacji danych** dla poszczególnych typów informacji. Uczestnicy często dobierali podobne formy prezentacji. I tak: **dane statystyczne w czasie** (np. zużycie energii w czasie, wydajność) większość osób przyporządkowała do **wykresów liniowych lub słupkowych, listy elementów lub kategorie** (np. lista robotów, lista zadań) – do **tabel lub prostych list tekstowych, proste statusy i ostrzeżenia** – do **ikon ze zmiennym kolorem** (sygnalizacja stanu) bądź **wskaźników**. Informacje o **lokalizacji** (np. po-

zycje robotów, mapy obszaru) jednoznacznie łączono z **wizualizacją na mapie**. Za bardzo przydatne uznano również **kolorystyczne kodowanie** poziomu baterii i statusów błędów – intuicyjnie zielony oznaczałby prawidłowy stan, żółty ostrzeżenie, czerwony alarm. Pojawiły się też ciekawe sugestie, np. by krytyczne alerty były zawsze widoczne na górze ekranu w formie czerwonego banera lub ikony powiadomienia.

3.4. Wnioski

Wnioski z sortowania kart stały się bezpośrednią podstawą do zaprojektowania układu i wyglądu interfejsu użytkownika. Na ich podstawie opracowano **architekturę informacji** dla systemu zarządzania flotą autonomicznych robotów mobilnych: wyodrębniono główne sekcje interfejsu (np. *"Panel systemu"* z ogólnym podsumowaniem stanu floty, *"Zarządzanie robotami"* ze statusem poszczególnych robotów, *"Zarządzanie mapą"*). Każda sekcja miała prezentować dane odpowiednio pogrupowane zgodnie z modelami mentalnymi użytkowników. Przy projektowaniu wizualnym interfejsu kierowano się preferencjami użytkowników co do form prezentacji danych – stąd przewaga wykresów dla prezentacji trendów, tabel dla danych szczegółowych oraz wyraźnych ikon/kolorów dla komunikatów wymagających uwagi [4].

4. Testy A/B prototypu interfejsu

Ostatnim etapem badań było **badanie porównawcze A/B**, mające na celu **ilościową ocenę** opracowanych rozwiązań projektowych. Przygotowano **prototyp nowego interfejsu (Wariant B)** w postaci interaktywnej makiety uwzględniającej ustalenia z wywiadów i sortowania kart. Został on zestawiony z **obecnie używaną aplikacją (Wariant A)** do zarządzania flotą robotów w firmie, aby sprawdzić, czy nowy interfejs zapewnia lepszą użyteczność. Testy A/B skupiały się na trzech kluczowych kryteriach oceny: **skuteczności**, **wydajności** oraz **satysfakcji użytkownika** [3].

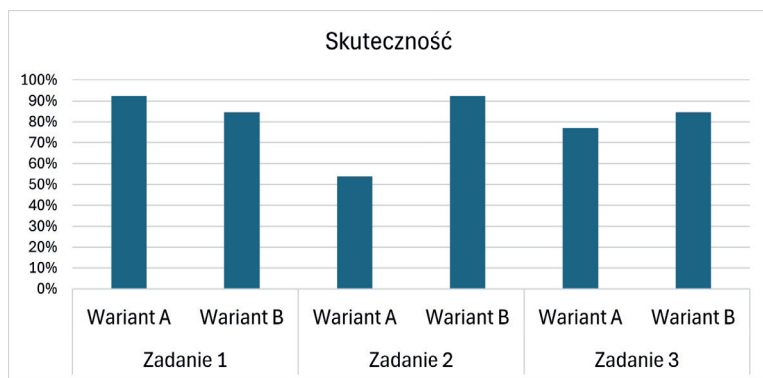
- **Skuteczność** mierzono jako odsetek poprawnie zrealizowanych zadań przez użytkowników (czyli czy użytkownik był w stanie wykonać dane zadanie w interfejsie).
- **Wydajność** oceniano poprzez czas potrzebny na wykonanie zadania (im krótszy, tym lepiej).
- **Satysfakcję** badano standaryzowanym kwestionariuszem **SUS (System Usability Scale)** wypełnianym przez uczestników po skorzystaniu z interfejsu.

4.1. Scenariusz testu

Uczestnicy mieli do wykonania serię typowych zadań w aplikacji do zarządzania flotą AMR. Przykładowe zadania obejmowały:

1. sprawdzenie statusu konkretnego robota i ocenę, czy wymaga on interwencji,
2. zlokalizowanie na interfejsie informacji o ostatnich błędach lub alertach,
3. wyciągnięcie statystyk wydajności floty za określony okres.

Każdy uczestnik wykonywał te zadania przy użyciu **Wariantu A** (istniejącej aplikacji) lub **Wariantu B** (nowej makiety). Mierzono sukces/porażkę zadania oraz czas realizacji. Po zakończeniu zadań uczestnik wypełniał ankietę **SUS** oceniając interfejs.



Rys. 2. Skuteczność dla obu wariantów

4.2. Skuteczność

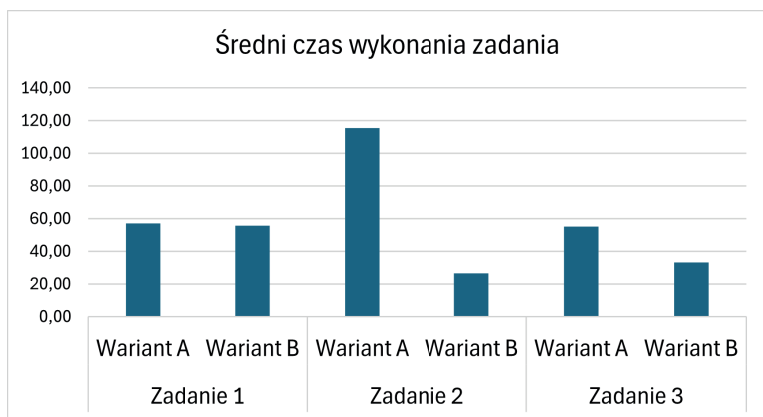
Nowy interfejs (Wariant B) ogólnie umożliwił użytkownikom skuteczniejsze wykonywanie zadań niż stara aplikacja, co widać na Rys. 2. Dla dwóch z trzech testowych zadań odsetek poprawnych realizacji był wyższy w wariancie B. Szczególnie **zadanie 2** (bardziej złożone, wymagające przeanalizowania danych diagnostycznych) wykazało dużą różnicę: użytkownicy korzystający z nowego interfejsu osiągnęli **95% skuteczności**, podczas gdy w starym było to tylko **55%**. Sugeruje to, że zaproponowany interfejs znacznie lepiej wspiera wykonywanie trudniejszych operacji, prawdopodobnie dzięki lepszemu organizowaniu informacji. Dla **zadania 1** (prostszego) oba interfejsy poradziły sobie podobnie – skuteczność wyniosła ok. 85–92% (nieznacznie lepszy okazał się tu wariant A). W **zadaniu 3** różnica była niewielka na korzyść wariantu B (85% vs 77%). Analiza statystyczna (test χ^2) potwierdziła istotność zaobserwowanych różnic dla zadania 2, co oznacza, że przewaga nowego interfejsu nie była przypadkowa.

4.3. Wydajność

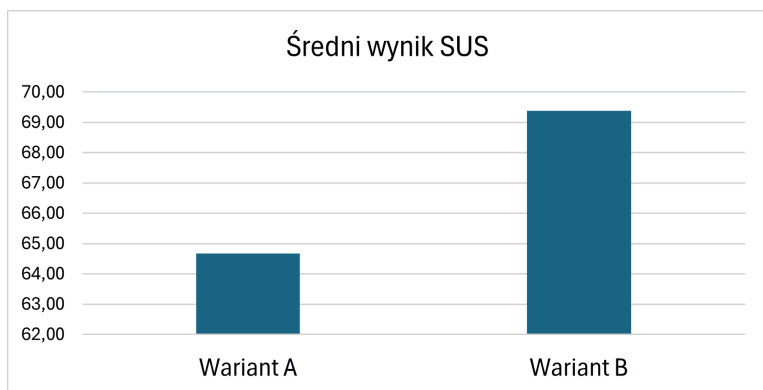
Średni czas potrzebny na ukończenie zadań okazał się **krótszy w nowym interfejsie** dla wszystkich scenariuszy (Rys. 3). Szczególnie przy zadaniu złożonym (zadanie 2) użytkownicy wykonywali średnio w ciągu 30s w wariancie B, co skróciło czas o kilkadziesiąt procent w stosunku do wariantu A (średni czas realizacji równy 110s). Ogólna poprawa wydajności wynika z usprawnionej nawigacji i czytelniejszego rozmieszczenia elementów w makiecie. Uczestnicy rzadziej błędzili po menu i ekranach – nowy design zgrupował powiązane funkcje, dzięki czemu zmniejszyła się liczba kroków potrzebnych aby dotrzeć do informacji.

4.4. Satysfakcja użytkowników

Wyniki ankiety użyteczności SUS również wskazały przewagę wariantu zaprojektowanego na podstawie badań (pokazane na Rys. 4). **Średni wynik SUS** dla nowej makiety wyniósł **69,4 pkt**, podczas gdy dla obecnej aplikacji **64,7 pkt**. Różnica ta sugeruje, że użytkownicy postrzegali nowy interfejs jako nieco bardziej użyteczny



Rys. 3. Średni czas wykonania zadań



Rys. 4. Średni wynik kwestionariusza SUS

i przyjazny. W skali interpretacji SUS wynik ok. 65 oznacza użyteczność przeciętną, a ok. 70 – dobrą, choć nie znakomitą. Nowy interfejs zbliżył się więc do oceny *dobrej*. W komentarzach użytkownicy chwalili przede wszystkim **przejrzystość układu** i **estetykę** wariantu B, wskazywali jednak, że brak pewnych funkcjonalności (była to wszak prototypowa makietka) ułatwiał orientację – sugerowali, że pełna implementacja powinna zachować prostotę. Zauważono, że **Wariant A** – jako dojrzały system produkcyjny – oferował więcej funkcji i ustawień, co z jednej strony zwiększa możliwości, ale z drugiej komplikowało interfejs (stąd niższy SUS). **Wariant B** był ograniczony do kluczowych funkcjonalności, przez co oceniono go wysoko pod kątem łatwości obsługi, lecz ocena ta może nie uwzględniać wyzwań, jakie niesłaby pełna wersja zaawansowanego systemu. Mimo to, trend jest jednoznaczny – uproszczony, przemyślany interfejs spotkał się z lepszym przyjęciem użytkowników.

5. Dyskusja

Przeprowadzone badania dostarczyły wielu cennych spostrzeżeń na temat interakcji człowiek–robot w kontekście systemów przemysłowych. **Studium przypadku** floty robotów mobilnych pokazało, że **projekt interfejsu HRI** (Human–Robot Interaction) powinien być silnie oparty na realnych potrzebach różnych grup użytkowników. Wykazano, iż w środowisku przemysłowym występuje duże zróżnicowanie ról użytkowników systemu AMR, co przekłada się na odmienne wymagania informacyjne. *Operatorzy liniowi* potrzebują interfejsów nastawionych na **szybką percepcję stanu** i natychmiastową reakcję – cenią elementy wizualne działające w czasie rzeczywistym, intuicyjną nawigację, minimalizm informacji (tylko to, co potrzebne tu i teraz). *Inżynierowie i specjaliści techniczni* oczekują możliwości **diagnozy i kontroli szczegółów** – interfejs musi dawać dostęp do logów, statystyk, trybów zaawansowanych, nawet jeśli są one ukryte dla przeciętnego użytkownika. Z kolei *kadra menedżerska* poszukuje **wysokopoziomowych wskaźników** i trendów – dla nich ważne są dashboards z podsumowaniami efektywności floty, wykresy wydajności, raporty historyczne. **Personalizacja interfejsu** pod te role wydaje się kluczowym kierunkiem – możliwość dostosowania widoku do potrzeb konkretnego użytkownika może znacząco zwiększyć zarówno efektywność pracy, jak i satysfakcję [5], [6], [7].

Wyniki testów A/B potwierdziły, że **wdrożenie zasad UX** i najlepszych praktyk projektowania interfejsu przekłada się na wymierne korzyści: użytkownicy szybciej i skuteczniej wykonują zadania, rzadziej popełniają błędy, lepiej oceniają pracę z systemem. Poprawa o **40 punktów procentowych** w skuteczności wykonania złożonego zadania (z 55% na 95%) czy skrócenie czasu realizacji o ~36% to znaczne usprawnienia osiągnięte dzięki badaniom UX. Oznacza to, że inwestycja czasu w analizę potrzeb użytkowników i iteracyjne projektowanie z *użytkownikiem w pętli* może przynieść **realne oszczędności czasowe i redukcję kosztów operacyjnych**. Co istotne, wzrosła też satysfakcja użytkowników mierzona skalą SUS – wskazuje to, że poza twardymi metrykami wydajności, poprawiło się również **postrzeganie systemu** przez ludzi, co może przełożyć się na lepszą akceptację technologii AMR w miejscu pracy [5].

Badania te wpisują się w szerszy nurt **Human-Centered Design** w robotyce przemysłowej. W literaturze HRI podkreśla się, że uwzględnienie czynnika ludzkiego jest krytyczne dla sukcesu wdrożeń robotów współpracujących. Nasze wyniki dostarczają *studium przypadku* na potwierdzenie tej tezy: pokazaliśmy krok po kroku, jak zastosować metody HCI/UX (wywiady, persony, card sorting, prototypowanie, testy A/B) w kontekście rzeczywistego systemu robotycznego i jak przełożyć wyniki na konkretne ulepszenia interfejsu. Uzyskane wnioski mogą być wartościowe zarówno dla badaczy zajmujących się **interakcją człowiek–robot (HRI)**, jak i dla praktyków – projektantów UX i inżynierów tworzących interfejsy dla zaawansowanych systemów automatyki [2], [6], [7], [8].

6. Wnioski

Podsumowując, badanie wykazało szereg kluczowych spostrzeżeń dotyczących projektowania interfejsów dla systemów AMR:

- **Intuicyjny interfejs zwiększa efektywność pracy z robotami:** Użytkownicy do-

ceniają klarowną prezentację informacji i prostą nawigację, co umożliwia szybkie podejmowanie decyzji operacyjnych. Nowy interfejs uprościł wykonywanie zadań, redukując czas i liczbę błędów.

- **Zaawansowane funkcje diagnostyczne są niezbędne dla niezawodności:** Dostęp do szczegółowych danych technicznych (logi, diagnostyka w czasie rzeczywistym) umożliwia inżynierom sprawne rozwiązywanie problemów i podnosi ogólną niezawodność systemu.
- **Dostosowanie interfejsu do roli użytkownika zwiększa satysfakcję:** Możliwość personalizacji widoku pod kątem potrzeb operatora, inżyniera czy menedżera wspiera efektywniejszą pracę. Użytkownicy chcą widzieć przede wszystkim te informacje, które są istotne dla ich funkcji – reszta powinna być dostępna na żądanie.
- **Dynamiczne wizualizacje i oznaczenia wspomagają orientację w systemie:** Wykorzystanie elementów takich jak *wykresy aktualizowane w czasie rzeczywistym, ikony statusów, kolorowe alerty czy mapy* znacząco poprawia orientację w stanie systemu i przyspiesza reakcje na zdarzenia. Zwłaszcza w środowisku czasu rzeczywistego ma to krytyczne znaczenie.

Powyższe wnioski stanowią istotny wkład w obszar projektowania interakcji człowiek–robot. Przeprowadzone badania mają charakter pilotażowy, co oznacza, że wskazane trendy i preferencje wymagają potwierdzenia na większych grupach użytkowników w kolejnych badaniach. Warto zaznaczyć, że nawet przy niewielkiej liczbie uczestników udało się zidentyfikować konkretne, praktyczne wskazówki projektowe, które już na obecnym etapie mogą znacząco wpłynąć na poprawę interakcji użytkowników z systemami AMR. Wyniki uzyskane w testach A/B jednoznacznie wskazują na potencjał projektowania zorientowanego na użytkownika, co podkreśla wartość przeprowadzonego studium przypadku.

Pomimo tych ograniczeń, przedstawione studium przypadku ma wyraźną wartość edukacyjną i innowacyjną, ukazując potencjał podejścia zorientowanego na użytkownika w kontekście autonomicznych robotów mobilnych. Wskazuje to na potrzebę dalszych, bardziej rozbudowanych badań, które pozwolą potwierdzić przedstawione wnioski w szerszym zakresie. W przyszłości warto kontynuować ścisłą współpracę interdyscyplinarną – łącząc wiedzę inżynierską z metodami badań UX/HRI – aby tworzyć rozwiązania efektywne i akceptowalne przez użytkowników, którzy z tymi systemami na co dzień pracują.

Literatura

- [1] T. Meneweger et al., „Factory Workers’ Ordinary User Experiences: An Overlooked Perspective,” *Human Technology*, pp. 209-232, 2018.
- [2] D. Wurhofer et al., "Insights from User Experience Research in the Factory", *Proceedings of Conferences*, 2016.
- [3] I. Mościchowska, B. Rogoś-Turek, *Badania User Experience jako podstawa projektowania*, Warszawa: Wydawnictwo Naukowe PWN SA, 2015.
- [4] Interaction Design Foundation - IxDF. “What is Information Architecture (IA)?” Interaction Design Foundation - IxDF. <https://www.interaction-design.org/literature/topics/information-architecture/> (accessed Apr. 17, 2025).
- [5] J. Strzałka, *Badania potrzeb informacyjnych użytkowników systemów autono-*

micznych robotów mobilnych, Praca magisterska, Politechnika Łódzka, 2025.

- [6] E. Prati et al., How to include User eXperience in the design of Human-Robot Interaction, Robotics and Computer-Integrated Manufacturing, 2021.
- [7] J. Hažík et al., Fleet Management System for an Industry Environment, Journal of Robotics and Control, 2022.
- [8] E. G. Ortiz et al., Fleet Management System for Mobile Robots in Healthcare Environments, Journal of Industrial Engineering and Management, 2021.

Research on the information needs of users of AMR systems

The aim of the article is to present the results of research into the information needs of users of fleet management systems for Autonomous Mobile Robots (AMR). The research employed qualitative methods (individual interviews, persona definition) and quantitative methods (card sorting, A/B testing), enabling the identification of key data and optimal forms of their visualization. Research findings highlight the necessity of customizing interfaces according to user roles, integrating advanced real-time diagnostics, and employing visual status indicators, which translates into increased operational efficiency and user satisfaction.

PLANOWANIE RUCHU ROBOTÓW

Analityczne wsparcie metody pól potencjałów w zadaniach planowania ruchu

Ignacy Duleba¹

Streszczenie

W referacie przedstawiono analityczne wyrażenia umożliwiające wyznaczenie wypadkowej siły odpychającej od naładowanego jednorodnym ładunkiem brzegu obiektów elementarnych (głównie segmenty liniowe) oraz działającej na punktowego robota w przestrzeni dwuwymiarowej. Wyznaczenie siły jest niezbędne w metodach planowania ruchu robotów mobilnych bazujących na popularnej metodzie sztucznych pól potencjałów. Wskazano jak przyspieszyć proces obliczania siły przez transformację zadania z globalnego układu współrzędnych do wyróżnionego układu lokalnego związanego z każdym z obiektów elementarnych. Rozważania zilustrowano przykładowymi symulacjami.

1. Wstęp

Metoda sztucznych pól potencjałów, MSPP, jest jednym z najbardziej popularnych sposobów planowania ruchu robotów mobilnych w środowiskach kolizyjnych [2, 3]. Jest metodą ogólnego przeznaczenia, ma interpretację fizyczną, jest łatwo algorytmizowalna oraz może być stosowana nawet wtedy, gdy jedynie lokalna informacja o otoczeniu robota jest dostępna. Ideą metody jest krótkookresowe wyznaczenie ruchu robota na podstawie interakcji siły przyciągającej robota do celu i wypadkowej z sił odpychających od przeszkód w jego otoczeniu. W przeciwieństwie do niektórych metod planowania, nie nakłada wymagań dotyczących szczególnego kształtu przeszkód (np. metoda grafu widoczności akceptuje jedynie przeszkody będące wielokątami) oraz może także wspomagać sterowanie reaktywne. Brak czułości MSPP na kształt przeszkód wynika z faktu, że zawsze istnieje możliwość rozlokowania odpowiedniej liczby punktowych ładunków na brzegu przeszkody odpychających robota i zastosowanie superpozycji ich wpływu. Jednak ten zabieg dodatkowo wprowadza minima lokalne funkcji potencjałów.

Idea MSPP może być implementowana w dwojaki sposób: albo explicite są wyliczane wirtualne siły pochodzące od przeszkód i działające na robota, albo wyliczane jest pole potencjałów, a siły w danym punkcie są antygradientem funkcji potencjału (funkcje nawigacyjne). Dla uproszczenia i przyspieszenia obliczeń, robot zwykle jest traktowany jako punkt materialny, po odpowiednim rozszerzeniu przeszkód. Metoda posiada jednak także wady, wynikające z lokalnych minimów funkcji nawigacyjnych, uniemożliwiających osiągnięcie celu bez dodatkowych zabiegów. Istnieje wiele wariantów MSPP uwzględniające modelowanie obiektów ruchomych z predykcją ich położenia [5], wygładzanie brzegów przeszkód nieregularnych przy użyciu funkcji sigmoidalnych [6], zastosowanie innych funkcji nawigacyjnych [8], czy też biorące

¹Katedra Cybernetyki i Robotyki, Politechnika Wrocławska,
ul. Janiszewskiego 11/17, 50-372 Wrocław, ignacy.duleba@pwr.edu.pl.

pod uwagę dynamiczne charakterystyki robota (maksymalne dopuszczalne prędkości, przyspieszenia, promienie skrętu, czy ograniczoną mobilność robota nieholonomicznego w pewnych kierunkach). Metody potencjałowe mogą być stosowane dla wielu klas robotów mobilnych: szybujących, dronów, ale najczęściej do kołowych robotów mobilnych (czasem wyposażonych w efektor). Zawsze jednak podstawowym podzadaniem MSPP będzie wyznaczenie siły/potencjału oddziałującego na robota, a wynikającego z każdej z przeszkód.

Głównym przedmiotem rozważań referatu jest efektywne i analityczne wyliczenie sił generowanych przez elementarne obiekty, lub obiekty przez nie aproksymowane. Zalety analitycznego podejścia w porównaniu z metodami numerycznymi są wielorakie, zarówno efektywnościowe jak i implementacyjne. Ich listę (niekoniecznie dotyczącą rozważanego zadania) przytoczono poniżej.

1. Dość często w metodach optymalizacyjnych są pożądane charakterystyki analityczne typu (anty-)gradient czy hesjan dla przyspieszenia zbieżność algorytmów. Jakkolwiek możliwa jest ich numeryczna aproksymacja, jednak zawsze wymaga dodatkowych obliczeń i jest przybliżeniem rzeczywistości. Szczególnie aspekt efektywnościowy jest widoczny w metodach iteracyjnych, gdzie zyski obliczeniowe w jednym kroku obliczeń są zwielokrotniane liczbą kroków niezbędnych do uzyskania rozwiązania.
2. Korzystając z metod analitycznych istnieje możliwość wyznaczania punktów szczególnych funkcji (minima/maksima, miejsca zerowe). Zerowanie się wyznacznika odpowiedniej macierzy w zadaniach kinematycznych dla manipulatorów pozwala na wyznaczenie oraz unikanie konfiguracji osobliwych.
3. Analityczne wzory umożliwiają szybsze obliczenia, bo podstawienie do wzoru jest obliczeniowo mniej kosztowe niż symulacja numeryczna odpowiednika wzoru.
4. W MSPP przez zastosowanie podejścia analitycznego unika się problemu odwzorowania brzegu obiektu jako zbioru naładowanych punktów (i problemu decyzyjnego: ilu i jak rozłożonych). Jeden wzór analityczny pozwala oddać dokładnie oddziaływanie nieskończonej liczby punktów.

W praktycznych przypadkach zaleca się, by metody numeryczne stosować jedynie wtedy, gdy analityczne zawodzą.

Referat zorganizowany jest następująco: w rozdziale 2 przedstawiono ogólny algorytm umożliwiający wyliczenie analitycznej zależności sił generowanych w danym (symbolicznie zadanym) punkcie przez elementarne obiekty. W rozdziale 3 wyliczono analitycznie siły dla jednorodnie naładowanego odcinka (dla kilku wybranych parametrów warunkujących siłę odpychającą) oraz przedyskutowano dla okręgu oraz części (łuku) okręgu. W rozdziale 4 symulacyjnie zilustrowano różnicę modelowania punktowego przeszkód od ciągłego oraz aproksymację okręgu N-kątem foremny. W rozdziale 5 zebrano wnioski końcowe.

2. Siły odpychające generowane przez obiekty elementarne

Wypadkowa siła odpychająca $\mathbf{f} = (f_x, f_y)^T$, działająca na punkt $\mathbf{p} = (x_a, y_a)^T \in \mathbb{R}^2$ opisana w globalnym układzie współrzędnych, a powodowana naładowanym brzeżem obiektu zadanego *explicit* jako $\mathbf{c}(x) = (x, y(x))^T$ (def. 1) lub w postaci parametrycznej $\mathbf{c}(s) = (x(s), y(s))^T$ (def. 2), w układzie globalnym/lokalnym(obiektu), jest

równa

$$\mathbf{f} \stackrel{(\text{def. 1})}{:=} \int_{x_{\min}}^{x_{\max}} \frac{k}{\|\mathbf{p} - \mathbf{c}(x)\|^r} \cdot \frac{\mathbf{p} - \mathbf{c}(x)}{\|\mathbf{p} - \mathbf{c}(x)\|} dx \stackrel{(\text{def. 2})}{:=} \int_{s_{\min}}^{s_{\max}} \frac{k}{\|\mathbf{p} - \mathbf{c}(s)\|^r} \cdot \frac{\mathbf{p} - \mathbf{c}(s)}{\|\mathbf{p} - \mathbf{c}(s)\|} \cdot \frac{\partial \mathbf{c}(s)}{\partial s} ds, \quad (1)$$

gdzie $\|\cdot\|$ oznacza normę euklidesową, $k \in \mathbb{R}_+$ jest dodatnim współczynnikiem wagowym, a r dodatnim parametrem (zwykle równym wielokrotności 1/2) warunkującym siłę wzrostu wartość oddziaływania wraz ze zbliżaniem do brzegu przeszkody. Dalej, dla uproszczenia zapisu, współczynnik wagowy przyjęto jako $k = 1$. Zakres całkowania należy odpowiednio dobrać w zależności od opisu obiektu. W definicjach (1) założono, że krzywa $\mathbf{c}$ jest jednoznaczna, tj. dla każdej dopuszczalnej wartości argumentu x istnieje dokładnie jedna wartość y . Gdyby było inaczej, krzywą można zdekomponować na części spełniające założenie, a następnie stosować zasadę superpozycji.

Jedynym wymaganiem na wartości sił odpychających jest ich monotoniczny wzrost wraz ze zmniejszaniem odległości robota od przeszkody i osiąganie wartości bliskich nieskończoności na jej brzegu. Zwykle w praktyce jest stosowana kulombowska interakcja między naładowanymi punktami (robot i każdy wybrany punkt na brzegu przeszkody) odpowiadająca wartości parametru r równego 2. Pierwszy czynnik w równaniu (1) określa amplitudę siły (potencjał), a drugi wskazuje unormowany kierunek jej działania. W przypadku wyliczania jedynie potencjału, drugi czynnik należy pominąć.

Opis w globalnym układzie współrzędnych jest uniwersalny, bowiem stanowi platformę określenia wzajemnej relacji robota i jego otoczenia (przeszkód). Jednak posiada poważną wadę w postaci zwykle skomplikowanego opisu analitycznego przeszkód, który można znacznie uprościć przez rozważenie sił w lokalnym (związany z każdą konkretną przeszkodą) układem współrzędnych, a następnie transformując siły do układu globalnego, aby były porównywalne z siłami pochodzącymi od innych przeszkód. Zabieg transformacji do układu lokalnego obiektu jest szczególnie efektywny, gdy wykorzystuje się symetrię obiektu. Nakłady obliczeniowe na transformacje: układ globalny-lokalny-globalny są niewielkie. Ten sposób działania opisuje para algorytmów 1, 2, która egzemplifikuje ogólną ideę transformowania problemu do prostszej, kanonicznej, postaci, której przykłady w teorii sterowania (linearyzacja), sprowadzanie kinematyk do postaci normalnych [10] czy wyrażanie układów w postaci łańcuchowej [1] są liczne. Algorytm 2 przygotowuje w trybie off-line wzory do obliczeń, które są z kolei wykorzystywane podczas jednokrokowego ruchu rzeczywistego robota realizowanego przez Algorytm 1.

3. Analityczne formuły opisujące siły odpychające generowane przez obiekty elementarne

W rozdziale przedstawiono wynikowe analityczne formuły służące do wyliczenia siły $\mathbf{f} = (f_x, f_y)^T$ oddziaływania elementarnych obiektów na punktowego robota, zgodnie ze wzorami (1). Jako obiekty elementarne przyjęto segment liniowy (podrozdział 3.1) i okrąg lub jego część (podrozdział 3.2). Wartości sił przedstawiono dla kilku wybranych wartości parametru r , zob. (1), który warunkuje sposób karania za zbliżanie się robota do przeszkody. Do obliczeń wykorzystano intensywnie pakiet obliczeń symbolicznych Wolfram Mathematica [11], dla którego alternatywą może być Maple.

Algorytm 1 Generowanie siły pochodzącej od obiektów elementarnych, a wywieranych na robota znajdującego się w zadanym położeniu p_g^n w układzie globalnym

Dane: Scena w $\mathbb{R}^2$ z przeszkodami oraz zadane (numerycznie) bieżące położenie punkтового robota p_g^n w układzie globalnym.

Krok 1. (wykonywany jednokrotnie, przy pierwszym wywołaniu Algorytmu 1). W trybie off-line, zdekomponuj każdą przeszkodę na obiekty elementarne i dla każdego z nich wykonaj Algorytm 2 uzyskując symboliczną wartość siły dla i -tego obiektu $F_{gi}(p_g)$.

Krok 2. Podstaw za wartość symboliczną p_g zadane numerycznie, bieżące położenie robota p_g^n i wykorzystując zasadę superpozycji wyznacz nieskończony kierunek ruchu robota w punkcie p_g^n wskazywany przez wektor

$$F(p_g^n) = \sum_i F_{gi}(p_g \leftarrow p_g^n) + F_{att}(p_g^n), \quad (2)$$

gdzie $F_{att}(p_g^n)$ jest siłą przyciągającą robota w kierunku celu.

3.1. Segment liniowy

Segmenty liniowe są najbardziej uniwersalnym sposobem modelowania brzegu przeszkód w przestrzeni dwuwymiarowej. W naturalny sposób tworzą wielokąty, które są jedynie akceptowalne przez niektóre metody planowania ruchu (graf widoczności [2]), a dla innych metod ułatwiają obliczenia (diagram Woronoia [2]).

Istnieje wiele sposobów reprezentowania segmentów liniowych w układzie globalnym [3], różniących się liczbą parametrów. Jednak reprezentacja zadana przez punkt środkowy odcinka, jego długość i orientację względem osi x jest najbardziej efektywna w wyliczeniu transformacji odcinka do układu lokalnego, rys. 1a.

Wszystkie dane w tej sekcji x_a, y_a, f_x, f_y są wyrażone w układzie lokalnym. Przyjmując oznaczenia:

$$x_- := a - x_a, \quad x_+ := a + x_a, \quad \beta_- := x_-^2 + y_a^2, \quad \beta_+ := x_+^2 + y_a^2, \quad (6)$$

wypadkowa siła $(f_x, f_y)^T$ działająca w punkcie $(x_a \geq 0, y_a \geq 0)$ a spowodowana jednorodnie naładowanym segmentem liniowym (a – oznacza połowę jego długości) umieszczonym na osi x i symetrycznie względem osi y , zob. rys. 1a, będzie przedstawiona dla kilku wartości parametru r zdefiniowanego po wzorze (1).

Dla $r = 2$ i następującej notacji dodatkowej

$$\xi_- := \beta_-^{-1/2}, \quad \xi_+ := \beta_+^{-1/2}, \quad (7)$$

otrzymujemy dla $y_a > 0$ wartości sił

$$f_x = \xi_- - \xi_+, \quad f_y = (x_- \xi_- + x_+ \xi_+) / y_a, \quad (8)$$

natomiast gdy $y_a = 0$ i $x_a > a$, wtedy

$$f_x = -(1/x_+ + 1/x_-), \quad f_y = 0. \quad (9)$$

Algorytm 2 Analityczne wyznaczenie siły pochodzącej od obiektu elementarnego działającego na punktowego robota

Dane: położenie $\mathbf{p}_g$ (symbolicznie) punktowego robota i przeszkoda elementarna, wyrażone w układzie globalnym.

Krok 1. Z obiektem elementarnym skojarz lokalny układ współrzędnych ułatwiający jego parametryczny opis, który względem układu globalnego jest przesunięty o wektor $(x_l, y_l)^T$ i obrócony (względem osi x) o kąt α_l . Transformacje położenia punktów, i sił, między układami są następujące [4]

$$\mathbf{T}_g^l = \text{Trans}(x, x_l) \cdot \text{Trans}(y, y_l) \cdot \text{Rot}(z, \alpha_l), \quad (3)$$

$$\mathbf{T}_l^g = \text{Rot}(z, -\alpha_l) \cdot \text{Trans}(y, -y_l) \cdot \text{Trans}(x, -x_l). \quad (4)$$

Krok 2. Załóż (w postaci symbolicznej) pewien punkt $\mathbf{p}_l$ w układzie lokalnym, w którym hipotetycznie znajduje się robot.

Krok 3. Wylicz w postaci analitycznej siłę $\mathbf{F}_l(\mathbf{p}_l)$ działającą na punkt $\mathbf{p}_l$ w układzie lokalnym, wykorzystując właściwe oprogramowanie symboliczne.

Wynik: siła opisana analitycznym wzorem i sparametryzowanym $\mathbf{p}_g$ wyrażona jako

$$\mathbf{F}_g(\mathbf{p}_g) = \mathbf{T}_g^l \cdot \mathbf{F}_l(\mathbf{p}_l) = \mathbf{T}_g^l \cdot \mathbf{F}_l(\mathbf{T}_l^g \mathbf{p}_g). \quad (5)$$

Dla $r = 3$ i $y_a > 0$:

$$f_x = \frac{2ax_a}{\beta_- \beta_+}, \quad f_y = (y_a(x_-/\beta_- + x_+/\beta_+) + \arctg(x_-/y_a) + \arctg(x_+/y_a)) / (2y_a^2). \quad (10)$$

Dla przypadku szczególnego, $y_a = 0$ i $x_a > a$, mamy

$$f_x = \frac{2ax_a}{x_-^2 x_+^2}, \quad f_y = 0. \quad (11)$$

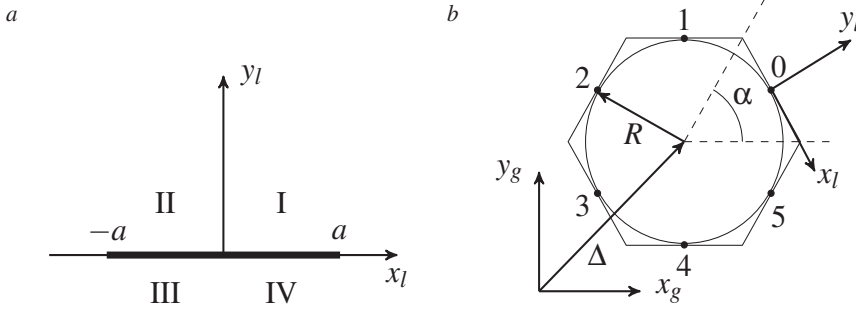
Dla wartości $r = 4$ i $y_a > 0$:

$$f_x = (\beta_-^{-3/2} - \beta_+^{-3/2})/3, \quad f_y = (x_- \beta_-^{-3/2} (2\beta_- + y_a^2) + x_+ \beta_+^{-3/2} (2\beta_+ + y_a^2)) / (3y_a^3), \quad (12)$$

a dla przypadku szczególnego: $y_a = 0$ i $x_a > a$ otrzymujemy

$$f_x = -(1/(x_-)^3 + 1/(x_+)^3)/3, \quad f_y = 0. \quad (13)$$

Łatwo zauważyć wzrost skomplikowania wyrażeń dla rosnących wartości r . Dla parzystych r wyrażenia nie zawierają odwrotności funkcji, a często występują wyrażenia do potęgi $-(r-1)/2$. Przypadki szczególne występujące przy zerowaniu mianowników, $y_a = 0$, w wyrażeniach (8), (10), (12) rozważono oddzielnie. Jednak dla bardzo małych wartości y_a należy oczekiwać złego uwarunkowania numerycznego. Z obliczeniowego punktu widzenia wartość $r = 2$ wydaje się najlepsza.



Rys. 1. a – odcinek w lokalnym układzie współrzędnych; b – okrąg i N-kąt foremny dla $N = 6, \psi = 0$; w układzie globalnym. Przedstawiono także układ lokalny dla segmentu 0

3.2. Okrąg lub jego część (łuk)

Do wyliczenia analitycznego siły pochodzącej od okręgu zgodnie z parametrycznym wzorem (1) okrąg o promieniu R scentrowano w punkcie $(0,0)^T$ a robota punktowego umieszczonego w lokalnym układzie współrzędnych w strefie poza przeszkodą $\mathbf{p} = (0, y_a > R)^T$, zob. rys. 1b. Obliczenia sił zaprogramowano i wykonano dla wartości parametru $r = 2, 3, 4, 5$, jednak nie będą tu przytaczane, bowiem ich postać analityczna jest zbyt obszerna do prezentacji. Na bazie ich analizy można sformułować parę spostrzeżeń:

1. dla nieparzystych r występują funkcje $\arctg$; dla parzystych r – funkcje eliptyczne typu E i F zdefiniowane [11] jako

$$E(\alpha, m) = \int_0^\alpha \sqrt{1 - m \sin^2(\theta)} d\theta, \quad F(\alpha, m) = \int_0^\alpha \frac{1}{\sqrt{1 - m \sin^2(\theta)}} d\theta, \quad \sqrt{1 - m \sin^2(\theta)}, \quad (14)$$

2. dla sprawdzanych wartości r , stopień wielomianu (zmiennych y_a, R) w mianowniku jest $(r - 1)$ razy większy niż w liczniku.

W praktycznych przypadkach stosowane jest chętnie podejście uproszczone, czyli zamiast wyliczać oddziaływanie okręgu na punkt wylicza się oddziaływanie skompresowanego okręgu do punktu leżącego w jego środku, a punktowego robota przesuwającego się w kierunku środka okręgu o promień okręgu. Wyliczenia są praktycznie natychmiastowe, gdyż kierunek oddziaływania siły jest skierowany wzdłuż odcinka łączącego środek okręgu (punkt) z położeniem robota, a siła wyliczana z oddziaływania punkt-punkt.

Wyliczenie całki (1) dla przypadku łuku (segmentu okręgu) jest jeszcze bardziej skomplikowane niż dla okręgu. Zatem dla segmentów okręgu można stosować dwie strategie:

1. albo aproksymować funkcje eliptyczne wielomianami (łatwo całkowne),
2. albo aproksymować segmentami liniowymi. Ta metoda wydaje się bardziej elastyczna, z możliwością wpływu na dokładność aproksymacji.

Dla przykładu: w globalnym układzie współrzędnych okrąg o promieniu R scentrowany w punkcie $\Delta := (\Delta x, \Delta y)^T$ może być aproksymowanym N-kątem foremnym. Wie-

lokąt z kątem $\alpha := 2\pi/N$, początkowo obrócony (przeciwnie do ruchu wskazówek zegara) o kąt ψ , zob. rys. 1b, jest określony przez punkty środkowe segmentów

$$(\mathbf{x}_{mid})_i = (\Delta x + R \cos(\alpha/2 + i\alpha + \psi), \Delta y + R \cos(\alpha/2 + i\alpha + \psi))^T, \quad i = 0, \dots, N-1 \quad (15)$$

i odpowiadające im kąty obrotu

$$\rho_i := (\alpha/2 + i\alpha + \psi) - \pi/2. \quad (16)$$

Połowa długości segmentów liniowych jest równa $a := R \tan(\alpha/2)$.

Podczas aproksymacji obiektu odcinkami warto stosować następujące reguły:

1. zmienną reprezentację przeszkód. Oczywiście, gdy robot jest daleko od przeszkody, jej wpływ można pominąć, redukując koszty obliczeniowe. Gdy jednak do przeszkody się przybliża, wtedy pewne części przeszkody (najbliższe robotowi powinny być uwzględniane) decydując jak wiele segmentów i jak rozmieszczonych należy użyć.
2. W większości przypadków (być może z wyjątkiem wąskich korytarzy między przeszkodami, gdzie precyzja aproksymacji jest konieczna) niewypukłe przeszkody powinny być aproksymowane wypukłymi.

3.3. Uwagi obliczeniowe

Podczas automatycznego (symbolicznego) wyliczanie skomplikowanych całek należy zaplanować obliczenia właściwie, nie polegając jedynie na podstawianiu do wzorów. Części (wyrażenia) niezależne od zmiennej całkowania powinny zostać zastąpione przez wprowadzone symbole (parametry), a po obliczeniach całek – na powrót rozpakowane. W ten sposób uzyskuje się mniej skomplikowane wyrażenia wynikowe (częściowe), a czasem nawet możliwość uzyskania rezultatu. Zawsze wyliczenia są szybsze i efektywniejsze. Pakiety obliczeń symbolicznych oferują także możliwość podania dodatkowych warunków na pewne zmienne (np. ich dodatnia wartość lub większa niż pewna stała). Wykorzystując ten mechanizm unika się otrzymywania długich wariantowych wyrażeń, a czasem wręcz możliwość otrzymania jakiegokolwiek wyniku. Zaobserwowano także, że obliczanie całek oznaczonych lepiej wykonać dwuetapowo: najpierw przez wyliczanie całek nieoznaczonych, z następnym podstawieniem granic całkowania. W tabeli 1 zebrano czasy wyliczenia składowych $(f_x, f_y)^T$ siły odpychającej wywieranej przez segment liniowy (jak na rys. 1a) na symbolicznie zadany punkt $(x_a, y_a)^T$ wyliczonych zgodnie ze wzorem (1) i poddanych systemowej funkcji upraszczającej wyrażenia.

4. Symulacje

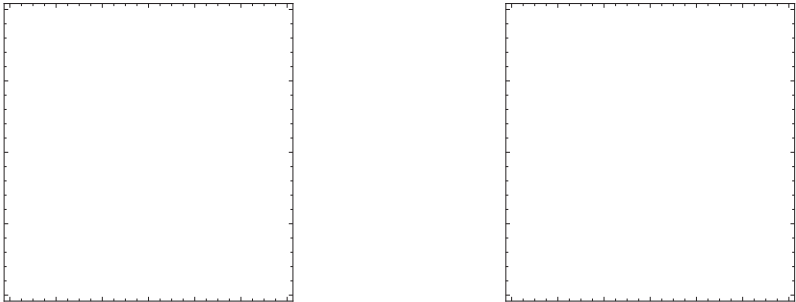
Pierwsza symulacja ilustruje siły wywierane na punktowego robota przez segment liniowy $[-a, a]$ umieszczony na osi x symetrycznie względem osi y , dla $a = 1$, oraz ładunku rozlokowanego jednorodnie na odcinku lub skumulowanego w trzech punktach: $(-1, 0)^T$, $(0, 0)^T$, $(1, 0)^T$. Na rys. 2 przedstawiono warstwicę składowych sił f_x , f_y (ze względu na symetrię względem osi y nie pokazano wartości dla $x < 0$; gdzie składowa f_x ma znak przeciwny). Dyskretny rozkład ładunków powoduje lokalne minima w liczbie o jeden mniejszej od liczby ładunków i głębokości zależnej od ich

Tab. 1. Czas obliczeń, $[s]$, składowych siły odpychającej od segmentu liniowego w zależności od parametru r oraz sposobu wyliczenia całek (oznaczona lub nieoznaczona, a następnie wartościowana w granicach całkowania). Dodatkowe (c) oznacza, że uzyskano wyrażenie zawierające liczby zespolone

r	całka oznaczona	całka nieoznaczona
3/2	3.0	0.05
2	6.5	0.35
5/2	12.4(c)	0.09
3	11.9	0.41
7/2	5.3(c)	0.21

a

b



Rys. 2. Wybrane warstwie składowych siły $\boldsymbol{f} = (f_x, f_y)^T$ w obszarze $[0, 1.2] \times [0.1, 0.5]$. Siła wywierana przez: a,b – segment liniowy; c,d – trzy punkty

liczby (czym większa, tym minima płytsze). Dla rozkładu ciągłego ładunku lokalne minima nie występują.

Druga symulacja ilustruje aproksymację okręgu (scentrowanego w $(0,0)$ i promieniu $R = 1$) N-kątem foremnym. Segment zerowy jest prostopadły do osi x , a kolejne są numerowane przeciwnie do ruchu wskazówek zegara. Na rys. 3 przedstawiono procent całkowitej siły pochodzący od kolejnych segmentów w zakresie odległości $d \in [0.05, 1]$ od brzegu N-kąta. Taka orientacje N-kąta względem robota ułatwia wizualizację pojedynczej siły (bowiem $f_y = 0$). Jak oczekiwano, czym bliżej brzegu

Rys. 3. Udział procentowy siły $f_y = \|f\|$ pochodzący od segmentów N -kąta foremnego, która jest wywierana na punkt odległy o d od brzegu

przeszkody, tym udział zerowego segmentu bardziej dominuje wypadkową siłę.

Udział pozostałych segmentów zwiększa się wraz ze wzrostem d , jednak amplituda siły maleje istotnie (400 razy, z $d = 0.05$ do $d = 1$). Udział pozostałych segmentów N wzrasta w relacji do dominującego segmentu, bowiem jest on krótszy, a pozostałe bardziej liczne. Jak łatwo zauważyć warto modelować brzeg przeszkody tylko paroma segmentami położonymi najbliżej robota.

5. Podsumowanie

W referacie przedstawiono analityczne (symboliczne) formuły dla wyliczenia wypadkowej siły wywieranej przez obiekty elementarne na punktowego robota. Zaproponowane podejście polega na transformowaniu zadania do przypadku kanonicznego w specjalnie dobranym układzie lokalnym, gdzie może być efektywnie (analitycznie) rozwiązane. Wszystkie obliczenia symboliczne są przeprowadzane w trybie off-line, a gotowe wzory mogą być implementowane w czasie rzeczywistym. Podkreślono właściwy dobór reprezentacji odcinków, a także możliwość ich wykorzystania do aproksymacji trudnych do wyliczenia kształtów obiektów. Rozszerzenie na przypadek trójwymiarowy, jakkolwiek możliwe, wiąże się z istotnym wzrostem skomplikowania obliczeń (całki podwójne, po obszarach niekoniecznie łatwo parametryzowalnych).

Literatura

- [1] C. Altafini. The General n -Trailer Problem: Conversion into Chained Form, Conf. on Decision and Control, Tampa, 1998, s. 3129–3130.
- [2] A. Gasparetto et al. Path Planning and Trajectory Planning Algorithms: a General Overview. In: Motion and Operation Planning of Robotic Systems, chap. 1, Springer, 2015, s. 3–25.
- [3] S. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [4] R.M. Murray, Z. Li, S.S. Sastry. *A mathematical introduction to robot manipulation*, CRC Press, 1994.
- [5] Y.S. Nam, B.H. Lee, N.Y. Ko. An analytic Approach to Moving Obstacles Avoidance Using an Artificial Potential Field. IEEE/RSJ Int. Conf. on Intelligent Ro-

- bots and Systems. Human Robot Interaction and Cooperative Robots, 2, 1995, s. 482–487.
- [6] J. Ren et al. A Potential Field Model Using Generalized Sigmoid Functions. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 37(2), 2007, s. 477–484. doi: 10.1109/TSMCB.2006.883866
 - [7] T. Rybus. The obstacle vector field (OVF) method for collision-free trajectory planning of free-floating space manipulator. Bull. Polish Acad. Sci. Tech., 70(2), e140691, 2022, doi 10.24425/bpasts.2022.140691.
 - [8] S. Panati, B. Bayanjargal, K.T. Chong. Autonomous Mobile Robot Navigation Using Harmonic Potential Fields. IOP Conf. Ser.: Mater. Sci. Eng. 83, 2015, doi 10.1088/1757-899X/83/1/012018
 - [9] M. Spong, M. Vidyasagar. *Robot dynamics and control*. Wiley, 2008.
 - [10] J.E. Ratajczak, K. Tchoń. Normal forms and singularities of non-holonomic robotic systems: A study of free-floating space robots. Systems & Control Letters. 2020, vol. 138, art. 104661, s. 1–9.
 - [11] Wolfram Research, Inc., Mathematica Version 8.0, Champain IL, 2010.

An analytic support of the artificial potential field method in motion planning tasks

In this paper analytic expressions are provided to calculate resulting repelling net-forces acting on a point robot due to uniformly charged elementary (mainly linear) objects. The net force is indispensable in many popular motion planning algorithms based on an artificial potential field method. It was shown how to speed-up computations via transforming the task into appropriately selected local frame associated with an object. Some issues were illustrated in simulations.

Control of Multi-Agent Systems with APF-Based Collision Avoidance - Analysis of Deadlocks Between Two Robots*

Wojciech Kowalczyk¹, Arpit Joon¹

Abstract

An analysis is carried out in this paper of a deadlock scenario between two robots in multi-agent systems. The approach uses a specially constructed Lyapunov function, which also serves as a potential function for the system. By differentiating this function with respect to the state variables, the agents' velocities can be determined. This method enables the analysis of the system's behavior using the eigenvalues of the Hessian matrix, which, for the two-robot case, have been derived explicitly. The theoretical results are illustrated through numerical simulations.

1. Introduction

In recent years, the deployment of multiple agents to collaboratively complete tasks has become increasingly common, particularly in environments such as warehouses where mobile robots operate in teams. However, coordinating multiple autonomous mobile robots introduces several challenges, one of which is the occurrence of deadlock. To address deadlocks, a thorough analysis can help identify preventive measures or develop new control strategies for teams of robots to resolve deadlock scenarios.

Various researchers have proposed different approaches to handle deadlock situations. For instance, the study in [1] presents a method for natural deadlock resolution by extending Gauss's Principle of Least Constraint (GPLC). In [2], a tiling-based movement constraint system is introduced to prevent deadlocks in Multi-Agent Pathfinding and Decision-making (MAPD) systems. A hierarchical control framework that dynamically reconfigures the network topology to alleviate deadlocks in large-scale robotic systems is discussed in [3]. Additionally, [4] provides a comparative analysis of various deadlock prevention algorithms.

The paper [18] presents an analytical study of deadlocks in multi-robot systems controlled via Control Barrier Function Quadratic Programs (CBF-QPs), characterizes the conditions leading to deadlock, and proposes a decentralized algorithm for provable deadlock resolution. The authors of paper [5] formally characterize deadlock using optimization duality and graph-theoretic analysis, demonstrating that deadlocks are tied to force equilibrium conditions among robots and occur near the boundaries of the safety set. They also show that the complexity of deadlock scenarios grows exponentially with the number of robots. Finally, they propose a provably correct

*This work was supported by statutory grant 0211/SBAD/0124.

¹Poznan University of Technology, Piotrowo 3A, Poznan, Poland wojciech.kowalczyk@put.poznan.pl, arpit.joon@put.poznan.pl

decentralized algorithm that guarantees deadlock resolution, verified through simulations and experiments with Khepera-IV robots.

Another research [6] proposes a reactive mission and motion planning framework designed for small teams of robots operating in dynamic environments with moving obstacles such as humans. Their method synthesizes a correct-by-construction mission plan from linear temporal logic specifications, ensuring collision avoidance, deadlock resolution, and fulfillment of complex tasks. A key contribution is the integration of offline high-level mission synthesis with online distributed motion planning, allowing robots to react dynamically to obstacles without global environment knowledge. The approach was validated through both physical experiments with humanoid robots and simulations with aerial vehicles.

A high-level mission planning is presented in paper [7], specified using linear temporal logic (LTL), with real-time local motion planning to ensure collision avoidance and deadlock resolution. The method excels in scalability by relying solely on local sensing data, eliminating the need for global environmental knowledge. The research further improves practicality by automatically generating environment assumptions, presented as user-friendly certificates, that guarantee task feasibility when initial specifications are unrealizable. Validated through simulations and experiments with ground and aerial robots, this work offers a theoretically sound and practical solution for multi-robot coordination in unpredictable settings.

A separate group of studies focuses on resolving deadlocks for teams of mobile robots using discrete-event systems. Methods of this kind [20] are effective even in complex scenarios involving multiple robots. However, the authors do not provide formal proof of stability for the full hybrid structure. In [21], a similar approach is applied to deadlock avoidance, particularly in situations where tasks share common resources. While deadlock-freeness and safe resource allocation are demonstrated, no stability analysis is presented. In [22], the authors proposed a solution to the problem of efficient robot coordination in large logistics facilities (e.g., warehouses, and crop tunnels), where collision avoidance and deadlock prevention are essential. A topological representation of the environment was used, which simplifies the model but often leads to bottlenecks that slow down robot flow. This work also does not provide a proof of stability.

The authors have previously published numerous works on multi-robot systems employing Artificial Potential Functions (APF). This approach offers several advantages, particularly the gradual increase of the potential function, which results in a gentle response to obstacles when distant and a stronger reaction as the robot approaches the obstacle's boundary. However, it is well known that combining set-point control or trajectory tracking with APF can lead to the emergence of points or regions distant from the goals where agents become stuck, a phenomenon known as deadlock.

The objective of this study was to investigate a scenario in which two robots move in opposite directions. Each of them is surrounded by a circular APF. In such a situation, a deadlock occurs. The authors aimed to determine whether the system in this case resides in a stable equilibrium (local minimum) or an unstable one (saddle point).

Section 2 presents the control algorithm, including the system model, collision avoidance, and the controller. The stability analysis based on the Lyapunov function of the closed-loop system is described in Section 3. Section 4 depicts the main nov-

elty of this article, a deadlock analysis using eigenvalues of the Hessian. Section 5 includes a brief discussion of deadlock analysis for $N \geq 3$. Section 6 shows a numerical simulation of two robots in a deadlocked situation. Section 7 contains concluding remarks.

2. Control

This section presents a model of the system, artificial potential functions (APFs) used to avoid collisions, and a set-point control algorithm for a team of mobile robots.

2.1. Model of the system

Consider a team of autonomous mobile robots consisting of N agents operating in an environment. Each robot is modeled using single-integrator dynamics, defined as:

$$\dot{h}_i = u_i, \quad (1)$$

where $h_i = [x_i, y_i]^T \in \mathbb{R}^2$ is the position of i -th agent, $u_i = [u_{xi}, u_{yi}]^T \in \mathbb{R}^2$ is control input and $i = 1 \dots N$, N - number of robots. Each agent has initially assigned a target location $h_{di} = [x_{di}, y_{di}]^T \in \mathbb{R}^2$.

2.2. Collision avoidance

The collision avoidance approach presented in this paper is based on the concept introduced in [11], which was later widely adopted by others for multi-agent systems [12], [13], [14], [15], [16].

An agent i operating in the environment avoids collisions with other agents j , $\forall j \in \{1, \dots, N\} \setminus i$ using an *Artificial Potential Function* (APF). Each robot is surrounded by APFs that rise to infinity near the robot/obstacle boundary r (j - the number of robot/obstacle) and decrease to zero at some distance R , $R > r$.

One can introduce the following function [15]:

$$B_{aij}(l_{ij}) = \begin{cases} 0 & \text{for } l_{ij} < r \\ e^{\frac{l_{ij}-r}{l_{ij}-R}} & \text{for } r \leq l_{ij} < R \\ 0 & \text{for } l_{ij} \geq R \end{cases}, \quad (2)$$

that gives output $B_{aij}(l_{ij}) \in \langle 0, 1 \rangle$. The distance between the i -th robot and the j -th robot is defined as the Euclidean length $l_{ij} = \|[x_j \ y_j]^T - [x_i \ y_i]^T\|$.

Scaling the function given by Eq. (2) within the range $\langle 0, \infty \rangle$ can be given as follows:

$$V_{aij}(l_{ij}) = \frac{B_{aij}(l_{ij})}{1 - B_{aij}(l_{ij})}, \quad (3)$$

that is used later to avoid collisions. This function is discontinuous at $l_{ij} = r$; however, this point cannot be reached, as proven in [19].

Function (3) has the following properties on the interval $l_{ij} \in (r, R)$:

$$V_{aij} > 0, \quad \frac{\partial V_{aij}}{\partial l_{ij}} < 0, \quad \frac{\partial^2 V_{aij}}{\partial l_{ij}^2} > 0. \quad (4)$$

The terms "collision area" or "collision region" refer to positions where the condition $l_{ij} < r$ holds, while the "collision avoidance area" or "collision avoidance region" is defined as the area satisfying $r < l_{ij} < R$.

Assumption 1. *The reference positions satisfy the condition $\forall \{i, j\}, i \neq j, \|[x_{di} - x_{dj} \ y_{di} - y_{dj}]\| > R$.*

Assumption 1 ensures that the APFs do not generate a collision avoidance component of the control if all the robots reach the desired positions.

2.3. Controller

$$\begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix} = \begin{bmatrix} k_p e_{xi} \\ k_p e_{yi} \end{bmatrix} - \sum_{j \in \{1, \dots, N\} \setminus i} \begin{bmatrix} \frac{\partial V_{aij}}{\partial x_i} & \frac{\partial V_{aij}}{\partial y_i} \end{bmatrix}^T \quad (5)$$

where

$$\begin{aligned} e_{xi} &= x_{di} - x_i \\ e_{yi} &= y_{di} - y_i \end{aligned} \quad (6)$$

and vector $[x_{di} \ y_{di}]^T$ is desired position of the i -th robot, and k_p is positive constant.

As a consequence, each pair of APFs associated with robots i and j meets the following conditions:

$$\begin{bmatrix} \frac{\partial V_{aij}}{\partial x_i} \\ \frac{\partial V_{aij}}{\partial y_i} \end{bmatrix}^T = - \begin{bmatrix} \frac{\partial V_{aij}}{\partial x_j} \\ \frac{\partial V_{aij}}{\partial y_j} \end{bmatrix}^T = - \begin{bmatrix} \frac{\partial V_{aji}}{\partial x_j} \\ \frac{\partial V_{aji}}{\partial y_j} \end{bmatrix}^T = \begin{bmatrix} \frac{\partial V_{aji}}{\partial x_i} \\ \frac{\partial V_{aji}}{\partial y_i} \end{bmatrix}^T. \quad (7)$$

3. Stability analysis

The Lyapunov function candidate for the system of N mobile robots is as follows:

$$V = \sum_{i=1}^N \left[k_p \frac{1}{2} (e_{xi}^2 + e_{yi}^2) + \sum_{j=1, j \neq i}^N \frac{1}{2} V_{aij} \right]. \quad (8)$$

The function (8) satisfies the following conditions:

1. $V(0) = 0$
 2. $V(e) > 0$ for $e \neq 0$
- where $e = [e_{x1} \ \dots \ e_{yN}]^T$.

In the next steps it will be shown that $\dot{V} \leq 0$ and the close-loop system is stable.

The time derivative of the Lyapunov function (8) is given by the following equation:

$$\dot{V} = \sum_{i=1}^N \left[k_p (e_{xi} \dot{e}_{xi} + e_{yi} \dot{e}_{yi}) + \sum_{j=1, j \neq i}^N \left(\frac{\partial V_{aij}}{\partial x_i} \dot{x}_i + \frac{\partial V_{aij}}{\partial y_i} \dot{y}_i \right) \right]. \quad (9)$$

In the next steps it is assumed that desired positions for robots $[x_{di} \ y_{di}]^T$ are constant.

Differentiating (6) with respect to time and considering set-point control ($\dot{x}_{di} = 0$ and $\dot{y}_{di} = 0$) one can write:

$$\begin{aligned}\dot{e}_{xi} &= -\dot{x}_i \\ \dot{e}_{yi} &= -\dot{y}_i\end{aligned}\quad (10)$$

Using above Eq. the formula (9) can be rewritten as follows:

$$\dot{V} = \sum_{i=1}^N \left[k_p e_{xi}(-\dot{x}_i) + k_p e_{yi}(-\dot{y}_i) + \sum_{j=1, j \neq i}^N \left(\frac{\partial V_{aij}}{\partial x_i} \dot{x}_i + \frac{\partial V_{aij}}{\partial y_i} \dot{y}_i \right) \right]. \quad (11)$$

After simple transformations Eq. (11) can be transformed as follows:

$$\dot{V} = \sum_{i=1}^N \left[(-\dot{x}_i) \left(k_p e_{xi} - \sum_{j=1, j \neq i}^N \frac{\partial V_{aij}}{\partial x_i} \right) + (-\dot{y}_i) \left(k_p e_{yi} - \sum_{j=1, j \neq i}^N \frac{\partial V_{aij}}{\partial y_i} \right) \right]. \quad (12)$$

Substituting Eq. (5) into (1) and then the result into (12) one can write:

$$\dot{V} = \sum_{i=1}^N \left[- \left(k_p e_{xi} - \sum_{j=1, j \neq i}^N \frac{\partial V_{aij}}{\partial x_i} \right)^2 - \left(k_p e_{yi} - \sum_{j=1, j \neq i}^N \frac{\partial V_{aij}}{\partial y_i} \right)^2 \right]. \quad (13)$$

As the right-hand side of the formula (13) is always non-positive the condition $\dot{V} \leq 0$ is always fulfilled and the closed-loop system is stable in the sense of Lyapunov.

The convergence of the system occurs under strict inequality $\dot{V} < 0$, that is, for all cases except for the states where the following equalities hold:

$$\left| k_p e_{xi} - \sum_{j=1, j \neq i}^N \frac{\partial V_{aij}}{\partial x_i} \right| = 0, \quad \left| k_p e_{yi} - \sum_{j=1, j \neq i}^N \frac{\partial V_{aij}}{\partial y_i} \right| = 0. \quad (14)$$

provided that the collision avoidance component is nonzero.

Authors of the paper [19] shown that collision avoidance is guaranteed if $\dot{V}_{aij} \leq 0$ and $\lim_{\| [x_i \ y_i]^T - [x_j \ y_j]^T \| \rightarrow r^+} V_{aij} = +\infty$.

4. Two-robot deadlock

The conditions in Eq. (14) cover a wide range of cases, which may involve some or all of the robots in the group. In the following, we will consider the case of a team of two robots ($N = 2$) for which the condition (14) is satisfied (Fig. 1).

Definition 1. *The robot i is in a deadlock if $u_i = \mathbf{0}$ and $[e_{xi} \ e_{yi}]^T \neq \mathbf{0}$.*

The above definition is equivalent to the one proposed in [18].

Deadlock is a situation for a robot where it still needs to converge to its target point but is trapped along its path due to external forces, such as the APF generated by other robots. This scenario should be avoided so that the robot can reach its target point.

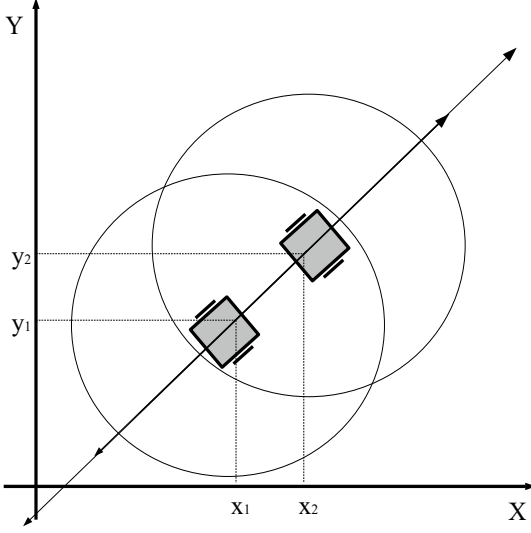


Fig. 1: Two robots in deadlock

The conditions given by Equation (14) can be written for two agents as follows:

$$k_p e_{x1} = \frac{\partial V_{a12}}{\partial x_1}, \quad k_p e_{y1} = \frac{\partial V_{a12}}{\partial y_1}, \quad k_p e_{x2} = \frac{\partial V_{a21}}{\partial x_2}, \quad k_p e_{y2} = \frac{\partial V_{a21}}{\partial y_2}. \quad (15)$$

As the partial derivatives fulfill conditions $\frac{\partial V_{a12}}{\partial x_1} = -\frac{\partial V_{a21}}{\partial x_2}$ and $\frac{\partial V_{a12}}{\partial y_1} = -\frac{\partial V_{a21}}{\partial y_2}$ (refer to Eq. (7)) the system will not converge to the global minimum of V ($[e_{x1} \ e_{y1}]^T = [e_{x2} \ e_{y2}]^T = \mathbf{0}$). In this scenario system remains in state where $[e_{x1} \ e_{y1}]^T = -[e_{x2} \ e_{y2}]^T$.

The Lyapunov function (Eq. 8) has a special structure because it also serves as a potential function for the system (1). Taking its gradient with respect to the state variables with a negative sign, one obtains (1) after substituting the control law (5). This particular property allows for a detailed analysis of the system's behavior at critical points using the Hessian:

$$H(V)_{(N=2)} = \begin{bmatrix} k_p + \frac{\partial^2 V_{a12}}{\partial x_1^2} & \frac{\partial^2 V_{a12}}{\partial x_1 \partial y_1} & \frac{\partial^2 V_{a12}}{\partial x_1 \partial x_2} & \frac{\partial^2 V_{a12}}{\partial x_1 \partial y_2} \\ \frac{\partial^2 V_{a21}}{\partial y_1 \partial x_1} & k_p + \frac{\partial^2 V_{a12}}{\partial y_1^2} & \frac{\partial^2 V_{a12}}{\partial y_1 \partial x_2} & \frac{\partial^2 V_{a12}}{\partial y_1 \partial y_2} \\ \frac{\partial^2 V_{a21}}{\partial x_2 \partial x_1} & \frac{\partial^2 V_{a21}}{\partial x_2 \partial y_1} & k_p + \frac{\partial^2 V_{a21}}{\partial x_2^2} & \frac{\partial^2 V_{a21}}{\partial x_2 \partial y_2} \\ \frac{\partial^2 V_{a21}}{\partial y_2 \partial x_1} & \frac{\partial^2 V_{a21}}{\partial y_2 \partial y_1} & \frac{\partial^2 V_{a21}}{\partial y_2 \partial x_2} & k_p + \frac{\partial^2 V_{a21}}{\partial y_2^2} \end{bmatrix}. \quad (16)$$

For the above matrix, it is possible to determine the eigenvalues in an explicit form. They are as follows:

$$e_1 = e_2 = k_p, \quad e_3 = k_p + 2 \frac{\partial^2 V_{a12}}{\partial l_{12}^2}, \quad e_4 = k_p + \frac{2}{l_{12}} \frac{\partial V_{a12}}{\partial l_{12}}. \quad (17)$$

The above result was obtained using Matlab with the Symbolic Toolbox, along with additional manual transformations.

We will then use the eigenvalues (17) to examine the system's behavior in the state described by Equation (15). The eigenvalues e_1 , e_2 , and e_3 are always positive ($\frac{\partial^2 V_{a12}}{\partial l_{12}^2} > 0$ for $l_{ij} \in (r, R)$), while e_4 will undergo further analysis. For the critical point (15), there will be a local minimum if $e_4 > 0$, and a saddle point if $e_4 < 0$. The first case would be problematic, as it would cause the robot to have a tendency to permanently get stuck away from the goal, while in the second case, the robot can easily be displaced from the equilibrium because it is in an unstable state.

In the following steps, we will check whether the following condition is met:

$$e_4 = k_p + \frac{2}{l_{12}} \frac{\partial V_{a12}}{\partial l_{12}} < 0. \quad (18)$$

We will use the property (see Section A in the appendix):

$$\sqrt{\left(\frac{\partial V_{aij}}{\partial x_i}\right)^2 + \left(\frac{\partial V_{aij}}{\partial y_i}\right)^2} = \left|\frac{\partial V_{aij}}{\partial l_{ij}}\right|. \quad (19)$$

Taking into account that $\frac{\partial V_{aij}}{\partial l_{ij}} < 0$ for $l_{ij} \in (r, R)$ above equation can be rewritten as follows:

$$\sqrt{\left(\frac{\partial V_{aij}}{\partial x_i}\right)^2 + \left(\frac{\partial V_{aij}}{\partial y_i}\right)^2} = -\frac{\partial V_{aij}}{\partial l_{ij}}. \quad (20)$$

Using Eq. (20) and the first two equalities in (15), we will carry out the analysis for robot 1 (the procedure for robot 2 would be analogous):

$$\sqrt{(k_p e_{x1})^2 + (k_p e_{y1})^2} = -\frac{\partial V_{a12}}{\partial l_{12}}, \quad (21)$$

that can be easily transformed into:

$$\frac{\partial V_{a12}}{\partial l_{12}} = -k_p \sqrt{e_{x1}^2 + e_{y1}^2}. \quad (22)$$

Substituting the above equation into (18) one obtains:

$$k_p - \frac{2k_p}{l_{12}} \sqrt{e_{x1}^2 + e_{y1}^2} < 0 \quad (23)$$

which is satisfied if the following inequality holds:

$$\sqrt{e_{x1}^2 + e_{y1}^2} > \frac{l_{12}}{2}. \quad (24)$$

As mentioned earlier, an analogous procedure can easily be carried out for the robot 2. In a deadlock situation, $l_{12} < R$ (a control component related to collision avoidance must be present to balance the term responsible for moving toward the goal), while Assumption 1 ensures that the goals are separated by a distance greater than R . These properties, along with the symmetry of the behavior of both robots, guarantee that condition (24) is always satisfied and that the critical point is unstable (it is a saddle point).

5. Deadlocks for $N \geq 3$

The analysis method presented in the previous section can also be applied to systems of three or more robots. While computing the Hessian matrix itself is straightforward, determining its eigenvalues becomes increasingly challenging as the number of robots grows, due to the rapidly escalating computational complexity. Moreover, obtaining explicit expressions for the eigenvalues may not be feasible for $N > 2$, owing to the Abel–Ruffini theorem. This theorem states that general polynomial equations of degree five or higher cannot be solved using radicals, which implies that analytical solutions for the eigenvalues of higher-dimensional Hessian matrices (e.g., a 6 by 6 matrix for $N = 3$) may not exist. Consequently, it might be impossible to derive closed-form eigenvalue expressions for systems with more than two robots. The authors intend to explore this issue in future research. An alternative solution may be numerical analysis. However, it does not possess the generalized result characteristics found in analytical solutions, such as the one presented in 4, which allows for the simplification of the deadlock resolution system to a simple disturbance generator within the system. In cases where the eigenvalues cannot be computed in closed form, an attempt will be made to conduct the widest possible numerical analysis.

6. Simulation results

The efficacy of the proposed approach is evaluated on a team of two mobile robots deliberately arranged to experience deadlock during their motion. Each robot is a two-wheeled, differentially driven platform subject to feedback linearization, which reduces its kinematic model to a single-integrator (see Eq. 1) using method presented in [17]. Initially, robot R1 is positioned at $(2, 0)$ m and robot R2 at $(-2, 0)$ m, with their respective goals located at $(-2, 0)$ m and $(2, 0)$ m. The control parameters are set to the following values $k = 0.2$, $r = 0.4$ m, and $R = 0.6$ m.

Figure 2a depicts the trajectories of both robots in the $x - y$ plane. The robots advance toward each other along the x -axis until they enter a deadlock. At $t = 5$ s, a small perturbation of R2's target position - an offset of $(0, 0.001)$ m in the y direction is applied for the period of 1 s to push the system from unstable equilibrium. Following this disturbance, the robots successfully circumvent one another without collision and converge to their reference positions in 40 s.

Figures 2b and 2c illustrate the evolution of position errors along x and y axes, respectively. Due to the small magnitude of the applied perturbation, its corrective influence becomes apparent at approximately 3 s (in the eighth second of the simulation). The controller reduces the positional errors of the robots to near zero, as observed in Figures 2b and 2c, thereby validating its effectiveness. Figures 2d and 2e

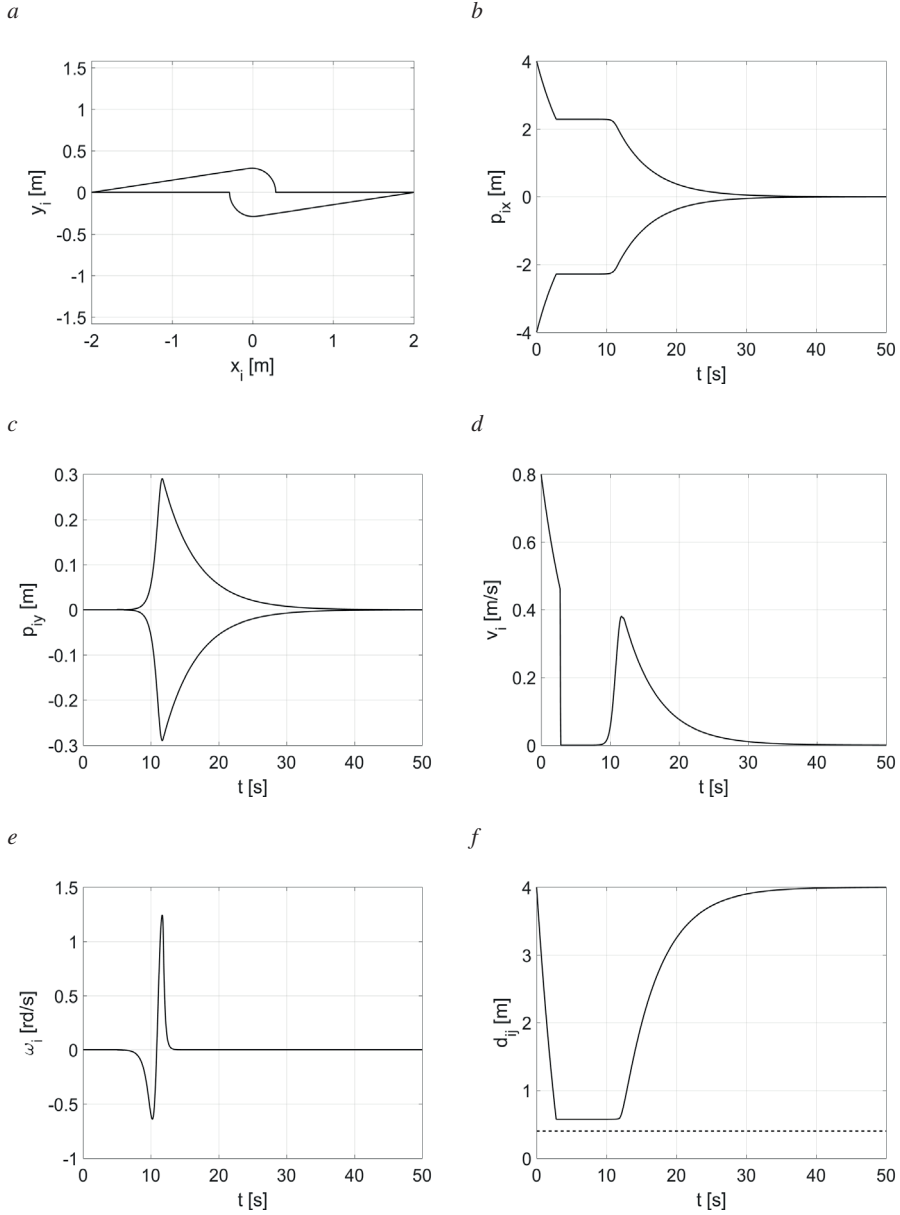


Fig. 2: Simulation results for $N = 2$ robots, a - Locations of robots in XY-plane, b - Position errors in x axis, c - Position errors in y axis, d - Linear velocity of the robots, e - Angular velocity of the robots, f - Distance between the robots

present the time graphs of the linear and angular velocities of robots, respectively. The robot's linear velocity remains close to zero between 2 and 8 seconds, corresponding to the period when it was trapped in a deadlock situation. A sudden increase in linear velocity is observed as the robot begins to escape the deadlock. In Figure 2e, the angular velocity is zero around 8 seconds, indicating that the robot is moving straight. Between 8 to 12 seconds, the robot's rotation is observed by angular velocity.

Fig. 2f shows the distance between robots throughout the trial; it remains strictly greater than r (dashed line), confirming that no collisions occurred.

The video presenting robot animation for the case discussed in this section is available at the following link: <https://wojciech.kowalczyk.pracownik.put.poznan.pl/research/two-robot-deadlock/two-robot-deadlock.html>

7. Conclusion

This study presents an analysis of the deadlock that involves two robots modeled as first-order integrators operating in a planar task space. By employing a specially constructed Lyapunov function, which also serves as a potential function for the system, the behavior of the agents at critical points is examined using the eigenvalues of the Hessian matrix, explicitly derived for this two-robot team. The analysis reveals that, under the assumption regarding the positioning of reference points, the occurrence of deadlock does not correspond to a local minimum but rather to a saddle point, indicating an unstable equilibrium. This characteristic is significant because it facilitates the resolution of deadlock situations. The analytical findings are illustrated through numerical simulations. Future work will focus on extending this analysis to deadlock scenarios that involve a larger number of robots.

A. Relationships between derivatives

The following relations will be used:

$$\frac{\partial V_{aij}}{\partial x_i} = \frac{\partial V_{aij}}{\partial l_{ij}} \frac{\partial l_{ij}}{\partial x_i}, \quad \frac{\partial V_{aij}}{\partial y_i} = \frac{\partial V_{aij}}{\partial l_{ij}} \frac{\partial l_{ij}}{\partial y_i}. \quad (25)$$

The derivatives of l_{ij} with respect to x_i and y_i are as follows:

$$\frac{\partial l_{ij}}{\partial x_i} = \frac{x_i - x_j}{l_{ij}}, \quad \frac{\partial l_{ij}}{\partial y_i} = \frac{y_i - y_j}{l_{ij}}. \quad (26)$$

Substituting (26) into (25) one obtains:

$$\frac{\partial V_{aij}}{\partial x_i} = \frac{\partial V_{aij}}{\partial l_{ij}} \cdot \frac{x_i - x_j}{l_{ij}}, \quad \frac{\partial V_{aij}}{\partial y_i} = \frac{\partial V_{aij}}{\partial l_{ij}} \cdot \frac{y_i - y_j}{l_{ij}}. \quad (27)$$

The above equations are squared and added side by side:

$$\left(\frac{\partial V_{aij}}{\partial x_i} \right)^2 + \left(\frac{\partial V_{aij}}{\partial y_i} \right)^2 = \left(\frac{\partial V_{aij}}{\partial l_{ij}} \right)^2 \left(\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{l_{ij}^2} \right). \quad (28)$$

As the following equality holds true:

$$\left(\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{l_{ij}^2} \right) = 1. \quad (29)$$

Equation (28) can be transformed as follows:

$$\sqrt{\left(\frac{\partial V_{aij}}{\partial x_i} \right)^2 + \left(\frac{\partial V_{aij}}{\partial y_i} \right)^2} = \left| \frac{\partial V_{aij}}{\partial l_{ij}} \right|. \quad (30)$$

References

- [1] B. Zhang and H. P. Gavin, "Natural Deadlock Resolution for Multi-agent Multi-Swarm Navigation," 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 2021, pp. 5958–5963, doi: 10.1109/CDC45484.2021.9683102
- [2] B. Flammini, D. Azzalini, and F. Amigoni, "Preventing Deadlocks for Multi-Agent Pickup and Delivery in Dynamic Environments." In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 580–588.
- [3] K. Garg, S. Hamilton, and C. Fan, "Deadlock resolution of connected multi-agent systems using hierarchical control," in Proceedings of the 63rd IEEE Conference on Decision and Control (CDC), Milan, Italy, Dec. 2024, pp. 1275–1282. doi: 10.1109/CDC56724.2024.10886421.
- [4] M. Sauer, A. Dachsberger, L. Gighuber and L. Zalewski, "Decentralized Deadlock Prevention for Self-Organizing Industrial Mobile Robot Fleets," 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain, 2022, pp. 1–6, doi: 10.1109/COINS54846.2022.9854958.
- [5] Grover, J. S., Liu, C., and Sycara, K. (2020, June). Deadlock analysis and resolution for multi-robot systems. In International Workshop on the Algorithmic Foundations of Robotics, pp. 294–312. Cham: Springer International Publishing, 2020.
- [6] Alonso-Mora, J., DeCastro, J.A., Raman, V., Rus, D. and Kress-Gazit, H., 2018. Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles. *Autonomous Robots*, 42, pp.801–824.
- [7] Kittisares, S., Kobayashi, Y., Kumagai, T., Yasuda, S. and Yoshida, H., 2024, November. Multi-Robot Scheduling for Deadlock Avoidance Using Nonstop Areas. In IECON 2024–50th Annual Conference of the IEEE Industrial Electronics Society (pp. 1–7). IEEE.
- [8] Panagou, D.; Turpin, M.; Kumar, V. Decentralized goal assignment and trajectory generation in multi-robot networks: A multiple Lyapunov functions approach. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–5 June 2014; pp. 6757–6762, doi: 10.1109/ICRA.2014.6907857
- [9] W. Kowalczyk, K. Kozłowski, Self-rearrangement in a group of differentially-driven mobile robots changing shape of the formation, 2021 25th International Conference on Methods and Models in Automation and

- Robotics (MMAR), Miedzyzdroje, Poland, 2021, pp. 53–58, doi: 10.1109/MMAR49549.2021.9528466.
- [10] W. Kowalczyk, Formation Control and Distributed Goal Assignment for Multi-Agent Non-Holonomic Systems, *Applied Sciences* 9 (7), 1311, 2019.
- [11] Khatib, O. Real-time Obstacle Avoidance for Manipulators and Mobile Robots, *Int. J. Robot. Res.*, 1986, 5, 90–98.
- [12] Mastellone, S.; Stipanovic, D.; Spong, M. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *Int. J. Robot. Res.*, 2008, 107–126, doi:10.1177/0278364907084441.
- [13] Do, K.D. Formation Tracking Control of Unicycle-Type Mobile Robots With Limited Sensing Ranges, *IEEE Trans. Control Syst. Technol.* 2008, 16, 527–538, doi:10.1109/TCST.2007.908214
- [14] Kowalczyk, W.; Kozłowski, K.; Tar, J.K. Trajectory Tracking for Multiple Unicycles in the Environment with Obstacles. In *Proceedings of the 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, Budapest, Hungary, 23–27 June 2010; pp. 451–456, doi:10.1109/RAAD.2010.5524544
- [15] Kowalczyk, W.; Michałek, M.; Kozłowski, K. Trajectory Tracking Control with Obstacle Avoidance Capability for Unicycle-like Mobile Robot. *Bull. Pol. Acad. Sci. Tech. Sci.* 2012, 60, 537–546.
- [16] Kowalczyk, W.; Kozłowski, K. Leader-Follower Control and Collision Avoidance for the Formation of Differentially-Driven Mobile Robots. In *Proceedings of the MMAR 2018—23rd International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, 27–30 August 2018.
- [17] Kowalczyk, W., Kozłowski, K. (2014). Feedback Linearization of the Differentially Driven Mobile Robot: An Experimental Verification. In: Fodor, J., Fullér, R. (eds) *Advances in Soft Computing, Intelligent Robotics and Control. Topics in Intelligent Engineering and Informatics*, vol 8. Springer, Cham.
- [18] Grover, J., Liu, C., Sycara, K., The before, during, and after of multi-robot deadlock, *The International Journal of Robotics Research*, 42(6), 2023, pp. 317–336.
- [19] D. M. Stipanovic, P. F. Hokayem, M. W. Spong, and D. D. Ćiljak, 2007. Cooperative Avoidance Control for Multiagent Systems, *Journal of Dynamic Systems, Measurement and Control* 129(5): 699–707. doi:10.1115/1.2764510
- [20] Zhou, Y., Hu, H., Deng, G., Cheng, K., Lin, S., Liu, Y., Ding, Z., Distributed Motion Control for Multiple Mobile Robots Using Discrete-Event Systems and Model Predictive Control, in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 2, pp. 997–1010, Feb. 2024, doi: 10.1109/TSMC.2023.3322154
- [21] Ballal, P.M., Trivedi, A.C., Giordano, V. et al. Deadlock-free dynamic resource assignment in multi-robot systems with multiple missions: application in wireless sensor networks. *J. Control Theory Appl.* 8, 12–19 (2010). <https://doi.org/10.1007/s11768-010-9193-9>
- [22] Zhu, Z., Das, G., Hanheide, M., 2023. Autonomous Topological Optimisation for Multi-robot Systems in Logistics. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23)*. Association for Computing Machinery, New York, NY, USA, 791–799. <https://doi.org/10.1145/3555776.3577666>

Kubiczna interpolacja orientacji w przestrzeni rzutowej

Jacek Grzegorzewski¹

Streszczenie

W referacie przedstawiono rozszerzenie metody interpolacji funkcjami sklejanymi z przestrzeni liniowych do przestrzeni orientacji. W tym celu zastosowano algebrę kwaternionową i szczególne własności przestrzeni rzutowej $\mathbb{RP}^3$ dyfeomorficznej ze specjalną grupą ortogonalną $SO(3)$ opisującą orientację. Wyprowadzono odpowiednie formuły dla pojedynczego segmentu oraz dla interpolacji wielosegmentowej sklejanymi wielomianami kubicznymi z zachowaniem ciągłości przyspieszeń. Rozważania zilustrowano symulacyjnie na dwóch przykładach liczbowych dla zestawu wybranych parametrów swobodnych, które parametryzują rodzinę dopuszczalnych rozwiązań zadania interpolacyjnego.

1. Wstęp

Interpolacja w przestrzeni konfiguracyjnej lub zadaniowej jest częstym zadaniem robotycznym. Zadanie rozwiązuje się często za pomocą wielomianowych funkcji sklejanых. Podejście daje kontrolę nie tylko nad punktami osiąganymi przez krzywe interpolacyjne, ale też ich klasą ciągłości, zachowując przy tym prostotę aparatu matematycznego. Zachowanie odpowiedniej ciągłości jest szczególnie ważne z perspektywy układów mechanicznych, gdyż nieciągłości przyspieszenia mogą doprowadzić do uszkodzeń mechanicznych, bądź niemożliwych do realizacji trajektorii.

Interpolacja pozycji efektora jest równoważna z interpolacją w przestrzeni $\mathbb{R}^3$, natomiast interpolacja orientacji jest mniej eksplorowana i trudniejsza. Najczęściej spotykanym zadaniem jest interpolacja dwóch orientacji (zwykle łącząc je krzywą geodezyjną). W teorii podejście to umożliwia sklepanie dowolnej sekwencji orientacji, jednak ciągłość prędkości i przyspieszeń kątowych wygenerowanej trajektorii niekoniecznie będzie zachowana w węzłach interpolacyjnych. Ge i Ravani [3] zaproponowali algorytm zachowujący ciągłość uogólnionych krzywizn krzywej w specjalnej grupie euklidesowej, $SE(3)$, przez podzielenie $\mathbb{RP}^3$ na dwa orientowalne zbiory. Parametry krzywej obliczane są rekurencyjnie, a prędkość kątowa w pierwszym węźle może być wybrana dowolnie. Podobne idee wykorzystane były w pracy [11], by uzyskać G^2 gładkie (czyli zapewniające ciągłość wartości, pochodnej i krzywizny w węzłach) krzywe interpolacyjne z pomocą wielomianów 5-tego stopnia. Z kolei w pracy [7] opisywany jest algorytm wykorzystujący wykładnicze funkcje wielomianów kwadratowych w algebrze $\mathfrak{so}(3)$ (odpowiadającej grupie $SO(3)$ i mającej interpretację prędkości w tej grupie), który zachowuje ciągłości prędkości i przyspieszeń kątowych, oraz aproksymuje krzywe o minimalnym przyspieszeniu między zadanymi orientacjami. Niniejszy referat przedstawia podejście do interpolacji w $SO(3)$ wyko-

¹Katedra Cybernetyki i Robotyki, Politechnika Wrocławska,
ul. Janiszewskiego 11/17, 50-372 Wrocław, jacek.grzegorzewski@pwr.edu.pl.

rzystujące, podobnie jak [3] czy [11], sklejane wielomiany. W przeciwieństwie do [11] stopień wykorzystywanych wielomianów jest niższy, kosztem jednak utraty bezpośredniej kontroli nad prędkościami w węzłach. Kwaternionowa reprezentacja $SO(3)$ została wykorzystana by dla zadanej sekwencji orientacji wygenerować rodzinę funkcji bezpośrednio w $SO(3)$ oraz zachować ciągłość przyspieszeń kątowych w węzłach interpolacyjnych. Wyznaczanie współczynników funkcji sklejanych polega na rozwiązaniu blokowego trójdziagonalnego równania liniowego, nie wymagającego dużego nakładu obliczeniowego.

Referat zorganizowany jest następująco: w rozdziale 2 i 3 scharakteryzowano zadania interpolacji w przestrzeniach liniowych i w $SO(3)$, odpowiednio. W podrozdziale 3.1 przedstawiono niezbędny aparat matematyczny, który zastosowano do rozwiązania dwóch przykładowych problemów interpolacyjnych w $SO(3)$. W symulacyjnym rozdziale 4 zilustrowano zależność wygenerowanych trajektorii od swobodnych parametrów, przedstawiono pewne spostrzeżenia i wskazano możliwości rozwoju metody. W rozdziale 5 podsumowano zawartość referatu. W treści referatu przyjęto konwencję oznaczania macierzy, wektorów oraz kwaternionów czcionką pogrubioną.

2. Interpolacja w $T(3)$

Na wstępie przypomniana zostanie interpolacja wielomianowa w grupie translacji euklidesowych $T(3)$. Między $\mathbb{R}^n$ a $T(n)$ występuje izomorfizm, wyprowadzenia przedstawione będą więc w $\mathbb{R}^n$. Założmy, że danych jest $(k+1)$ ograniczeń punktowych, określających wartość funkcji interpolującej lub jej wybranych pochodnych w ustalonych punktach trajektorii (węzłach):

$$\mathbf{P}^{(n_0)}(t_0) = \mathbf{X}_0, \quad \dots \quad \mathbf{P}^{(n_k)}(t_k) = \mathbf{X}_k, \quad n_i \in \mathbb{N} \cup \{0\}, \quad t_0 \leq t_1 \leq \dots \leq t_{k-1} \leq t_k, \quad (1)$$

gdzie $\mathbf{P}^{(n)}(t)$ jest wartością n -tej pochodnej w chwili t , oraz naturalnie założono, że każda para (n, t) występuje w zbiorze ograniczeń tylko raz. Po wprowadzeniu oznaczenia $t^{\{N\}} := [1 \quad t \quad \dots \quad t^N]$, celem jest znalezienie wielomianu N -tego stopnia:

$$\mathbf{P}(t) = \sum_{i=0}^N \mathbf{P}_i t^i = \langle t^{\{N\}}, [\mathbf{P}_0 \quad \mathbf{P}_1 \quad \dots \quad \mathbf{P}_N]^T \rangle, \quad \mathbf{P}_i \in \mathbb{R}^n, \quad (2)$$

spełniającego zadane ograniczenia (1) ($\langle \cdot, \cdot \rangle$ oznacza iloczyn skalarny). Niech $T(n, t) := d^n t^{\{N\}} / dt^n$, nadużywając nieco notacji macierzowej, współczynniki $\mathbf{P}(t)$ można wyznaczyć rozwiązując równanie liniowe:

$$\begin{bmatrix} T(n_0, t_0) \\ \vdots \\ T(n_k, t_k) \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_N \end{bmatrix} = \begin{bmatrix} \mathbf{X}_0 \\ \vdots \\ \mathbf{X}_k \end{bmatrix}. \quad (3)$$

Jeśli $N = k$, rozwiązanie jest jednoznaczne. Dla $N > k$ układ jest niedookreślony, więc (jeśli rozwiązanie istnieje) współczynniki są liniowo zależne od $(N - k)$ zmiennych swobodnych, które mogą być dookreślone przez optymalizację pewnego kryterium

lub dobierane arbitralnie. Często spotykane są, tak zwane, ograniczenia Hermite’a, gdzie zakładane są tylko dwa węzły interpolacyjne ograniczające wartości i odpowiednio wysokie pochodne. W efekcie możliwe jest uzyskanie odpowiednio ciągłych funkcji interpolacyjnych między więzami.

Warto rozważyć również inną perspektywę na interpolację pochodzącą od de Casteljau, który przedstawił rekurencyjny algorytm na wyznaczanie wielomianów interpolacyjnych Beziea [9]. Algorytm de Casteljau nie pozwala bezpośrednio na nakładanie ograniczeń na pochodne, ma natomiast ciekawe właściwości geometryczne, i może być wyrażony w rekurencyjnej formie, którą łatwo przenieść do innych przestrzeni. Operując w przestrzeni euklidesowej, niech dany będzie zbiór punktów $\mathbf{X}_i \in \mathbb{R}^n, i = 0, \dots, k$ oraz $\mathbf{P}_{\mathbf{X}_i}(t) = \mathbf{X}_i$, wtedy

$$\mathbf{P}(t) = \mathbf{P}_{\mathbf{X}_0 \mathbf{X}_1 \dots \mathbf{X}_k}(t) = \mathbf{P}_{\mathbf{X}_0 \dots \mathbf{X}_{k-1}}(t) + t(\mathbf{P}_{\mathbf{X}_1 \dots \mathbf{X}_k}(t) - \mathbf{P}_{\mathbf{X}_0 \dots \mathbf{X}_{k-1}}(t)). \quad (4)$$

Wielomian ten przechodzi kolejno przez $\mathbf{X}_0$ i $\mathbf{X}_k$, oraz znajduje się wewnątrz otoczki wypukłej zbioru $\{\mathbf{X}_0, \dots, \mathbf{X}_k\}$. Problem staje się bardziej skomplikowany po przeniesieniu na inne rozmaitości. O subtelnościach zachodzących w tym ogólniejszym środowisku więcej dowiedzieć się można w artykule [6] oraz zawartej tamże bibliografii.

3. Interpolacja w $SO(3)$

W przeciwieństwie do przestrzeni liniowych, $SO(3)$ ma bardziej skomplikowaną strukturę. Najprostszym problemem interpolacyjnym w $SO(3)$ jest oczywiście interpolacja dwóch orientacji. Problem ten zazwyczaj jest rozwiązywany z pomocą funkcji wykładniczych, generując trajektorię – krzywą geodezyjną [10]:

$$\mathbf{P}(t) = \mathbf{R}_0 e^{t \ln(\mathbf{R}_0^{-1} \mathbf{R}_1)}, \quad \mathbf{R}_0, \mathbf{R}_1 \in SO(3), \quad t \in [0, 1]. \quad (5)$$

Stosując powyższe podejście można rozszerzyć algorytm de Casteljeu do $SO(3)$ [5]. Metoda jednak nie skutkuje łatwymi do zastosowania numerycznymi funkcjami, oraz nie pozwala na łatwe nakładanie ograniczeń na prędkości i przyspieszenia kątowe. By rozważyć takie właśnie problemy interpolacyjne wykorzystany będzie dyfeomorfizm między $SO(3)$ a $\mathbb{RP}^3$, który odpowiada bezpośrednio kwaternionowej reprezentacji elementów $SO(3)$. Kwaternion $\mathbf{q} \in \mathbb{H}$ ($\mathbb{H}$ pochodzi od nazwiska J.R. Hamiltona) składa się z części skalarnej q_0 , wektorowej $i\mathbf{q}_1 + j\mathbf{q}_2 + k\mathbf{q}_3$ i jest zdefiniowany następująco:

$$\mathbf{q} = q_0 + i\mathbf{q}_1 + j\mathbf{q}_2 + k\mathbf{q}_3, \quad q_i=0, \dots, 3 \in \mathbb{R}. \quad (6)$$

Dodawanie kwaternionów wykonywane jest jak na wektorach czterowymiarowych. Mnożenie wersorów bazowych i, j, k , jest rozdzielne, asocjatywne oraz nieprzemienne, spełniając przy tym następujące właściwości:

$$i^2 = j^2 = k^2 = -1, \quad ijk = -1. \quad (7)$$

Sprzężenie kwaternionowe jest definiowane analogicznie jak dla liczb zespolonych:

$$\bar{\mathbf{q}} = q_0 - (i\mathbf{q}_1 + j\mathbf{q}_2 + k\mathbf{q}_3). \quad (8)$$

Podobnie można zdefiniować (nieco niestandardowo) normę kwaternionową:

$$\bar{q}q = q_0^2 + q_1^2 + q_2^2 + q_3^2. \quad (9)$$

Częściej spotykana jest norma z pierwiastkiem, forma (9) prowadzi jednak do prostszych wzorów i będzie stosowana w niniejszym referacie. Element $\mathbb{RP}^3$ o współrzędnych jednorodnych $[q_0 : q_1 : q_2 : q_3]$, odpowiada rotacji reprezentowanej przez kwaternion (6). Rozważać będziemy zatem trajektorie wielomianowe w $\mathbb{RP}^3$, natomiast by wyznaczyć ograniczenia na prędkości skorzystamy z algebry kwaternionowej.

3.1. Interpolacja w $\mathbb{RP}^3$

Interpolacja w $\mathbb{RP}^3$ nie różni się znacząco od interpolacji w $\mathbb{R}^4$. W przeciwieństwie jednak do innych metod interpolacji $SO(3)$ pozwala na łatwe nakładanie szerszej klasy ograniczeń. Rozważmy problem znalezienia sześcienniej trajektorii $P(t)$ (w przestrzeni $\mathbb{H}[t]$ wielomianów kwaternionowych jednej zmiennej) takiej, że:

$$P(0) = \alpha_0 Q_0, \quad P(1) = \alpha_1 Q_1, \quad \omega(0) = \omega_0, \quad \omega(1) = \omega_1, \quad \alpha_0, \alpha_1 \in \mathbb{R} \setminus \{0\}.$$

W przestrzeni rzutowej wielokrotności jej elementów są tożsame, jednak różnym wartościom α_i odpowiadają inne trajektorie. Relacja między prędkością kątową w układzie przestrzeni, a pochodną jednostkowego kwaternionu jest następująca [1]:

$$\dot{q}(t) = \frac{1}{2} \omega(t) q(t). \quad (10)$$

Wielomianowe trajektorie nie są jednak reprezentowane przez jednostkowe kwaterniony, więc relacja między pochodnymi a prędkością kątową jest następująca:

$$\dot{q} = \left(\frac{1}{2} \omega + c\right) q, \quad c \in \mathbb{R}. \quad (11)$$

Dowód 3.1. Niech $r = qr_0\bar{q}/N$, gdzie $N = \bar{q}q$, oraz $q, r_0 \in \mathbb{H}$. Wtedy:

$$\dot{r} = \frac{(\dot{q}r_0\bar{q} + qr_0\dot{\bar{q}}) - N\dot{N}r}{N^2}.$$

Podstawiając $r_0 = \bar{q}rq/N$ mamy:

$$\dot{r} = \frac{(\dot{q}\bar{q}r + r\dot{q}\bar{q})N - N\dot{N}r}{N^2}.$$

Niech $\alpha = \dot{q}\bar{q} = \alpha_s + \alpha_v$, gdzie $\alpha_s = (\alpha + \bar{\alpha})/2$, $\alpha_v = (\alpha - \bar{\alpha})/2$ są częścią skalarną i wektorową α . Wtedy:

$$\dot{r} = \frac{(\alpha_v r - r\alpha_v)N + rN(2\alpha_s - \dot{N})}{N^2}.$$

Ponieważ $\dot{q}\bar{q} + \bar{q}\dot{q} = \dot{N} = \alpha + \bar{\alpha} = 2\alpha_s$ otrzymujemy:

$$\dot{r} = \frac{\alpha_v r - r\alpha_v}{N}.$$

Interpretując kwaterniony wektorowe jako wektory, można zapisać iloczyn wektorowy $\mathbf{a} \times \mathbf{b} = \frac{1}{2}(\mathbf{ab} - \mathbf{ba})$, zatem z równania $\dot{\mathbf{r}} = \boldsymbol{\omega} \times \mathbf{r}$ uzyskujemy $\boldsymbol{\omega} = 2\boldsymbol{\alpha}_v/N$. By otrzymać pożądaną relację (11) należy wyznaczyć $\dot{\mathbf{q}}$ z zależności:

$$\boldsymbol{\omega} = i\omega_x + j\omega_y + k\omega_z = \frac{\dot{\mathbf{q}}\bar{\mathbf{q}} - \mathbf{q}\dot{\bar{\mathbf{q}}}}{N}, \quad (12)$$

jeżeli $\mathbf{q}$ jest wielomianem o kwaternionowych współczynnikach, co zawsze zachodzi w niniejszym referacie, $\omega_x, \omega_y, \omega_z \in \mathbb{R}(t)$ są funkcjami wymiernymi zmiennej t . Równanie (12) jest osobiwe, więc nie można jednoznacznie wyznaczyć $\dot{\mathbf{q}}$, a jedynie z dokładnością do pewnej stałej c :

$$\dot{\mathbf{q}} = \left(\frac{1}{2}\boldsymbol{\omega} + c\right)\mathbf{q}, \quad c \in \mathbb{R}.$$

Na podstawie wzoru (12) łatwo wykazać, że przyspieszenie kątowe jest równe

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\varepsilon} = \frac{(\dot{\mathbf{q}}\bar{\mathbf{q}} - \mathbf{q}\dot{\bar{\mathbf{q}}}) - \dot{N}\boldsymbol{\omega}}{N}. \quad (13)$$

Podobnym argumentem można pokazać, że:

$$\ddot{\mathbf{q}} = \left(\frac{1}{2}\boldsymbol{\varepsilon} + c\boldsymbol{\omega} + d\right)\mathbf{q}, \quad c, d \in \mathbb{R}. \quad (14)$$

Możemy więc wyrazić wcześniejsze ograniczenia jako ograniczenia na pochodne:

$$\mathbf{P}(0) = \alpha_0 \mathbf{Q}_0, \quad \mathbf{P}(1) = \alpha_1 \mathbf{Q}_1, \quad \dot{\mathbf{P}}(0) = \left(\frac{1}{2}\boldsymbol{\omega}_0 + c_0\right)\alpha_0 \mathbf{Q}_0, \quad \dot{\mathbf{P}}(1) = \left(\frac{1}{2}\boldsymbol{\omega}_1 + c_1\right)\alpha_1 \mathbf{Q}_1,$$

lub, alternatywnie, w postaci macierzowej:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \mathbf{Q}_0 \\ \alpha_1 \mathbf{Q}_1 \\ \frac{\alpha_0}{2} \boldsymbol{\omega}_0 \mathbf{Q}_0 + c_0 \alpha_0 \mathbf{Q}_0 \\ \frac{\alpha_1}{2} \boldsymbol{\omega}_1 \mathbf{Q}_1 + c_1 \alpha_1 \mathbf{Q}_1 \end{bmatrix}. \quad (15)$$

Rozwiązaniem równania liniowego (15) jest wektor współczynników:

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q}_0 \\ \mathbf{V}_0 \\ -3\mathbf{Q}_0 + 3\alpha \mathbf{Q}_1 - 2\mathbf{V}_0 - \alpha \mathbf{V}_1 \\ 2\mathbf{Q}_0 - 2\alpha \mathbf{Q}_1 + \mathbf{V}_0 + \alpha \mathbf{V}_1 \end{bmatrix}, \quad \mathbf{V}_i = \frac{1}{2}\boldsymbol{\omega}_i \mathbf{Q}_i + c_i \mathbf{Q}_i, \quad \alpha = \alpha_1/\alpha_0 \in \mathbb{R}. \quad (16)$$

Przykład 3.1. Rozważmy interpolację trzech elementów $SO(3)$ – reprezentowanych przez kwaterniony – na przedziale $[0, 2]$ za pomocą kubicznych funkcji sklepanych. Prędkość kątowa na początku i końcu trajektorii ma być równa 0. Prędkość kątowa

w środkowym węźle jest nieokreślona. Choć może być dobrana dowolnie, preferowana będzie jej wartość gwarantująca ciągłość przyspieszeń kątowych $\mathbf{\epsilon}$. Dane są:

$$\mathbf{P}_1(t) = t^{\{3\}} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{V}_1 \\ -3\mathbf{Q}_1 + 3\alpha\mathbf{Q}_2 - 2\mathbf{V}_1 - \alpha\mathbf{V}_2 \\ 2\mathbf{Q}_1 - 2\alpha\mathbf{Q}_2 + \mathbf{V}_1 + \alpha\mathbf{V}_2 \end{bmatrix}, \quad \mathbf{V}_i = \frac{1}{2}\omega_i\mathbf{Q}_i + c_i\mathbf{Q}_i, \quad \alpha \in \mathbb{R}, \quad (17)$$

$$\mathbf{P}_2(t) = t^{\{3\}} \begin{bmatrix} \alpha\mathbf{Q}_2 \\ \alpha\mathbf{V}_2 \\ -3\alpha\mathbf{Q}_2 + 3\beta\mathbf{Q}_3 - 2\alpha\mathbf{V}_2 - \beta\mathbf{V}_3 \\ 2\alpha\mathbf{Q}_2 - 2\beta\mathbf{Q}_3 + \alpha\mathbf{V}_2 + \beta\mathbf{V}_3 \end{bmatrix}, \quad \mathbf{V}_i = \frac{1}{2}\omega_i\mathbf{Q}_i + c_i\mathbf{Q}_i, \quad \alpha, \beta \in \mathbb{R}. \quad (18)$$

Sklejaną interpolantę można zapisać jako:

$$\mathbf{P}(t) = \begin{cases} \mathbf{P}_1(t), & \text{dla } 0 \leq t < 1, \\ \mathbf{P}_2(t-1), & \text{dla } 1 \leq t < 2. \end{cases} \quad (19)$$

Dodatkowo założona została ciągłość $\mathbf{P}(t)$ i $\mathbf{P}'(t)$ w $t = 1$ oraz $\alpha = \beta = 1$. Założenia te nie są konieczne, bowiem funkcja $\mathbf{P}(t)$ znajduje się w przestrzeni rzutowej, a z zależności (11) widać, że ciągłość prędkości kątowej nie wymaga ciągłości pierwszej pochodnej, jednak przyjęto je dla ułatwienia obliczeń. By zapewnić ciągłość $\mathbf{\epsilon}(1)$, należy rozwiązać równanie $\mathbf{\epsilon}_1(1) = \mathbf{\epsilon}_2(0)$. Z zależności (13) mamy:

$$\begin{aligned} \mathbf{\epsilon}_1(1) &= \frac{\ddot{\mathbf{P}}_1(1)\bar{\mathbf{P}}_1(1) - \mathbf{P}_1(1)\ddot{\mathbf{P}}_1(1) - \dot{\mathbf{N}}_1(1)\omega_1(1)}{N_1(1)}, \\ \mathbf{\epsilon}_2(0) &= \frac{\ddot{\mathbf{P}}_2(0)\bar{\mathbf{P}}_2(0) - \mathbf{P}_2(0)\ddot{\mathbf{P}}_2(0) - \dot{\mathbf{N}}_2(0)\omega_2(0)}{N_2(0)}. \end{aligned} \quad (20)$$

Zdefiniujemy ważną i dalej często wykorzystywaną operację $[\mathbf{p}, \mathbf{q}] := \mathbf{p}\bar{\mathbf{q}} - \mathbf{q}\bar{\mathbf{p}}$ (choć zapis to sugeruje, nie jest komutatorem algebry kwaternionowej) o własnościach:

$$\begin{aligned} [a\mathbf{p}, \mathbf{q}] &= [\mathbf{p}, a\mathbf{q}] = a[\mathbf{p}, \mathbf{q}], \\ [\mathbf{p}, \mathbf{q}] &= -[\mathbf{q}, \mathbf{p}] = -[\bar{\mathbf{p}}, \bar{\mathbf{q}}], \\ [\mathbf{p}_1 + \mathbf{p}_2, \mathbf{q}] &= [\mathbf{p}_1, \mathbf{q}] + [\mathbf{p}_2, \mathbf{q}], \quad [\mathbf{p}, \mathbf{q}_1 + \mathbf{q}_2] = [\mathbf{p}, \mathbf{q}_1] + [\mathbf{p}, \mathbf{q}_2], \\ [\mathbf{h}\mathbf{p}, \mathbf{q}] &= [\mathbf{h}, \mathbf{q}\bar{\mathbf{p}}], \end{aligned} \quad (21)$$

gdzie $\mathbf{p}_i, \mathbf{q}_i, \mathbf{p}, \mathbf{q}, \mathbf{h} \in \mathbb{H}$, $a \in \mathbb{R}$.

Z ciągłości $\mathbf{P}(t)$ w $t = 1$ wynika $N_2(0) = N_1(1)$, natomiast z ciągłości prędkości kątowych: $\omega(1) = \omega_1(1) = \omega_2(0)$ dając, po odpowiednich przekształceniach algebraicznych, zależność:

$$N(\mathbf{\epsilon}_1(1) - \mathbf{\epsilon}_2(0)) = [\ddot{\mathbf{P}}_1(1), \mathbf{P}_1(1)] - [\ddot{\mathbf{P}}_2(0), \mathbf{P}_2(0)] + \omega(\dot{N}_2(0) - \dot{N}_1(1)) = 0. \quad (22)$$

Po rozpisaniu:

$$\begin{aligned} [\ddot{\mathbf{P}}_1(1), \mathbf{P}_1(1)] &= [6\mathbf{Q}_0 + 2\dot{\mathbf{P}}_1(0), \mathbf{Q}_1] + 4N\omega, \\ [\ddot{\mathbf{P}}_2(0), \mathbf{P}_2(0)] &= [6\mathbf{Q}_2 - 2\dot{\mathbf{P}}_2(1), \mathbf{Q}_1] - 4N\omega, \end{aligned} \quad (23)$$

można wyznaczyć prędkość kątową w środkowym węźle:

$$\boldsymbol{\omega} = \frac{[(6 - 2c_2)\mathbf{Q}_2 - (6 + 2c_0)\mathbf{Q}_0, \mathbf{Q}_1]}{8N}. \quad (24)$$

Przykład 3.2. Niech danych będzie $(n + 1)$ węzłów interpolacyjnych o założonych, kolejno, orientacjach $\mathbf{Q}_0, \dots, \mathbf{Q}_n$, oraz $\boldsymbol{\omega}_0 = \boldsymbol{\omega}_n = 0$. Zdefiniujmy:

$$\mathbf{f}_i := [(6 - 2c_{i+1})\mathbf{Q}_{i+1} - (6 + 2c_{i-1})\mathbf{Q}_{i-1}, \mathbf{Q}_i].$$

Dla i -tego węzła równanie ciągłości przyspieszeń kątowych można zapisać następująco (rozpisując pochodne w (23)):

$$[\boldsymbol{\omega}_{i-1}, \mathbf{Q}_i \bar{\mathbf{Q}}_{i-1}] + 8\mathbf{Q}_i \bar{\mathbf{Q}}_i \boldsymbol{\omega}_i + [\boldsymbol{\omega}_{i+1}, \mathbf{Q}_i \bar{\mathbf{Q}}_{i+1}] = \mathbf{f}_i. \quad (25)$$

Dla zachowania ciągłości przyspieszeń kątowych na całej trajektorii, układ $(n - 1)$ liniowych równań kwaternionowych o $(n - 1)$ zmiennych $\boldsymbol{\omega}_i$, musi być spełniony:

$$\begin{cases} 8\mathbf{Q}_1 \bar{\mathbf{Q}}_1 \boldsymbol{\omega}_1 + [\boldsymbol{\omega}_2, \mathbf{Q}_1 \bar{\mathbf{Q}}_2] = \mathbf{f}_1, \\ \dots \\ [\boldsymbol{\omega}_{i-1}, \mathbf{Q}_i \bar{\mathbf{Q}}_{i-1}] + 8\mathbf{Q}_i \bar{\mathbf{Q}}_i \boldsymbol{\omega}_i + [\boldsymbol{\omega}_{i+1}, \mathbf{Q}_i \bar{\mathbf{Q}}_{i+1}] = \mathbf{f}_i, \quad i = 2, \dots, n - 2 \\ \dots \\ [\boldsymbol{\omega}_{n-2}, \mathbf{Q}_{n-1} \bar{\mathbf{Q}}_{n-2}] + 8\mathbf{Q}_{n-1} \bar{\mathbf{Q}}_{n-1} \boldsymbol{\omega}_{n-1} = \mathbf{f}_{n-1}. \end{cases} \quad (26)$$

Istnieją techniki rozwiązywania pojedynczych równań pozostając w algebrze kwaternionowej [8]. Choć matematycznie eleganckie, z perspektywy praktyczniej prościej jest zamienić ten układ równań na układ równań liniowych wykorzystując tak zwany operator kolumnowy [4]. Niech $\mathbf{col}(\mathbf{a}) = [a_0 \ a_1 \ a_2 \ a_3]^T$ i

$$\mathbf{l}_1(\mathbf{a}) = \begin{bmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{bmatrix}, \quad \mathbf{l}_2(\mathbf{a}) = \begin{bmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & a_3 & -a_2 \\ a_2 & -a_3 & a_0 & a_1 \\ a_3 & a_2 & -a_1 & a_0 \end{bmatrix}. \quad (27)$$

Dla $\mathbf{a}, \mathbf{b} \in \mathbb{H}$ spełniona jest użyteczna własność:

$$\mathbf{col}(\mathbf{ab}) = \mathbf{l}_1(\mathbf{a})\mathbf{col}(\mathbf{b}) = \mathbf{l}_2(\mathbf{b})\mathbf{col}(\mathbf{a}).$$

Z pomocą tych operatorów możemy zapisać dla $\mathbf{Q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3$:

$$\mathbf{col}([\boldsymbol{\omega}, \mathbf{Q}]) = (\mathbf{l}_2(\bar{\mathbf{Q}}) - \mathbf{l}_1(\mathbf{Q})\mathbf{O})\mathbf{col}(\boldsymbol{\omega}) = 2 \begin{bmatrix} 0 & 0 & 0 & 0 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \mathbf{col}(\boldsymbol{\omega}),$$

gdzie $\mathbf{O}$ jest macierzą spełniającą $\mathbf{col}(\bar{\mathbf{x}}) = \mathbf{Ocol}(\mathbf{x})$. Ponieważ z definicji $\boldsymbol{\omega}$ ma zerową część skalarną, wartości w pierwszej kolumnie nie mają wpływu na rozwiązanie równania, zatem zdefiniujmy:

$$\mathbf{col}([\boldsymbol{\omega}, \mathbf{Q}]) := 2 \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

Zapisując operatorowo $\text{col}([\omega, \mathbf{Q}_i \bar{\mathbf{Q}}_j]) = \mathbf{G}_{i,j} \text{col}(\omega)$ (zauważając że $\mathbf{G}_{i,j} = \mathbf{G}_{j,i}^T$), można przedstawić (26) jako:

$$\underbrace{\begin{bmatrix} 8\mathbf{Q}_1 \bar{\mathbf{Q}}_1 \mathbf{I}_3 & \mathbf{G}_{1,2} & & \\ \mathbf{G}_{1,2}^T & 8\mathbf{Q}_2 \bar{\mathbf{Q}}_2 \mathbf{I}_3 & \mathbf{G}_{2,3} & \\ & \ddots & \ddots & \\ & & \mathbf{G}_{n-2,n-1}^T & 8\mathbf{Q}_{n-1} \bar{\mathbf{Q}}_{n-1} \mathbf{I}_3 \end{bmatrix}}_{\mathbf{A}(\mathbf{Q}_1, \dots, \mathbf{Q}_{n-1})} \begin{bmatrix} \text{col}(\omega_1) \\ \text{col}(\omega_2) \\ \vdots \\ \text{col}(\omega_{n-1}) \end{bmatrix} = \begin{bmatrix} \text{col}(f_1) \\ \text{col}(f_2) \\ \vdots \\ \text{col}(f_{n-1}) \end{bmatrix}. \quad (28)$$

Macierz współczynników jest symetryczną, trójdziagonalną macierzą blokową o stałych współczynnikach – stałych jeżeli nie parametryzujemy wielokrotności $\mathbf{Q}_i$. Nie trudno jednak uwzględnić zależność od norm indywidualnych orientacji $\mathbf{Q}_i$ w węzłach interpolacyjnych, zakładając przy tym dalej ciągłość $\mathbf{P}(t)$. Podstawiając $\mathbf{Q}_i \rightarrow \alpha_i \mathbf{Q}_i$, można zauważyć, że:

$$\begin{aligned} \tilde{f}_n := f_n(c_{n-1}, c_{n+1}, \alpha_{n-1}, \alpha_n, \alpha_{n+1}) &= \underbrace{6[\alpha_{n+1} \mathbf{Q}_{n+1} - \alpha_{n-1} \mathbf{Q}_{n-1}, \alpha_n \mathbf{Q}_n]}_{\mathbf{F}_n(\alpha_{n-1}, \alpha_n, \alpha_{n+1})} \\ &\quad - \underbrace{2[c_{n+1} \alpha_{n+1} \mathbf{Q}_{n+1} + c_{n-1} \alpha_{n-1} \mathbf{Q}_{n-1}, \alpha_n \mathbf{Q}_n]}_{c_{n-1} \mathbf{G}_n(\alpha_{n-1}, \alpha_n, \alpha_{n+1}) + c_{n+1} \mathbf{H}_n(\alpha_{n-1}, \alpha_n, \alpha_{n+1})}. \end{aligned}$$

Warto zaznaczyć, że $f_n(0, 0, \alpha_{n-1}, \alpha_n, \alpha_{n+1}) = \mathbf{F}_n(\alpha_{n-1}, \alpha_n, \alpha_{n+1})$ oraz że:

$$\alpha \mathbf{A}(\mathbf{Q}_1, \dots, \mathbf{Q}_{n-1}) \alpha \begin{bmatrix} \text{col}(\omega_1) \\ \text{col}(\omega_2) \\ \vdots \\ \text{col}(\omega_{n-1}) \end{bmatrix} = \begin{bmatrix} \text{col}(\tilde{f}_1) \\ \text{col}(\tilde{f}_2) \\ \vdots \\ \text{col}(\tilde{f}_{n-1}) \end{bmatrix}, \quad (29)$$

gdzie $\alpha = \text{diag}(\mathbf{I}_3 \alpha_1, \dots, \mathbf{I}_3 \alpha_{n-1})$. Niech $\mathbf{Q}_i \bar{\mathbf{Q}}_i = 1$, dla:

$$\tilde{\mathbf{A}}(\mathbf{Q}_1, \dots, \mathbf{Q}_{n-1}) = \begin{bmatrix} 8\mathbf{I}_3 & \mathbf{G}_{1,2} & & \\ \mathbf{G}_{1,2}^T & 8\mathbf{I}_3 & \mathbf{G}_{2,3} & \\ & \ddots & \ddots & \\ & & \mathbf{G}_{n-2,n-1}^T & 8\mathbf{I}_3 \end{bmatrix}, \quad (30)$$

można zapisać ogólne rozwiązanie (29) jako:

$$\begin{bmatrix} \text{col}(\omega_1) \\ \text{col}(\omega_2) \\ \vdots \\ \text{col}(\omega_{n-1}) \end{bmatrix} = \alpha^{-1} \tilde{\mathbf{A}}(\mathbf{Q}_1, \dots, \mathbf{Q}_{n-1})^{-1} \alpha^{-1} \begin{bmatrix} \text{col}(\tilde{f}_1) \\ \text{col}(\tilde{f}_2) \\ \vdots \\ \text{col}(\tilde{f}_{n-1}) \end{bmatrix}. \quad (31)$$

Dla uzyskania wszystkich rozwiązań, przy ustalonej sekwencji orientacji, należy odwrócić jedną macierz trójdziagonalną i jedną diagonalną.

Rezultaty poprzedzającego przykładu możemy podsumować w poniższym twierdzeniu, stanowiącym również główny rezultat referatu:

Twierdzenie 3.1. *Dla sekwencji $n + 1$ orientacji reprezentowanych przez kwaterniony $\mathbf{Q}_0, \dots, \mathbf{Q}_n$, oraz przy założeniu zerowych prędkości brzegowych $\boldsymbol{\omega}_0 = \boldsymbol{\omega}_n = 0$, istnieje $2n$ -parametrowa rodzina sklepanych, kubicznych funkcji interpolujących zadane orientacje, zapewniając przy tym ciągłość przyspieszeń kątowych $\boldsymbol{\varepsilon}(t)$ w węzłach pośrednich. Między więzami interpolacyjnymi funkcje te dane są jako rozwiązanie (15), gdzie prędkości kątowe pozyskiwane są z rozwiązania (29).*

Skoro wyznaczenie rozwiązania zależy od odwracalności pewnych macierzy występujących w zależności (31), warto podać warunki na ich odwracalność. Jeżeli wszystkie $\alpha_i \neq 0$, odwracalność $\mathbf{A}$ decyduje o tym, czy równanie jest osobliwe czy nie. Odwracalność tej macierzy blokowej można pokazać z pomocą norm operatorowych [2]. Jeżeli wszystkie diagonalne bloki $\mathbf{B}_{ii}$ macierzy blokowej $\mathbf{B}$ są odwracalne, oraz spełniony jest warunek:

$$\|\mathbf{B}_{ii}^{-1}\|^{-1} > \sum_{j \neq i} \|\mathbf{B}_{ij}\|.$$

Dla jakiegś normy macierzowej $\|\cdot\|$, macierz $\mathbf{B}$ jest blokowo przekątniowo dominująca, a zatem też odwracalna. W przypadku macierzy $\tilde{\mathbf{A}}$ diagonalne bloki są oczywiście odwracalne, a powyższy warunek zapisać można jako:

$$8 > \|\mathbf{G}_{n,n+1}\| + \|\mathbf{G}_{n+1,n+2}\|. \quad (32)$$

Użyjemy normy $\|\mathbf{B}\|_2 = \sqrt{\max \sigma(\mathbf{B}\mathbf{B}^T)}$. Łatwo wykazać, że największa wartość własna $\mathbf{G}_{i,j}\mathbf{G}_{i,j}^T$ jest równa $\sigma = 4(q_0^2 + q_1^2 + q_2^2 + q_3^2)$. Z założenia jednostkowych norm kwaternionów $\mathbf{Q}_i$, oraz właściwości norm iloczynu kwaternionów $\|\mathbf{Q}_i\mathbf{Q}_j\| = \|\mathbf{Q}_i\| \cdot \|\mathbf{Q}_j\|$ mamy $\|\mathbf{G}_{i,j}\| = \sqrt{4(q_0^2 + q_1^2 + q_2^2 + q_3^2)} = 2$. Zatem nierówność (32) jest spełniona w każdym przypadku. Osobliwości w (31) pojawiają się, gdy macierz $\boldsymbol{\alpha}$ nie jest odwracalna, co ma miejsce gdy α_i , $i \in 1, \dots, n-1$ wynosi 0. Oznacza to, że nie można w sposób ciągły przekształcić interpolanty dla której $\alpha_i > 0$ w taką z $\alpha_i < 0$. Można więc stwierdzić, że dla danej sekwencji $(n+1)$ orientacji, przestrzeń rozwiązań podzielona jest na 2^{n-2} rozłącznych podzbiorów warunkowanych znakami α_i , $i = 1, \dots, n-1$.

Czytelnikom zaznajomionym ze strukturą $SO(3)$ i krzywymi geodezyjnymi w tej przestrzeni może narzucić się naturalne pytanie: co się stanie, gdy następujące po sobie orientacje są antypodalne? Krzywa geodezyjna zdefiniowana jest jako najkrótsza krzywa łącząca dwa punkty na zadanej rozmaitości, względem pewnej normy. Antypodalne orientacje na $SO(3)$ można połączyć nieskończenie wieloma krzywymi geodezyjnymi, co jest pewnego rodzaju osobliwością generującą niejednoznaczności w wyborze konkretnej krzywej podczas planowania. Przedstawiony algorytm jednak nie wykorzystuje krzywych geodezyjnych, zatem powyższy problem nie występuje. Dla dwóch zadanych punktów w $\mathbb{RP}^3$ i dwóch danych prędkości kątowych istnieje rodzina krzywych kubicznych je łącząca i spełniająca zadane warunki na prędkości

brzegowe, która jest rozwiązaniem (15). Początkowy i końcowy kierunek obrotu określony jest przez warunki brzegowe, całość krzywej jednak zależy od wcześniej wymienionych parametrów c_i oraz α_i . Pytaniem otwartym pozostaje więc: jak te parametry celowo dobierać? Nie można oczekiwać, że możliwym będzie zadanie sekwencji prędkości kątowych w węzłach i na ich podstawie wyznaczyć z (31) wymagane wartości parametrów. Można natomiast próbować aproksymować zadane prędkości numerycznie, bądź próbować znaleźć najlepszą możliwą aproksymację zadanego wektora prędkości symbolicznie.

Alternatywnie, postać (31) sugeruje zastosowania w programach CAD. W tym kontekście, prostota obliczeniowa pozwoliłaby na interaktywne projektowanie funkcji interpolacyjnych. Problemem w tym przypadku jest uwikłanie parametrów w wyznaczonym wektorze prędkości kątowych w węzłach, co utrudnia celową, interaktywną modyfikację profili prędkościowych.

4. Symulacje

Przedstawione zostaną profile prędkościowe trajektorii interpolujących pięć orientacji przy pomocy kubicznych, $N = 3$, funkcji sklepanych. Należy zaznaczyć, że profile te zależą nie tylko od zadanych orientacji, lecz także od norm reprezentujących ich kwaternionów, oraz parametrów c_i . Zależności te są wysoce uwikłane i trudno przewidzieć jak zmiana wartości jednej zmiennej wpłynie na całą trajektorię.

Przykład 4.1. *Dana jest sekwencja kwaternionów reprezentujących pięć orientacji:*

$$\mathbf{Q}_0 = 1 + i2 + j3 + k4 \quad \mathbf{Q}_1 = 2 - i + k \quad \mathbf{Q}_2 = 0.5 + k4 \quad \mathbf{Q}_3 = 1 + i - j + k \quad \mathbf{Q}_4 = 1 + i,$$

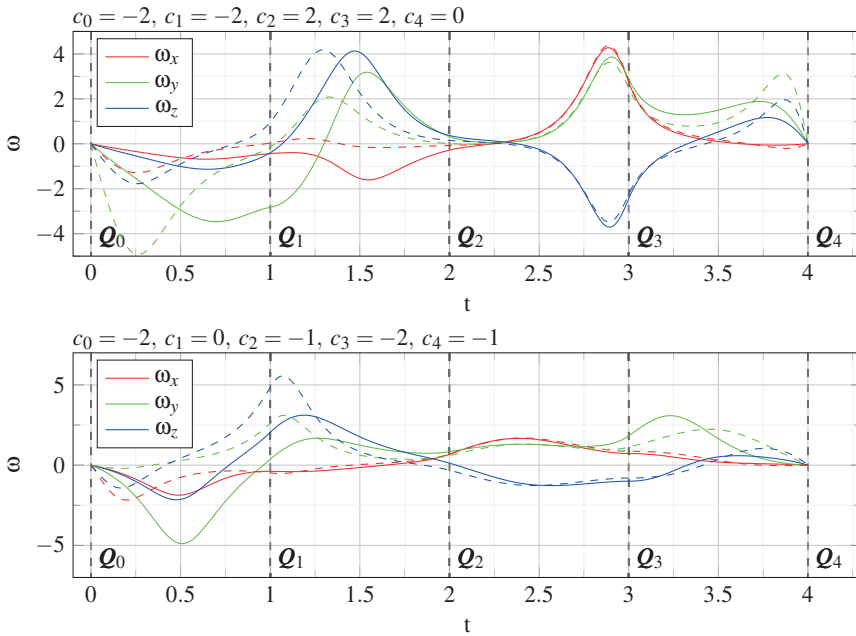
oraz sekwencja reprezentująca te same orientacje z pomocą kwaternionów o innych normach:

$$\tilde{\mathbf{Q}}_0 = \mathbf{Q}_0/4, \quad \tilde{\mathbf{Q}}_1 = \mathbf{Q}_1, \quad \tilde{\mathbf{Q}}_2 = 2\mathbf{Q}_2, \quad \tilde{\mathbf{Q}}_3 = 2\mathbf{Q}_3, \quad \tilde{\mathbf{Q}}_4 = \mathbf{Q}_4/\sqrt{2}.$$

Na podstawie twierdzenia 3.1 wyznaczono funkcję interpolacyjną $P(t)$. Na wykresach 1 przedstawiono $\boldsymbol{\omega} = (\dot{\mathbf{P}}\mathbf{P} - \mathbf{P}\dot{\mathbf{P}})/N$ jako funkcję wolnego parametru t .

5. Podsumowanie

W referacie przedstawiono metodę interpolacji orientacji za pomocą wielomianowych funkcji kwaternionowych jednej zmiennej. Metoda pozwala na zachowanie ciągłości przyspieszeń kątowych, pozostawiając przy tym wolne parametry, które można wykorzystać do kształtowania profili prędkościowych, bądź spełniania innych ograniczeń. Praktyczne i celowe kształtowanie tych parametrów nie jest obecnie jasne. Rozwiązania (31) wskazują, że prędkości w każdym węźle zależą od wszystkich parametrów, zatem zmodyfikowanie parametru intuicyjnie przypisanego do konkretnego węzła wpłynie na całość trajektorii oraz prędkości we wszystkich węzłach. Metodę przedstawiono dla interpolacji orientacji, jednak wydaje się, że jest rozszerzalna do przestrzeni $SE(3)$, by umożliwić interpolację także pozycji z zachowaniem ciągłości przyspieszeń kątowych i liniowych. Należy oczekiwać, że kwaterniony zostaną zastąpione kwaternionami dualnymi, co jest przedmiotem obecnych prac.



Rys. 1. Przebiegi prędkości kątowych dla kubicznych krzywych sklejanych i wybranych parametrów c_i . Linie ciągłe – dla sekwencji $\mathcal{Q}_i$, przerywane – dla sekwencji $\tilde{\mathcal{Q}}_i$

Literatura

- [1] O. Bottema, B. Roth. *Theoretical Kinematics*. Dover Books on Physics. Dover Publications 1990.
- [2] C. Echeverría, J. Liesen, R. Nabben. Block diagonal dominance of matrices revisited: Bounds for the norms of inverses and eigenvalue inclusion sets. *Linear Algebra and its Applications*, 2018, wolumen 553, s. 365–383.
- [3] Q.J. Ge, B. Ravani. Computer aided geometric design of motion interpolants. *Journal of Mechanical Design*, 09, 1994, wolumen 116, numer 3, s. 756–762.
- [4] D. Janovska, G. Opfer. Linear equations in quaternionic variables. *Mitteilungen der Mathematischen Gesellschaft in Hamburg*, 01, 2008, wolumen 27.
- [5] M-J Kim, M-S Kim, Y Shin, S. A compact differential formula for the first derivative of a unit quaternion curve. *Journ. of Visualization and Computer Animation*, 1996, wolumen 7, numer 1, s. 43–57.
- [6] C. Mancinelli, E. Puppo. Splines on manifolds: A survey. *Computer Aided Geometric Design*, 2024, wolumen 112, s. 102349.
- [7] F.C. Park, B. Ravani. Smooth invariant interpolation of rotations. *ACM Trans. Graph.*, 1997, wolumen 16, numer 3, s. 277–295.
- [8] C. Shao, H. Li, L. Huang. Basis-free solution to general linear quaternionic equation. *Linear and Multilinear Algebra*, 2020, wolumen 68, numer 3, s. 435–457.
- [9] E.V. Shikin, A.I. Plis. *Handbook on Splines for the User*. Taylor & Francis 1995.
- [10] K. Shoemake. Animating rotation with quaternion curves. In: *Conf. on Computer Graphics and Interactive Techniques. Proceedings*, New York, Association for Computing Machinery, 1985, SIGGRAPH, s. 245–254.

- [11] J. Tan et al. Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve. *Journal of Computational and Applied Mathematics*, 2018, wolumen 329, s. 256–267. The International Conference on Information and Computational Science, 2–6 August 2016, Dalian, China.

Cubic interpolation of orientation in projective space

In this paper a spline interpolation method is transferred from linear spaces into interpolation in the orientation space. For this purpose, the quaternion algebra and particular properties of the projective space $\mathbb{RP}^3$ (diffeomorphic with a special orthogonal group, $SO(3)$ describing orientations) are exploited. The corresponding formulas for single segment and for multi-segment interpolation with cubic spline polynomials preserving continuity of accelerations are derived. The considerations are illustrated by simulations on two numerical examples for a set of selected free parameters, which parameterize the family of admissible solutions of the interpolation task.

PERSPEKTYWY ROBOTYKI

Wyzwania i perspektywy rozwoju technologii robotycznych w kontekście zrównoważonego budownictwa

Ryszard Dindorf¹, Jakub Takosoglu¹, Piotr Woś¹

Streszczenie

W artykule omówiono wyzwania i perspektywy rozwoju technologii robotycznych w kontekście zrównoważonego budownictwa. Przedstawiono nowe technologie cyfrowe i robotyczne stosowane w budownictwie. Scharakteryzowano technologie robotyczne w budownictwie ze względu na rodzaje robotów, cele zrównoważonego rozwoju, wyzwania technologiczne, ekonomiczne, behawioralne i bezpieczeństwa oraz różne czynniki ograniczające. Przedstawiono stosowane na świecie zrobotyzowane urządzenia do automatycznego murowania. Omówiono wyniki demonstracyjnego murowania na placu budowy przy stawianiu ścian działowych wewnątrz budynku mieszkalnego przez zrobotyzowany systemy murarski (ZSM).

1. Wprowadzenie

W sektorze budowlanym zachodzą rewolucyjne zmiany związane z wprowadzeniem nowych i innowacyjnych technologii wpływające na poprawę struktury, organizacji i efektywność budownictwa [1]. Do nowych technologii wdrażanych w branży budowlanej zalicza się komunikację 5G, rozwiązania oparte na chmurze, wspólne środowiska danych CDE (Common Data Environments), modelowanie informacji o budynku BIM (Building Information Modeling), automatyzację i robotyzację, drony, skanowanie 3D, rzeczywistość rozszerzoną XR (Extended Reality), druk 3D, sztuczną inteligencję, bliźniaki cyfrowe, nowe materiały budowlane oraz rozwiązania proekologiczne [13]. Zastosowanie druku 3D znanego jako technologia addytywna wpływa na zrationalizowane zużycie materiałów budowlanych i dostosowuje proces budowlany do nowej technologii. BIM wprowadza cyfryzację do robotyzacji technologii budowlanych przez umożliwienie dostępu do wszystkich informacji projektowych i danych zbieranych w czasie rzeczywistym na placu budowy. XR, jako zbiorcza technologia wirtualna, wspiera technologie cyfrowe i robotyczne na budowie. Technologie cyfrowe i robotyczne są kluczowymi czynnikami umożliwiającymi wdrożenie gospodarki o obiegu zamkniętym, jako ekonomii cyrkularnej CE (Circular Economy) w budownictwie, przechodząc od liniowego modelu konsumpcji (użytkowanie–konsumpcja–utylizacja) do modelu cyrkularnego (użytkowanie–odzyskiwanie–recyrkulacja). Szacuje się, że w 2025

¹ Katedra Mechatroniki i Uzbrojenia, Wydział Mechatroniki i Budowy Maszyn, Politechnika Świętokrzyska, al. Tysiąclecia Państwa Polskiego 7, 25-314 Kielce, <https://tu.kielce.pl/>, dindorf@tu.kielce.pl, gba@tu.kielce.pl, wos@tu.kielce.pl.

roku w polskim budownictwie wrosną koszty na poziomie od 5% do 6% związane z inwestycjami, technologiami, materiałami i pracą ludzką [15]. Branża budowlana stoi przed wyzwaniem transformacji nowych technologii informatycznych i robotycznych, aby sprostać celom zrównoważonego rozwoju. Zrównoważone budownictwo to koncepcja, która łączy w sobie ekologiczne rozwiązania z nowoczesnymi technologiami, mając na celu minimalizację negatywnego wpływu budownictwa na środowisko.

Jedną z technologii cyfrowych wykorzystywanych przez architektów i inżynierów budowlanych są narzędzia modelowania BIM dostępne w programie Autodesk Revit. Autodesk Revit umożliwia użytkownikom tworzenie, edytowanie i przeglądanie modeli budynków w trójwymiarowym środowisku 3D. Modele BIM w Autodesk Revit automatycznie aktualizują wszelkie wprowadzone zmiany w projekcie budowlanym, ułatwiają sprawdzanie i wykrywanie wszystkich niespójności projektowych. Wirtualny model BIM budynku stworzony w Autodesk Revit, obejmujący budynek w trakcie budowy, podział ścian wewnętrznych i wymiary wydzielonych ścian bazowych. Modele budynków nie tylko zawierają atrybuty fizyczne (geometryczne), a również zachowują swoje właściwości funkcjonalne. Wirtualne modele BIM budynków utworzone w programie Autodesk Revit można dostosować do technologii robotycznych, np. zrobotyzowanego murowania, przez zintegrowanie ich z narzędziami ABB RobotStudio. W aplikacji ABB RobotStudio dostępny jest dodatek do eksportu geometrii budynku, w związku z tym geometrię ścian można przenieść do zrobotyzowanego murowania. Na rys. 1 przedstawiono schemat blokowy zrobotyzowanego murowania przy użyciu modeli BIM w Autodesk Revit i narzędzi symulacyjnych ABB RobotStudio.

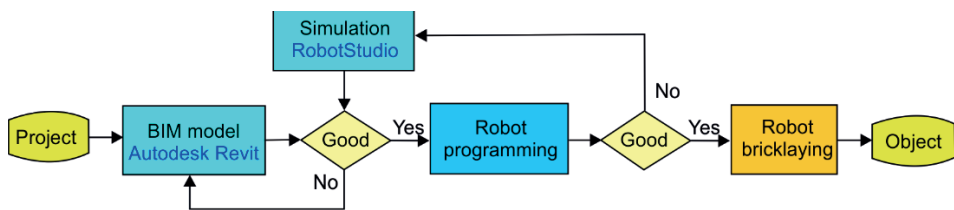


Fig. 1. Schemat blokowy procedury zrobotyzowanego murowania

2. Technologie robotyczne w budownictwie

Branża budowlana ma duży potencjał wykorzystania nowych technologii robotycznych do wykonywania wielu rodzaju zadań budowlanych. W robotyzacji budownictwa wykorzystuje się roboty przemysłowe, ale nazwy używanych robotów związane są z określonym zadaniem budowlanym, jak roboty: murarskie, tynkarskie, malarskie, rozbiórkowe, spawalnicze, transportowe, architektoniczne, montażowe, prefabrykacyjne, do drukowania 3D, skanujące, inspekcyjne, monitorujące, a także drony i bezzałogowe statki powietrzne UAV (Unmanned Aerial Vehicle).

W budownictwie wykorzystuje się kilka rodzajów robotów przemysłowych, takich jak: roboty SCARA do prefabrykacji i konstrukcji modułowych; roboty przegubowe do wykonywania szerokiego zakresu zadań jak spawanie, malowanie, tynkowanie, murowanie, montaż; roboty cylindryczne do zadań wymagających ruchu pionowego i obrotowego przy obsłudze maszyn i pakowaniu; roboty kartezyjańskie do drukowania 3D i obsługi materiałów; roboty równoległe typu Delta do operacji pick-and-place, sortowania i pakowania materiałów; roboty współpracujące (coboty) do obsługi lekkich materiałów, wykonywania prac montażowych oraz pomocniczych.

W budownictwie mogą być stosowane roboty o różnym stopniu autonomiczności: roboty półautonomiczne, które potrzebują początkowego udziału pracownika, po czym mogą samodzielnie wykonywać zadania budowlane; roboty autonomiczne, które nie wymagają żadnej pomocy człowieka; roboty zdalnie sterowane, których sterowanie polega na pewnym stopniu interakcji z teleoperatorem.

Ze względu na cele zrównoważonego budownictwa nowe technologie robotyczne umożliwiają rozwój i postęp w zakresie:

- Poprawy wydajności i jakości: roboty budowlane mogą wykonywać zadania dokładniej niż pracownicy, zapewniają wyższą wydajność i jakość prac budowlanych przy mniejszej liczbie błędów.
- Zwiększenia bezpieczeństwa pracy: przy spełnieniu wymaganych norm bezpieczeństwa stanowiska zrobotyzowane zapewniają wyższy poziom bezpieczeństwa na budowie.
- Zmniejszenie ilości odpadów: zastosowanie robotów przemysłowych w budownictwie przyczynia się do zmniejszenia ilości odpadów budowlanych oraz racjonalizacji zużycia materiałów na budowie.
- Optymalizacji i symulacji: nowe technologie cyfrowe i robotyczne umożliwiają optymalizację i symulację projektów budowlanych, co prowadzi do podejmowania racjonalnych decyzji podczas prac budowlanych.
- Monitorowania w czasie rzeczywistym: nowe technologie cyfrowe i robotyczne stosowane do monitorowania placu budowy w czasie rzeczywistym prowadzą do optymalizacji prac budowlanych i terminowych realizacji inwestycji.

Wdrażanie nowych technologii robotycznych w budownictwie związane jest z takim wyzwaniem, jak:

- Rozproszona informacja: największym wyzwaniem przy wdrożenia technologii robotycznych na placach budowy jest rozproszona informacja związana z różnymi systemami pozyskiwania, przechowywania i przekazywania danych.
- Cyberbezpieczeństwo: cyberbezpieczeństwo związane jest z dokładnością, bezpieczeństwem i zaufaniem do przekazywanych danych wykorzystywanych w technologiach cyfrowych i robotycznych.
- Zachowania behawioralne i kulturowe: opór przed zmianami wśród interesariuszy, kadry kierowniczej i pracowników stanowi przeszkodę w wdrażaniu technologii robotycznych.

- Bariery finansowe: Wysokie koszty wdrożenia technologii robotycznych stanowią poważną przeszkodę dla wielu firm budowlanych.
- Bezpieczeństwo i kwestie etyczne: w miarę jak technologie robotyczne staną się coraz bardziej powszechne w budownictwie będą musiały być rozwiązane problemy bezpieczeństwa i kwestie etyczne dotyczące pracowników.
- Integracja i interoperacyjność: technologie robotyczne wymagają opracowywania zintegrowanego i interoperacyjnego systemu informatycznego do określonych zadań budowlanych.
- Badania i rozwój: branża budowlana może sprostać wszystkim wyzwaniom technologicznym, gdy będzie inwestować w badania i rozwój, a także współpracować ze wszystkimi interesariuszami.

W ocenie ekonomicznej opłacalność robotyzacji sektora budowlanego uwzględnia się następujące koszty:

1. Koszty rozwoju obejmujące całkowite wydatki związane z siłą roboczą, zasobami i infrastrukturą wykorzystywaną do badania, eksperymentowania i oceny różnych rozwiązań systemów robotycznych.
2. Koszty inwestycyjne, w tym takie składniki, jak amortyzacja wartości w czasie i odsetki zapłacone od inwestycji, itp.
3. Koszty konfiguracji systemu informatycznego i technicznego do wymagań instalacji robota i sprzętu budowlanego na placu budowy.
4. Koszty konserwacji i naprawy robotów oraz sprzętu budowlanego.
5. Koszty operacyjne uwzględniające składniki materiałowe, energię, wynagrodzenia i inne związane z pracą robotów i sprzętu budowlanego.
6. Koszty pośrednie obejmujące koszty ogólne i koszty zarządzania związane z wprowadzaniem technologii robotycznych.

Główne czynniki ograniczające rozwój technologii robotycznych w branży budowlanej:

1. Duża zmienność topografii i lokalizacji budynków.
2. Infrastruktura budowlana nie jest przygotowana do integracji z technologiami cyfrowymi i robotycznymi.
3. Niski poziom przygotowania i wykorzystania systemów informatycznych do potrzeb technologii robotycznych.
4. Brak wykwalifikowanych pracowników budowlanych do obsługi i konserwacji zrobotyzowanych stanowisk pracy.
5. Wysoki koszt jednostkowy zrobotyzowanego stanowiska pracy.
6. Sceptyczne podejście inżynierów, pracowników budowlanych, kadry zarządzającej, interesariuszy i klientów do innowacyjnych technologii robotycznych (istnieje silna tendencja do utrzymywania ustalonych praktyk).
7. Łańcuchy dostaw materiałów budowlanych nie są dostosowane do organizacji zrobotyzowanej pracy na budowie.
8. Istnienie różnych regionalnych rozwiązań dotyczących rynku budowlanego, kosztów pracy i materiałów, norm, przepisów i uregulowań prawnych, itp.

W ETH Zurich (Szwajcaria) realizowano zadania zastosowania robotów przemysłowych ABB jako robotów architektonicznych do montażu na sucho (bez materiału wiążącego) złożonych konstrukcji z drewna, betonu, cegły i pianki na budowach w Zurychu, Londynie, Barcelonie i Nowym Jorku [14].

Przykłady wykorzystania robotów na budowach w Polsce:

1. Firma Budimex wprowadziła robot współpracujący na budowę, który może transportować materiały, wiercić w ścianie, wykonywać natryski powierzchniowe oraz składać proste szalunki [18]. Robot jest sterowany gestami, za pomocą tabletu lub manualnie w trybie „Follow Me”. Nawigacja robota odbywa się w oparciu o kodowane znaczniki lub przy wykorzystaniu modelu BIM.
2. Firma CB RTP (Centrum Badań i Rozwoju Technologii dla Przemysłu) w ramach projektu badawczego POIR.04.01.02-00-0045/18 wykonała zrobotyzowany system tynkarski (ZST), który był testowany na placu budowy [19]. ZST tynkuje w miarę wydajnie i sprawnie, ale problemem są nadmierne braki materiału tynkarskiego i dokładności tynkowania.

3. Automatyzacja i robotyzacja murowania

W miarę jak otaczający nas świat stale urbanizuje się, wzrasta zapotrzebowanie na prace murarskie przy braku wykwalifikowanych pracowników. Rozwiązaniem tego problemu jest automatyzacja i robotyzacja murowania, które według badań ankietowych [7] są coraz bardziej doceniane w budownictwie światowym. Roboty murarskie (struktury kinematyczne robotów przemysłowych) są wykorzystywane do wykonywania jednego z najbardziej czasochłonnnych i fizycznie wymagających zadań w budownictwie, jak układanie cegieł. Roboty wspierając murarzy, wykonują powtarzalne i ciężkie fizycznie zadania murarskie, podczas gdy pracownicy mogą wykonywać bardziej złożone i specjalistyczne zadania. Półautomatyczne systemy murarskie układają cegły, podczas gdy pracownicy wykonują zadania takie jak nakładanie zaprawy i wyrównując rozmieszczenie cegieł. Automatyczne systemy murarskie wyposażone są w czujniki i oprogramowanie wykonują cały proces murowania, jak pobieranie cegły, nakładanie zaprawy i precyzyjne układanie cegły zgodnie z złożeniami projektowymi. Zrobotyzowane murowanie jest ograniczone przy układaniu cegieł w warstwach przysufitowych i bocznych, wnękach, podczas murowania nadproży, narożników, itp.

W ramach grantu badawczego na Uniwersytecie Technologii i Ekonomii w Budapeszcie (Węgry) przeprowadzono badania oceny czasów automatycznego murowania prostych ścian na dwa sposoby: przez optymalizację tradycyjnej metody murowania oraz przez zastosowanie niekonwencjonalnej metody murowania [9]. Zastosowano 4-osiowy robot Dobot Magician (dydaktyczny) przesuwany na szynie i wyposażony w chwytak pneumatyczny. Cegły były pobierane z palety i układane bez zaprawy w ścianie według różnych wzorców. Do programowania prostych zadań robota stosowano aplikację DobotStudio, a do bardziej złożonych zadań zastosowano język programowania Python.

Firma Wienerberger A.G. w Republice Czeskiej przedstawiła robot murarski do murowania długich, prostych ścian z wydajnością do 10 m² na godzin do wysokości 2,75 m. Robot był testowany na placu budowy GEMO a.s. w Šumperku [10]. Operator używa tabletu do ustawienia robota w miejsca murowania, robot samodzielnie pobiera cegłę z palety, nakłada zaprawę i układa ją w określonym miejscu. Do murowania używane są specjalne cegły RR (Robot-ready), podobne do standardowych cegieł, ale mające specjalne rowki do chwytania. Robot nie może wykonać pierwszego rzędu fundamentów ścian, a także narożników.

Prototyp autonomicznego zrobotyzowanego system budowlanego ARCS (Autonomous Robotic Construction System) wykonano w ramach czeskiego projektu badawczego [8]. System ARCS wykorzystuje oprogramowanie BIM do tworzenia cyfrowego planu rozmieszczenia ścian dla zrobotyzowanego murowania. Zastosowana została konwersja modelu BIM z uwzględnieniem reprezentacji granic BREP (Boundary Representation) do cyfrowych rozwiązań interoperacyjnych IFC (Industry Foundation Classes). Cyfrowy model pozwala na generowanie kodu KRL (Kuka Robotic Language) z interfejsem do robota przemysłowego KUKA Quantec KR120 R3900 bazującego na sterowniku KRC4.

Firma Ballast Nedam (Holandia) opracowała i zbudowała robot murarski, który został wdrożony w ramach projektu niskoenergetycznego budownictwa Tuinbuurt Vrijlandt w Rotterdamie [11]. Robot murarski realizuje cele zrównoważonego budownictwa, ponieważ zapobiega marnotrawstwu surowców, zmniejszając ilość odpadów kamiennych do 70%. Zaprawa jest precyzyjnie dozowana na cegły, wtedy zużywa się 455 gramów zaprawy na cegłę, podczas gdy tradycyjny murarz zużywa 1000 gramów na cegłę. Firma Ballast Nedam inwestuje w innowacyjne technologie, a przez wdrażanie nowych technologii robotycznych przekształca plac budowy w nowoczesny zakład produkcyjny.

Firma Construction Robotization z Pocklington/Yorkshire (Wielka Brytania) wykonała zautomatyzowany robot murarski ABLR (Automated Brick Laying Robot) prowadzony na szynach, który jest przeznaczony do zewnętrznego murowania z dowolnego rodzaju cegły przy użyciu różnych rodzaju zaprawy [12]. ABLR jest wyposażony w zaawansowane oprogramowanie sterujące z interfejsem do modelu BIM. Podczas automatycznego murowania wymagana jest obsługa pracownika do ładowania cegieł i zaprawy, usuwania nadmiaru zaprawy oraz wykończania nadproży i narożników. ABLR został zintegrowany z systemami planowania zapotrzebowania materiałowego MRP (Material Requirements Planning) i kontrolą przepływu materiałów za pomocą kodów QR, co pozwoliło zminimalizować odpady na placu budowy zgodnie z praktykami zrównoważonego budownictwa.

Zaprojektowany przez firmę Construction Robotics (CR) w USA półautomatyczny murarz SAM (Semiautomatic Mason) usprawnia pracę murarzy podczas układania cegieł [16]. SAM porusza się po szynie, dlatego w dużej mierze jest elastyczny i niezależny od nierówności na palcu budowy. Zadania murarskie SAMa są dostosowane do specyfikacji budynku i warunków pracy. Efektor SAMa jest wyposażony w laser, co umożliwia jego orientację względem ściany podczas układania cegieł i pomaga określić położenie cegły w następnym cyklu. SAM nadaje

się do murowania ścian zewnętrznych dużych budynków, takich jak duże budynki, pawilony handlowe i hale przemysłowe.

Australijska firma FBR (Fastbrick Robotics) wprowadziła zaawansowany system robotyczny Hadriana X, który jest zdolny do całkowicie niezależnego zewnętrznego murowania całych konstrukcji budowlanych z poziomu wysięgnika dźwigowego [17]. Precyzyjne rozmieszczanie cegieł jest zgodnie ze specyfikacjami modelu BIM. FBR współpracuje z M&G Investment Management w celu sfinansowania trzech nowych robotów Hadrian X® do użytku w USA. Następna generacja robotów Hadrian X® wyznaczyła niezwykle tempo murowania, przekraczając 300 bloczków cementowych na godzinę zgodnie z amerykańskim standardem zrównoważonego budownictwa LEED (Leadership in Energy and Environmental Design), co stanowi znaczący postęp w technologii zrobotyzowanego murowania. Hadrian X® wykorzystuje specjalistyczne zaprawy budowlane, co ogranicza emisję dwutlenku węgla, zwiększając jednocześnie czterokrotnie wytrzymałość i trwałość konstrukcji budowlanych.

W ramach projektu badawczego POIR.04.01.02-00-0045/18 opracowano i wykonano pierwszy w Polsce mobilny ZSM w wersji prototypowej do demonstracji procesu murowania wewnątrz budynków. ZSM powstał przy współpracy Politechniki Świętokrzyskiej w Kielcach, wiodącego badacza, z partnerem przemysłowym STRABAG Sp. z o.o. Polska, który jest światowym liderem w branży budowlanej. Innowacyjne rozwiązanie, panel sterowania i moduły hydrauliczne ZSM szczegółowo przedstawiono w publikacjach [2], [3] i [4]. Kluczowym elementem ZSM jest robot przemysłowy ABB o sześciu stopniach swobody 6 DoF (Degree of Freedom) typu IRB 4600-40, który został przystosowany do zrobotyzowanego murowania. W ramach projektu badawczego przeprowadzono symulacje murowania z użyciem aplikacji ABB RobotStudio, wykonano testy laboratoryjne oraz demonstrację murowania na placu budowy (on-site). Mobilny ZSM został stworzony do konkretnych zastosowań, przede wszystkim do murowania fasad i ścian działowych wewnątrz budynków biurowych, mieszkalnych i przemysłowych. Obecnie trwają przygotowania do wykonania wersji komercyjnej ZSM spełniającego wymagania przemysłowego murowania.

4. Ocena efektywności zrobotyzowanego murowania

Podstawowym oczekiwaniem odnośnie zrobotyzowanego murowania jest zmniejszenie udziału pracy ludzkiej, poprawa wydajności i jakości murowania, obniżenie kosztów materiałowych, ograniczenie odpadów oraz poprawa bezpieczeństwa pracy. Przeprowadzono ocenę efektywności ZSM na placu budowy podczas zrobotyzowanego murowania prostych ścian działowych i ścian z otworami wewnątrz budynku mieszkalnego. Cykl zrobotyzowanego murowania obejmuje: pobieranie pustaka z zasobnika, nakładanie zaprawa na pustak, układanie pustaka w ściennie. Widok ZSM podczas demonstracyjnego zrobotyzowanego murowania ścian działowych wewnątrz budynku mieszkalnego przedstawiono na rys. 2.



Fig. 2. Widok ZSM podczas zrobotyzowanego murowania ścian działowych

Wyniki zrobotyzowanego murowania na placu budowy przy murowaniu ścian działowych wewnątrz budynku mieszkalnego [5]:

1. Podczas demonstracyjnego zrobotyzowanego murowania stawiano ściany działowe jednowarstwowe o grubości: 8, 12, 15, 18 i 24 cm ze standardowych materiałów murarskich, takich jak: cegły, pustaki komórkowe, pustaki ceramiczne, cegły klinkierowe i bloczki betonowe.
2. Średni czas zrobotyzowanego cyklu układania pustaka komórkowego o wymiarach $24 \times 24 \times 49$ cm i masie 24,4 kg wynosił 42 s (0,7 min). Do wymurowania 1 m^2 ściany potrzeba 8 pustaków komórkowych, a czas murowania wynosi około $8 \times 0,7 = 5,6$ min (w zaokrągleniu 6 min). W takim przypadku wydajność zrobotyzowanego murowania wynosi 10 m^2 ściany na godzinę. Przy tym zużywa się 102 pustaków komórkowych. Podczas 10 godzin pracy dziennie można wymurować 100 m^2 ścian, a przy 250 dniach pracy rocznie można wymurować 25.000 m^2 ścian. Według norm TZKBNK (Tymczasowego Zakładowego Katalogu Norm Budowlano-Konserwatorskich) murarz może precyzyjnie i starannie ułożyć 8 takich samych pustaków komórkowych na godzinę. Oznacza to, że wydajność zrobotyzowanego murowania jest ponad 12 razy większa na godzinę w porównaniu do pracy murarza, którego czas pracy nie jest porównywalny z robotem.
3. Odchylenie powierzchni ściany od płaszczyzny pionowej wynosiło ± 10 mm na dwóch metrach, a odchylenie grubości ściany wynosiło ± 5 mm. Odchylenia wymiarów otworów (okien i drzwi) wynosiły $+6/-3$ mm na szerokości 1 m i $+15/-10$ mm na wysokości 1 m. Jakość zrobotyzowanego murowania jest zgodna z normami i wytycznymi ITB (Instytutu Technik Budowlanych) [6].

Przeprowadzone demonstracje murowania wykazały efektywność ZSM w zadaniach zrobotyzowanego murowania w kontekście zrównoważonego budownictwa.

5. Podsumowanie

Technologie robotyczne mają ogromny potencjał dla rozwoju sektora budowlanego w zakresie: produktywności i efektywności przez automatyzację powtarzalnych prac; bezpieczeństwa pracy przez eliminując lub ograniczenie ciężkiej i uciążliwej pracy ludzkiej; kosztów operacyjnych przez minimalizację przeróbek i błędów ludzkich; zrównoważonego rozwoju przez optymalizację technologii i zużycia materiałów, zmniejszenia odpadów, zastosowanie energooszczędnych rozwiązań.

Rozwój technologii robotycznych w budownictwie wiąże się z dużą ilością niepewności dla wszystkich zaangażowanych stron: deweloperzy muszą być pewni optymalnego procesu inwestycyjnego; wykonawcy muszą być pewni jakości technologii robotycznych; pracownicy i podwykonawcy muszą dostosować organizację pracy do systemów zrobotyzowanych; edukacja i szkolenie pracowników budowlanych wymaga dostosowania programów nauczania do technologii robotycznych, itp.

Technologie robotyczne w budownictwie wymagają opracowania standardów robotyzacji budownictwa z uwzględnieniem norm, procedur oceny i certyfikacji na poziomie międzynarodowym, które będą obejmować aspekty technologiczne i jakościowe, a także obszary pracy i usług. Upowszechnienie standardów, norm i certyfikacji technologii i usług przyczyni się do intensyfikacji transformacji nowych technologii robotycznych do budownictwa.

Literatura

- [1] R. Dindorf, P. Woś. Challenges of robotic technology in sustainable construction practice. *Sustainability* 2024, 16, 5500.
- [2] R. Dindorf, P. Woś. Innovative solution of mobile robotic unit for bricklaying automation. *J. Civ. Eng. Transp.* 2022, 4, s. 21–32.
- [3] R. Dindorf, J. Takosoglu, P. Woś. Panel operatorski do zrobotyzowanego systemu murarskiego (ZSM), 16 Krajowa Konferencja Robotyki, Trzebieiszowice, 31.08-02.09.2022. *Prace naukowe Politechniki Warszawskiej, Elektronika* 2022, 197, s. 265-274.
- [4] R. Dindorf, J. Takosoglu, P. Woś, Ł. Chłopek. Hydraulic modules of the mobile robotic bricklaying system. In *Prec. Advances in Hydraulic and Pneumatic Drives and Control* 2023. *Lecture Notes in Mechanical Engineering*; Springer: Cham, Switzerland, 2024, s. 174–183.
- [5] R. Dindorf, J. Takosoglu, P. Woś, Ł. Chłopek. *Zrobotyzowany system murarski (ZSM). Raport badawczy*. Politechnika Świętokrzyska: Kielce, 2022.
- [6] M. M. Przybylski. *Warunki techniczne wykonania i odbioru robót budowlanych. A. Roboty ziemne i konstrukcyjne. Z3. Konstrukcje murowane*. Instytut Techniki Budowlanej, Warszawa 2024.
- [7] W. Pan. *Methodological development for exploring the potential to implement on-site robotics and automation in the context of public housing construction in Hong Kong*. Ph.D. Thesis, Technische Universität München, Lehrstuhl für Baurealisierung und Baurobotik, München, Germany, 2020.

- [8] V. Usmanov, J. Illetško, R. Šulc. Digital plan of brickwork layout for robotic bricklaying technology. *Sustainability* 2021, 13, 3905.
- [9] I. Vidovszky, A. Pém, A. Analyses of automated bricklaying workflow regarding time and arrangement. *IOP Conf. Ser. Mater. Sci. Eng.* 2022, 1218, 012004.
- [10] Bricklaying Robot for the First Time in the Czech Republic and for the Construction by GEMO.
<https://www.gemo.cz/en/aktuality/bricklaying-robot-for-the-first-time-in-the-czech-republic-and-on-the-construction-by-gemo>.
- [11] Bricklaying Robot Is the Future of Masonry.
<https://www.ballast-nedam.com/news/2024/the-bricklaying-robot-is-the-future-of-masonry>
- [12] Construction Automation's Brick Laying Robot Builds House.
<https://www.yorkpress.co.uk/news/18773907.construction-automations-brick-laying-robot-builds-house>.
- [13] DOM 3E. *Wyzwania branży budowlanej w 2025 roku*.
<https://www.linkedin.com/pulse/wyzwania-bran%C5%BCy-budowlanej-w-2025-roku-dom-3e-ve22f/>.
- [14] ETH Zurich Robots Use a New Digital Construction Technique to Build Timber Structures.
<https://www.dezeen.com/2018/04/16/robotic-construction-architecture-technology-eth-zurich-switzerland-spatial-timber-assemblies>.
- [15] Inżynier budownictwa. Transformacja technologiczna w budownictwie. Perspektywy i wyzwania.
<https://inzynierbudownictwa.pl/transformacja-technologiczna-w-budownictwie-perspektywy-i-wyzwania>.
- [16] Semi-Automated Robot—Sam100.
<https://www.planswift.com/blog/semi-automated-robot-sam100>
- [17] Robotic Construction Is Here.
<https://www.fbr.com.au/view/hadrian-x>.
- [18] Robot Budimex na budowie.
<https://atbudownictwo.pl/artykul/robot-budimex-na-budowie>
- [19] Zrobotyzowany system tynkarski.
<https://polskiprzemysl.com.pl/wiadomosci-budownictwo/zrobotyzowany-system-tynkarski/>

Challenges and prospects for the development of robotic technologies in the context of sustainable construction

The article discusses the challenges and development prospects of robotic technologies in the context of sustainable construction. New digital and robotic technologies used in construction are presented. Robotic technologies in construction are characterized by the types of robots, the goals of sustainable development, the technological, economic, behavioral, and safety challenges, and various limiting factors. Robotic devices used around the world for automatic bricklaying have been presented. The results of a demonstration of robotic bricklaying on a construction site during the construction of partition walls inside a residential building using a robotic bricklaying system (ZSM) are discussed.

Wybrane zagadnienia transformacji prędkości do celów sterowania w robotyce

Przemysław Herman¹

Streszczenie

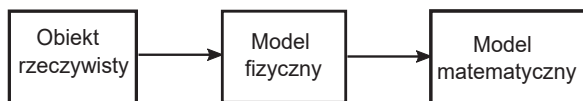
Praca zawiera krótki przegląd zastosowań quasi prędkości-inercyjnych (QPI) dla wybranych układów dynamicznych. W odróżnieniu od klasycznie rozumianych quasi-prędkości jako uogólnionych składników prędkości omawiane QPI zawierają parametry masowe i geometryczne wynikające z dekompozycji macierzy inercji. W konsekwencji stosując równania ruchu z wykorzystaniem OPI możliwe jest uzyskanie zarówno pewnego wglądu w dynamikę modelu układu jak i zrealizowanie różnych schematów sterowania. W pracy podano przykładowe równania ruchu w różnych QPI a także odpowiednie równania sterowników.

1. Wprowadzenie

1.1. Objaśnienie i cel transformacji prędkości

W technice obiekt rzeczywisty zastępowany jest jego modelami, które służą do analizy fizycznej lub matematycznej. Relacje między tymi modelami pokazano na Rys.1.

Transformację, w tym transformacja prędkości, można rozumieć jako przekształcenie opisu modelu matematycznego z jednej przestrzeni do innej. Dotyczy ona zatem modelu matematycznego. W kinematyce stosowana jest ona do zastąpienia współrzędnych kartezjańskich przez inne współrzędne, np. współrzędne biegunowe (na płaszczyźnie lub w przestrzeni) albo współrzędne walcowe.



Rys. 1. Obiekt rzeczywisty a jego modele

Przykładowe cele stosowania transformacji współrzędnych albo transformacji prędkości to:

- uzyskanie bardziej zwartej formy zapisu matematycznego,
- wygodniejsze obliczenia,
- zauważenie pewnych cech modelu niedostępnych przy opisie klasycznym,
- uzyskanie interpretacji fizycznej zmiennych,
- wykorzystanie opisu po transformacji zmiennych do celów sterowania.

Transformacja współrzędnych dotyczy zmiennych położenia, natomiast transformacja

¹Instytut Automatyki i Robotyki, Politechnika Poznańska, ul.Piotrowo 3a, 60-965 Poznań, Polska, przemyslaw.herman@put.poznan.pl.

prędkości - zastąpienia klasycznych prędkości tzw. quasi-prędkościami, czyli liniowymi kombinacjami prędkości znanych też jako składniki prędkości uogólnionych.

1.2. *Quasi-prędkości w mechanice analitycznej*

Quasi-prędkości są bardzo dobrze znane w mechanice analitycznej, przykładowo z prac [1, 10]. Dla układu mechanicznego o N stopniach swobody, którego położenie określone jest za pomocą współrzędnych uogólnionych $\theta_1, \dots, \theta_N$ ich definicję można podać następująco [1]:

$$w_j = \sum_{i=1}^N a_{ji}(\theta_1, \dots, \theta_N) \dot{\theta}_i + a_{j0}(\theta_1, \dots, \theta_N), \quad j = 1, \dots, N \quad (1)$$

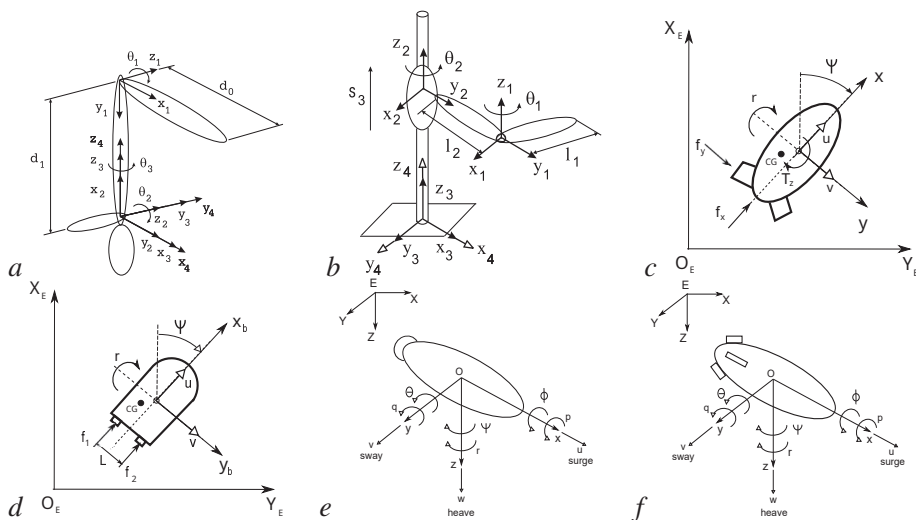
Wielkości w_j ze wzoru (1) nazywamy quasi-prędkościami. Natomiast $\dot{\theta}_1, \dots, \dot{\theta}_N$, będące pochodnymi względem czasu współrzędnych uogólnionych, nazywane są prędkościami uogólnionymi. Zmienne współczynniki $a_{ji}(\theta_1, \dots, \theta_N), a_{j0}(\theta_1, \dots, \theta_N)$ występujące we wzorze (1) są funkcjami współrzędnych uogólnionych.

1.3. *Quasi-prędkości w robotyce*

Istotną pracą, w której rozważano QPI (znormalizowane i nieznormalizowane), czyli wynikające z dekompozycji macierzy inercji, w zastosowaniu do manipulatorów jest [8]. Jej znaczącą wartością było przedstawienie przejrzystej interpretacji fizycznej wielkości występujących w równaniach ruchu. Natomiast dla układów mechanicznych (niekoniecznie manipulatorów) zaproponowano równania ruchu oparte na dekompozycji spektralnej macierzy mas (QPS) [9] oraz odsprężone równania ruchu zawierające składniki prędkości uogólnionych (SPU) [12]. Te ostatnie zostały rozszerzone o wersję znormalizowaną [2, 3]. Przegląd na temat różnych rodzajów QPI i ich analizy w odniesieniu do manipulatorów zawarto w [7] a pewnym podsumowaniem była monografia [4]. W późniejszym okresie opublikowano szereg prac, w których badano za pomocą testów symulacyjnych modele pojazdów i odpowiadające im algorytmy sterowania wyrażone w QPI, co znalazło swoje odzwierciedlenie w [5]. Było to możliwe, ponieważ okazało się, że QPI mogą być wykorzystywane w przypadku sterowań pojazdami morskimi, poduszgowcami, czy sterowcami laboratoryjnymi.

Komentarz do [8]. W tej pracy pokazano opis dynamiki w ujęciu macierzowo-wektorowym odnoszący się tylko do manipulatora. Jest to zapis w pozwalający wykonywać obliczenia rekurencyjne, co ułatwia wykonanie oprogramowania. Jednocześnie używamy wgląd w to, jak przekształcane są z jednej strony manipulatora prędkości, a z drugiej strony momenty sił w złączach. Natomiast pokazana interpretacja wielkości fizycznych jest czymś wyjątkowym, ponieważ w klasycznym opisie dynamiki manipulatora wielkości fizyczne sprzężone są w taki sposób, że niemożliwa jest ich jednoznaczne zinterpretowanie. Jednak wadą tego opisu jest generowanie równań o bardzo skomplikowanej postaci matematycznej.

Weryfikacja eksperymentalna przeprowadzana była tylko sporadycznie dla manipulatorów o dwóch stopniach swobody z pełnym wymuszeniem sygnałów sterujących, np. [2, 6, 11] oraz manipulatora typu pendubot [13].



Rys. 2. Modele układów mechanicznych w robotyce

a – manipulator typu RRR, b – manipulator typu PRR, c – pojazd w ruchu płaskim
d – pojazd w ruchu płaskim (niedosterowany), e – pojazd podwodny
f – sterowiec laboratoryjny

2. Równania ruchu układu swobodnego w quasi-prędkościach

2.1. Modele układów mechanicznych w robotyce

W robotyce układ rzeczywisty zastępowany jest jego modelem, który na początku może być reprezentowany przez odpowiedni schemat. Na Rys.2 pokazano przykładowe modele różnych układów robotycznych. Niniejsza praca dotyczy przede wszystkim zastosowania QPI dla manipulatorów a zatem równania ruchu też będą odnosiły się do tych układów.

Stosując formalizm Lagrange'a przy wyprowadzaniu równań ruchu manipulatora swobodnego otrzymamy:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = Q_i, \quad i = 1, 2, \dots, N \quad (2)$$

gdzie $L = K - V$ jest różnicą energii kinetycznej i potencjalnej zwana lagranżjanem co prowadzi do równań w postaci macierzowo-wektorowej [14, 15]:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = Q \quad (3)$$

przy czym składnik Coriolisa jest równy:

$$C(\theta, \dot{\theta}) = \dot{M}\dot{\theta} - \frac{1}{2}\dot{\theta}^T M_{\theta}\dot{\theta} \quad (4)$$

gdzie $\theta, \dot{\theta}, \ddot{\theta} \in R^N$ są wektorami współrzędnych uogólnionych oraz ich pierwszych i drugich pochodnych, $M(\theta) \in R^{N \times N}$ jest macierzą inercji manipulatora, $C(\theta, \dot{\theta}) \in R^N$ wektorem sił Coriolisa i sił odśrodkowych, $G(\theta) \in R^N$ wektorem sił grawitacyjnych, $Q \in R^N$ jest wektorem sił uogólnionych, a $\dot{\theta}^T M_\theta \dot{\theta}$ oznacza wektor kolumnowy zawierający pochodne cząstkowe macierzy $M(\theta)$. Ponadto we wzorze (3) można uwzględnić funkcję modelu tarcia wiskotycznego $f(\dot{\theta})$ uzyskując nieco bardziej ogólny model manipulatora.

Komentarz. Macierz inercji manipulatora oraz pozostałe wielkości uzyskiwane są przy zastosowaniu notacji Denavita-Hartenberga i zawierają nie tylko współrzędne uogólnione ale zależą także od mas ogniów, macierzy rotacji, tensora bezwładności oraz funnkcji jacobianowych. Dokładne postacie tych macierzy i wektorów wyprowadzone zostały np. w [14, 15].

2.2. Opis dynamiki manipulatora przy użyciu quasi-prędkości inercyjnych

Powody wprowadzenia opisu dynamiki w quasi-prędkościach inercyjnych są następujące:

- każde równanie jest odsprężone od pozostałych w sensie quasi-przyspieszeń,
- naturalne rozdzielanie równania dynamiki od równania transformacji prędkości,
- dwa równania pierwszego rzędu zamiast jednego równania drugiego rzędu,
- możliwa jest niekiedy interpretacja fizyczna,
- równania ruchu oparte na dekompozycji macierzy bezwładności pozwalają na pewien wgląd w dynamikę układu,
- macierz inercji jest diagonalna lub jednostkowa,
- stabilny wykładniczo regulator prędkości o lepszych osiąгах niż jego klasyczny odpowiednik (w sensie redukcji energii kinetycznej).

2.3. Przegląd wybranych równań ruchu manipulatora zawierających quasi-prędkości inercyjne oraz ich wykorzystanie do rozpoznania właściwości dynamicznych

Poniżej przedstawiono krótkie zestawienie równań wyrażonych za pomocą różnych wariantów QPI, które odpowiadają opisowi ze wzoru (3).

Równania ruchu w quasi-prędkościach znormalizowanych (QPZ), przy czym $M(\theta) = m(\theta)m^T(\theta)$ [4, 8]:

$$\dot{v} + C_v(\theta, v) + G_v(\theta) = \varepsilon \quad (5)$$

$$v = m^T \dot{\theta}, \quad \dot{\theta} = (m^T)^{-1} v \quad (6)$$

Równania ruchu w quasi-prędkościach nieznormalizowanych (QPN), przy czym $M(\theta) = L(\theta)D(\theta)L^T(\theta)$ [4, 8]:

$$D\ddot{\xi} + C_\xi(\theta, \xi) + G_\xi(\theta) = \kappa \quad (7)$$

$$\xi = D^{-\frac{1}{2}} v = L^T \dot{\theta}, \quad \dot{\theta} = L^T \xi \quad (8)$$

Równania ruchu oparte na dekompozycji spektralnej macierzy mas (QPS), przy czym $M(\theta) = C_e^T(\theta)S^T(\theta)S(\theta)C_e(\theta)$ [4, 9]:

$$\dot{\eta} + C_\eta(\theta, \dot{\theta}, \eta) = \varepsilon \quad (9)$$

$$\eta = SC_e \dot{\theta}, \quad \dot{\theta} = C_e^T S^{-1} \eta \quad (10)$$

gdzie $\varepsilon = S^{-1}C_e(Q - G(\theta))$.

Odsprężone równania ruchu zawierające składniki prędkości uogólnionych (SPU), przy czym $M(\theta) = \Upsilon^{-T}(\theta)N(\theta)\Upsilon^{-1}(\theta)$ [4, 12]:

$$N\dot{u} + C_u(\theta, u) + G_u(\theta) = \pi \quad (11)$$

$$u = \Upsilon^{-1}\dot{\theta}, \quad \dot{\theta} = \Upsilon u \quad (12)$$

Znormalizowane składniki prędkości uogólnionych (ZSPU), przy czym $M(\theta) = \Phi^T(\theta)\Phi(\theta)$ [2, 3, 4]:

$$\dot{\vartheta} + C_{\vartheta}(\theta, \vartheta) + G_{\vartheta}(\theta) = \varpi \quad (13)$$

$$\vartheta = \Phi\dot{\theta}, \quad \dot{\theta} = \Phi^{-1}\vartheta \quad (14)$$

Uwagi krytyczne dotyczące quasi-prędkości inercyjnych:

- niecałkowalność postaci analitycznej,
- bardziej skomplikowana postać matematyczna niż równań klasycznych,
- bardzo rzadko spotykane przypadki weryfikacji eksperymentalnej,
- brak powszechnego użycia równań dynamiki.

Równania wyrażone za pomocą QPI służą m.in. do badania dynamiki manipulatora i wprowadzania wskaźników oceny dynamiki opartych na energii kinetycznej. Dlatego też można wskazać kilka obserwacji dotyczących projektowania manipulatora, a mianowicie:

- uzyskujemy informacje o zmiennych reprezentujących każde ogniwo manipulatora indywidualnie,
- w rezultacie analizy równań ruchu oraz wyników badań symulacyjnych ustalamy parametry konstrukcyjne (masowe, geometryczne itp.), które mają istotny wpływ na powstawanie sprzężeń w układzie,
- na podstawie wyników badań można zmniejszyć sprzężenia między ogniwami manipulatora (zmodyfikowany model może być wykorzystany do sterowania w klasycznych zmiennych, jak i w QPI),
- badanie równań ruchu wyrażonych w QPI jest szczególnie przydatne, gdy zmienia się jednocześnie kilka parametrów, a skutków tych zmian nie można ustalić za pomocą analizy klasycznych równań dynamiki,
- badania symulacyjne oparte na opisie dynamiki za pomocą QPI pozwalają uniknąć kosztownych eksperymentów na etapie projektowania manipulatora.

Poniżej podano schemat postępowania przy badaniu dynamiki z użyciem QPI.

- Zapis klasycznych równań ruchu w postaci macierzowej.
- Ustalenie zbioru parametrów (masy, momenty bezwładności, momenty statyczne, wymiary geometryczne) oryginalnego (pierwotnej wersji) manipulatora i podanie go w zapisie macierzowo-wektorowym.
- Dekompozycja macierzy inercji manipulatora.
- Wyznaczenie metodą symulacyjną wartości istotnych zmiennych.
- Wyznaczenie wskaźników oceny dynamiki opartych na energii kinetycznej.
- Analiza klasycznych równań ruchu, elementów macierzy i występujących w nich wektorów oraz macierzy zawierających parametry dynamiczne i geometryczne w celu ustalenia zbioru tych parametrów, który pozwoli na uzyskanie manipulatora o

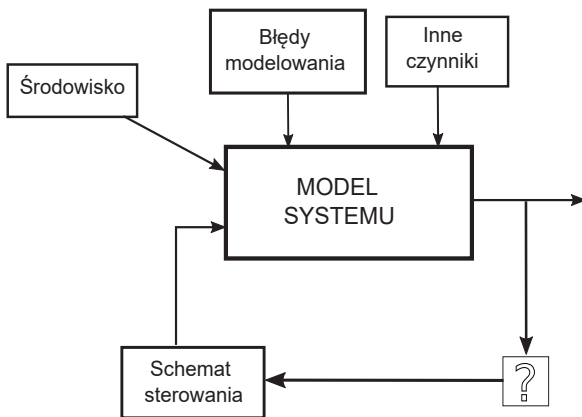
zredukowanych sprzężeniach między ogniwami. Można przy tym zmienić kolejno każdy z parametrów albo jednocześnie kilka z nich.

- Wykonanie obliczeń zgodnie z punktami od 3 do 5 i porównanie uzyskanych wyników (odpowiedzi czasowych, wskaźników) z wariantem oryginalnym w celu stwierdzenia, czy nastąpiła redukcja sprzężeń między ogniwami manipulatora.
- Powtórzenie procedury dla kilku wariantów konstrukcyjnych.
- W przypadku niejednoznacznych wyników należy zdecydować, które wskaźniki oraz redukcja których wielkości mają największe znaczenie. Oznacza to wybór wielokryterialny w celu ustalenia najlepszego wariantu konstrukcyjnego (np. redukcja sprzężeń w granicach dopuszczalnych i wartości energii kinetycznej).

3. Zastosowanie równań w quasi-prędkościach do celów sterowania

3.1. Schemat pętli zamkniętej system – sterownik zawierającej quasi-prędkości

Na Rys.3 pokazano ogólny schemat modelu układu mechanicznego wraz ze sterownikiem. Model taki powinien uwzględniać różne czynniki ale do celów symulacyjnych, w których badane są własności równań zawierających QPI mogą być one pominięte ponieważ czynniki te zniekształcają historię czasową. Natomiast gdy konieczne jest badanie zachowania układu w pętli zwrotnej w warunkach zbliżonych do rzeczywistych trzeba je oczywiście brać pod uwagę. Dla lepszej przejrzystości w dalszej części pracy nie zostały one uwzględnione.



Rys. 3. Ogólny schemat modelu system – sterownik

3.2. Przegląd wybranych sterowań zawierających quasi-prędkości inercyjne

Algorytmy klasyczne służące do realizacji różnych celów sterowania można znaleźć w wielu publikacjach z zakresu robotyki, np. [14, 15]. Poniżej podano równania sterowników zawierających QPI [4] oraz ich klasyczne odpowiedniki (KL).

Sterowanie prędkością w złączach (cel: zerowanie prędkości - zastosowanie w

astronautyce):

$$(KL) \quad Q = -k_D \dot{\theta} + G(\theta), \quad (QPI) \quad \varepsilon = -k_D v + G_v(\theta), \quad \kappa = -k_D \xi + G_\xi(\theta) \quad (15)$$

Sterowanie typu PD w przestrzeni wewnętrznej (cel: sterowanie położeniem do punktu - zastosowanie w robotyce):

$$(KL) \quad Q = -k_D \dot{\theta} + k_P \tilde{\theta} + G(\theta), \quad \text{gdzie} \quad \tilde{\theta} = \theta_d - \theta \quad (16)$$

$$(QPI) \quad \varepsilon = -k_D v + m^{-1}(\theta) k_P \tilde{\theta} + G_v(\theta), \quad \kappa = -k_D \xi + L^{-1}(\theta) k_P \tilde{\theta} + G_\xi(\theta) \quad (17)$$

Sterowanie z dynamiką odwrotną w przestrzeni wewnętrznej (cel: śledzenie prędkości przy użyciu QPI- zastosowanie w robotyce):

$$(KL) \quad Q = M(\theta)(\ddot{\theta}_d + k_D \dot{\tilde{\theta}} + k_P \tilde{\theta}) + C(\theta, \dot{\theta}) + G(\theta) \quad (18)$$

$$\text{gdzie} \quad \tilde{\theta} = \theta_d - \theta, \quad \dot{\tilde{\theta}} = \dot{\theta}_d - \dot{\theta}, \quad (19)$$

$$(QPI) \quad \varepsilon = \dot{v}_d + k_D \tilde{v} + C_v(\theta, v) + G_v(\theta), \quad \kappa = D(\dot{\xi}_d + k_D \tilde{\xi}) + C_\xi(\theta, \xi) + G_\xi(\theta) \quad (20)$$

Nieadaptacyjna wersja algorytmu Slotine'a i Liego w przestrzeni wewnętrznej (cel: śledzenie położenia - zastosowanie w robotyce):

$$(KL) \quad Q = M(\theta)\ddot{\theta}_r + C^*(\theta, \dot{\theta})\dot{\theta}_r + G(\theta) + k_D s, \quad C^*(\theta, \dot{\theta})\dot{\theta} = C(\theta, \dot{\theta}) \quad (21)$$

$$\text{gdzie} \quad s = \dot{\theta}_r - \dot{\theta} = \dot{\tilde{\theta}} + \Lambda \tilde{\theta}, \quad \ddot{\theta}_r = \ddot{\theta}_d + \Lambda \dot{\tilde{\theta}}, \quad \dot{\theta}_r = \dot{\theta}_d + \Lambda \tilde{\theta} \quad (22)$$

$$(QPI) \quad \pi = N\dot{u}_r + C_u^*(\theta, u)u_r + G_u(\theta) + k_D s_u, \quad C_u^*(\theta, u)u = C_u(\theta, u) \quad (23)$$

$$s_u = u_r - u = \Upsilon^{-1}s = \Upsilon^{-1}(\dot{\tilde{\theta}} + \Lambda \tilde{\theta}), \quad u_r = \Upsilon^{-1}\dot{\theta}_r, \quad \dot{u}_r = \Upsilon^{-1}\ddot{\theta}_r + \dot{\Upsilon}^{-1}\dot{\theta}_r \quad (24)$$

3.3. Przydatność sterowań w quasi-prędkościach inercyjnych

- Algorytmy sterowania wyrażone w QPI mogą być stosowane przy dokładnej znajomości modelu zawartego w macierzy mas.
- Uzyskujemy indywidualne sterowanie każdym ogniwnem manipulatora w sensie uwzględnienia sprzężeń dynamicznych.
- Wzmocnienie współczynników dla sterowania typu QPI daje szybszą zbieżność błędów niż w przypadku klasycznego regulatora ale konieczne jest sprawdzenie wartości dopuszczalnych sił oraz momentów sił w złączach manipulatora.
- Wyniki badań symulacyjnych mogą być wykorzystane przy projektowaniu manipulatora. W zmodyfikowanym manipulatorze klasyczny algorytm sterowania będzie działał skuteczniej dzięki zredukowaniu części składników od sprzężeń.
- Sterowanie z dynamiką odwrotną w przestrzeni wewnętrznej przy użyciu QPI jest przydatne przy projektowaniu ze względu na obecność składników od poszczególnych sił (Coriolisa, grawitacyjnych i innych) wartość tego sterowania polega na uzyskaniu wglądu w dynamikę układu podczas badań symulacyjnych (zwykle jest to rozpoznawanie sprzężeń w celu ich redukcji).
- Przydatność sterowań typu PD oraz z dynamiką odwrotną w przestrzeni zewnętrznej przy użyciu QPI jest ograniczona gdyż trudno zaobserwować ich przewagę nad algorytmami klasycznymi a dodatkowo wymagają one transformacji do przestrzeni QPI. Natomiast mogą być wykorzystane na etapie badań symulacyjnych.

3.4. Zastosowanie quasi-prędkości do sterowania pojazdami

Podstawowa różnica między równaniami ruchu dla pojazdów morskich lub sterowców a równaniami dla manipulatorów polega na tym, że elementy macierzy inercji modelu mają stałe wartości, czyli przyjmuje się macierz M zamiast $M(\theta)$. Schematy sterowania dla modeli takich pojazdów można znaleźć w pracy [5].

4. Podsumowanie i wnioski

Pomimo bogatej literatury, w której omawiane są wyniki uzyskane przy użyciu równań ruchu wyrażonych w quasi-prędkościach ineryjnych można zauważyć kilka problemów wymagających rozwiązania lub rozstrzygnięcia. Po pierwsze, najbardziej przydatne okazały się QPI w badaniu dynamiki układów mechanicznych w pętli zamkniętej zawierającej algorytmy sterowania. Po drugie, schematy sterowania uwzględniające QPI wcale nie muszą prowadzić do uzyskania lepszych osiągnięć niż algorytmy klasyczne (w sensie powszechnie stosowanych kryteriów jakości). Na początku ruchu układu powstają duże siły i momenty sił ponieważ w składnikach algorytmów typu QPI występują parametry dynamiczne modelu. Natomiast czas osiągnięcia stanu końcowego zwykle jest skrócony (z tego samego powodu). Po trzecie, weryfikacja eksperymentalna przeprowadzana jest bardzo rzadko i dla układów o dwóch stopniach swobody, co nie daje jednoznacznej odpowiedzi na pytanie o praktyczną przydatność QPI.

Z powyższych powodów należałoby zwiększyć liczbę scenariuszy w testach symulacyjnych ale może, przede wszystkim wykonać eksperymenty dla układów rzeczywistych (manipulatorów, pojazdów) o więcej niż dwóch stopniach swobody.

Literatura

- [1] R. Gutowski. *Mechanika analityczna*. Warszawa, PWN, 1971.
- [2] P. Herman. Normalised-generalised-velocity-component-based controller for a rigid serial manipulator. *IEE Proceedings - Control Theory and Applications*, 2005, 152, s.581–586.
- [3] P. Herman. On Using Generalized Velocity Components for Manipulator Dynamics and Control. *Mechanics Research Communications*, 2006, 33, s.281–291.
- [4] P. Herman. *Quasi-prędkości ineryjne oraz ich zastosowanie w badaniu dynamiki i sterowaniu manipulatorów*. Poznań, Wydawnictwo Politechniki Poznańskiej, 2006.
- [5] P. Herman. *Inertial Quasi-Velocity Based Controllers for a Class of Vehicles With Simulation Applications for Underwater Vehicles, Hovercrafts, and Indoor Airships*. Springer Tracts in Mechanical Engineering. Cham, Springer, 2022.
- [6] Herman P., K. Kozłowski. A Comparison of Certain Quasi-Velocities Approaches in PD Joint Space Control. In: 2001 IEEE International Conference on Robotics & Automation. *Proceedings*. Seoul, Korea, May 21–26, 2001, s.3819–3824. <https://ieeexplore.ieee.org/document/933213>
- [7] P. Herman, K. Kozłowski. A Survey of Equations of Motion in Terms of Inertial Quasi-Velocities for Serial Manipulators. *Archive of Applied Mechanics*, 2006, 76, s.579–614.

- [8] A. Jain, G. Rodriguez. Diagonalized Lagrangian Robot Dynamics. IEEE Transactions on Robotics and Automation, 1995, 11, s.571–584.
- [9] J.L. Junkins, H. Schaub. An Instantaneous Eigenstructure Quasivelocity Formulation for Nonlinear Multibody Dynamics. The Journal of the Astronautical Sciences, 1997, 45, s.279–295.
- [10] H.G. Kwatny, G.L. Blankenship. *Nonlinear Control and Analytical Mechanics*. Boston, Birkhäuser, 2000.
- [11] K. Kozłowski, P. Herman. Control of Robot Manipulators in Terms of Quasi-Velocities. Journal of Intelligent and Robotic Systems, 2008, 53, s.205–221.
- [12] T.A. Loduha, B. Ravani. On First-Order Decoupling of Equations of Motion for Constrained Dynamical Systems. Transactions of the ASME Journal of Applied Mechanics, 1995, 62, s.216–222.
- [13] D. Pazderski, P. Parulski, P. Bartkowiak, P. Herman. Sub-Optimal Stabilizers of the Pendubot Using Various State Space Representations. Energies, 2022, 15, 5146.
- [14] L. Sciavicco, B. Siciliano. *Modeling and Control of Robot Manipulators*. London, Springer-Verlag, 2000.
- [15] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo. *Robotics. Modeling, Planning and Control*. Springer-Verlag, London Limited, 2010.

Selected velocity transformation issues for control in robotics

The paper provides a brief overview of the inertial quasi-velocities (IQV) and their applications to selected dynamical systems. Unlike classically understood quasi-velocities as generalized velocity components, the discussed IQV include mass and geometric parameters resulting from the decomposition of the inertia matrix. Consequently, by applying the equations of motion using IQV, it is possible both to gain some insight into the dynamics of the system model and to realize various control schemes. The paper gives examples of equations of motion in various IQV and also the corresponding controller equations.

Zastosowanie manipulatora UR3 w nauczaniu podstaw teoretycznych robotyki

Krzysztof Arent¹, Katarzyna Zadarnowska¹, Jakub Kusz¹, Jakub Malewicz¹, Dawid Rogaliński²

Streszczenie

Praca jest poświęcona użyciu przemysłowego manipulatora współpracującego UR3 w nie-standardowym zastosowaniu jako narzędzia dydaktycznego wspomagającego nauczanie wybranych zagadnień z podstaw teoretycznych robotyki. Prezentacja i analiza dwóch przypadków użycia: planowanie ruchu manipulatorów oraz kalibracja kinematyki manipulatorów pokazuje, że manipulator UR3 zintegrowany programowo z ROS, Matlabem i kamerą RGB-D jest zaawansowanym i efektywnym narzędziem dydaktycznym, odpowiednim do nauczania podstaw robotyki na zaawansowanym poziomie.

1. Wstęp

Wszechstronny i dynamiczny rozwój robotyki na przestrzeni ostatnich dekad doprowadził do zastosowań robotów w licznych dziedzinach życia, w szczególności w edukacji. W pracy [9] wskazuje się na dwie zasadnicze role robotów w tym obszarze: jako narzędzi motywujących uczniów i studentów do odkrywania nauk ścisłych, technologii, inżynierii i matematyki (STEM[†]) oraz jako narzędzi dydaktycznych wspomagających nauczanie konkretnych treści programowych. Niniejszy referat nawiązuje do drugiej roli i koncentruje się na technologii wspomagającej nauczanie podstaw teoretycznych manipulatorów w zakresie analizy i weryfikacji empirycznej matematycznych algorytmów modelowania i sterowania.

W literaturze można spotkać liczne doniesienia na temat manipulatorów dydaktycznych, np. [14], [5], [10], [23], [19] (manipulatory budżetowe), [4], [8], [12] (manipulatory komercyjne, badawczo-dydaktyczne) oraz [22] [3] (przemysłowe stanowiska dydaktyczne).

W pracach [5], [10] zaproponowano manipulatory, o 4 DOF i 6 DOF odpowiednio, umożliwiające studentom poznanie wybranych zagadnień z zakresu konstrukcji, programowania i modelowania. Poszczególne ogniwa są wytworzone z tworzyw sztucznych z wykorzystaniem modelowania CAD[‡] i analizy CAE[§]. Napędy poszczególnych ogniw są oparte na serwomechanizmach, podłączonych do modułów z mi-

¹Katedra Cybernetyki i Robotyki Wydział Elektroniki, Fotoniki i Mikrosystemów, Politechnika Wrocławska, ul. Wybrzeże Wyspiańskiego 27 50-370 Wrocław krzysztof.arent@pwr.edu.pl, katarzyna.zadarnowska@pwr.edu.pl, jakub.kusz@pwr.edu.pl, jakub.malewicz@pwr.edu.pl

²Wydział Elektroniki, Fotoniki i Mikrosystemów, Politechnika Wrocławska, ul. Wybrzeże Wyspiańskiego 27 50-370 Wrocław

[†]science, technology, engineering and mathematics

[‡]Computer Aided Design (projektowanie wspomagane komputerowo)

[§]Computer Aided Engineering (inżynieria wspomagana komputerowo)

krokontrolerami (ATmega328 i Arduino MEGA 2560P odpowiednio). Pomiary kątów w poszczególnych przegubach odczytywane są przez osobny moduł z mikrokontrolerem (PIC16F877), w pierwszym przypadku, i przez Arduino MEGA 2560P, w drugim. Wszystkie moduły z mikrokontrolerami komunikują się z PC, gdzie są wyliczane pozycje i prędkości referencyjne dla serwomechanizmów w programie Matlab, przy użyciu funkcji z Robotics Toolbox [6]. Obydwie platformy były stosowane w roli narzędzi dydaktycznych podczas nauczania podstawowych zagadnień związanych ze sterowaniem kinematycznym. W pracy [14] omówiono pokrewne konstrukcje o 2 DOF. W tym przypadku manipulator fizyczny miał swój wirtualny odpowiednik w środowisku V-REP, natomiast wszystkie obliczenia w zakresie sterowania kinematycznego były przeprowadzane w Matlabie. W pracy [23] przedstawiono zbliżoną konstrukcję pod względem elektromechanicznym do wcześniej wymienionych, o 6 DOF, ale nowocześniejszą z punktu widzenia zastosowanego oprogramowania. Na PC jest zainstalowane środowisko ROS. Planowanie ruchu i sterowanie kinematyczne odbywa się na bazie frameworku MoveIt. W środowisku RViz obecna jest wirtualna wersja manipulatora, która pozwala na ograniczoną formę symulacji. Ta propozycja jest bardziej ukierunkowana na edukację w zakresie programowania komponentów układu sterowania manipulatorów w językach C/C++, Python, niż na zagadnienia teoretyczne. Z kolei, w pracy [19] zaproponowano niewielki manipulator o 5 DOF, którego rozwój był mocno oparty o rozwiązania *open hardware* i *open source*. W skład systemu sensorycznego wchodzi kamera internetowa USB. Komputerem głównym jest Raspberry Pi 3 Model B, na którym jest zainstalowane środowisko ROS. Z uwagi na ograniczone zasoby obliczeniowe opracowano dedykowany stos oprogramowania bazujący na HTTP API dla podstawowych poleceń manipulatora, co pozwala na przeniesienie złożonych obliczeń na inne komputery. Obsługa generowania żądań HTTP jest powszechna w wielu językach programowania, w szczególności w Pythonie, Matlabie i Octave. Autorzy [19] używali zaproponowany przez siebie manipulator na zajęciach z robotyki, programowania i wizji komputerowej.

Powyżej wymienione manipulatory edukacyjne reprezentują całą klasę podobnych, oferowanych przez liczne sklepy internetowe. Na pewno są wartościowym wsparciem dydaktycznym w zakresie podstaw konstrukcji, programowania i wizji komputerowej. Niemniej, to są bardzo uproszczone urządzenia w porównaniu do manipulatorów komercyjnych i dlatego, z perspektywy kształcenia na poziomie zaawansowanym, są niewystarczające. Z punktu widzenia podstaw teoretycznych robotyki, implementacja i analiza algorytmów planowania ruchu i sterowania kinematycznego będzie trudna. Niedoskonałości konstrukcji sprawiają, że wyniki badań empirycznych dla poszczególnych algorytmów będą obciążone błędem, którego źródła niekoniecznie leżą w samych algorytmach i ich implementacji. O ile nabycie takich manipulatorów nie jest kosztowne, to ich wdrożenie i utrzymanie w gotowości do użycia przez większą liczbę studentów zwykle wymaga sporych nakładów czasu i pracy.

Alternatywą dla tanich rozwiązań mogą być badawczo-dydaktyczne platformy komercyjne, takie jak Kinova Gen 3 [4], Franka-Research 3 [8] czy QArm [12]. Zarówno Kinova Gen 3 jak i Franka-Research 3 są manipulatorami redundantnymi o 7 DOF. Konstrukcja układu elektromechanicznego w każdym przypadku jest sztywna i trwała. Zestaw sensorów obejmuje czujniki położenia, prądu, siły oraz kamerę RGB-D. W przypadku Kinova Gen 3 interfejs użytkownika Kinova Kortex API jest dobrze

rozbudowany, udokumentowany i umożliwia implementację własnych algorytmów sterowania we wszystkich warstwach architektury układu sterowania, w C/C++ lub Pythonie. W przypadku Franki-Research 3 interfejs użytkownika składa się z trzech warstw, gdzie najniższa, Franka control interface (FCI), może być postrzegana jako odpowiednik Kinova Kortex API. Zarówno Kinova Kortex API jak FCI są kompatybilne z Matlabem/Simulinkiem oraz ROS/ROS2. Kinova Gen 3 i Franka Research 3 mają swoje wirtualne wersje w Gazebo i Robotics System Toolbox w Matlabie. QArm jest nieco tańszym rozwiązaniem. Producent deklaruje, że zaproponowany dla tego robota program nauczania jest zgodny z popularnymi podręcznikami robotyki autorstwa Marka Sponga i Johna Craiga. Niemniej, jest to robot o 4 DOF, co ogranicza obszar zastosowań dydaktycznych.

Z perspektywy nauczania w zakresie podstaw teoretycznych robotyki badawczo-dydaktyczne platformy komercyjne takie jak Kinova Gen 3 czy Franka-Research 3 są niezwykle atrakcyjne. W przypadku takich zadań jak badanie algorytmów planowania ruchu czy sterowanie kinematyczne możliwa jest ich realizacja zarówno z poziomu API tych robotów, Matlaba/Simulinka czy środowiska MoveIt w ROS. Szczególne znaczenie ma interfejs w Matlabie/Simulinku, gdyż to środowisko jest powszechnie stosowane do badań symulacyjnych algorytmów sterowania, uprzednio rozwiniętych w toku badań teoretycznych. Z drugiej strony koszty nabycia takich robotów są na tyle wysokie, że ich szersze zastosowanie w procesie dydaktycznym w kraju, na dzień dzisiejszy, nie ma miejsca.

Na wyposażeniu dydaktycznych laboratoriów robotycznych na uczelniach znajdują się roboty przemysłowe, w tym roboty współpracujące. Ich producenci często oferują szkolenia w zakresie różnych form eksploatacji, w tym programowania operacji wykonywanych przez manipulatory [22], [3]. Niektórzy producenci wspierają zastosowania nieprzemysłowe swoich produktów. Przykładowo firma Universal Robots wspiera i podtrzymuje rozwój interfejsu do ROS [18] dla manipulatorów UR. Dzięki temu możliwe jest wykorzystanie manipulatorów współpracujących typu UR3, w podobny sposób jak manipulatory badawcze Kinova Gen 3 czy Franka-Research 3 (do pewnego stopnia). W szczególności, możliwa jest integracja programowa manipulatorów UR z kamerą RGB-D, planowanie ruchu i sterowanie kinematyczne z poziomu MoveIt. Co więcej, możliwe jest planowanie ruchu i sterowanie kinematyczne na bazie Robotics System Toolbox w Matlabie dzięki Robotics System Toolbox Support Package for Universal Robots UR Series Manipulators [17]. Wirtualna wersja UR3 jest obecna w Robotics System Toolbox i może być wykorzystana tak jak jest to proponowane w [11].

Na dwóch przypadkach użycia zostanie pokazane, że powyżej nakreślona platforma sprzętowo-programowa może się stać bardzo dobrym i efektywnym narzędziem dydaktycznym w zakresie nauczania zagadnień teoretycznych robotyki na zaawansowanym poziomie. Dzięki niej studenci uzyskują możliwość nabycia wiedzy nie tylko o tym jak działają poszczególne funkcjonalności manipulatora URx ale również dlaczego tak działają. W pierwszej kolejności, w rozdziale 3, omawiany jest proces badań empirycznych algorytmów planowania ruchu. Następnie, w rozdziale 4, przedstawiona jest procedura kalibracji kinematyki manipulatora. Prezentacja i analiza przypadków użycia są poprzedzone szerszym omówieniem platformy sprzętowo-programowej, opartej na UR3 i jej komponentów w rozdziale 2.

2. Platforma sprzętowo-programowa stanowiska edukacyjnego z manipulatorem UR3

W skład platformy sprzętowej stanowiska dydaktycznego wchodzi następujące komponenty:

- manipulator UR3 ze sterownikiem CB3, chwytakiem dwupalczastym Robotiq 2F-85, kamerą nadgarstkową Robotiq i czujnikiem siły Robotiq FT300,
- komputer PC,
- Kamera RGB-D Intel RealSense D435.

Kamera jest podłączona do komputera PC kablem USB podczas gdy komputer PC komunikuje się ze sterownikiem CB3 za pośrednictwem TCP/IP.

Kluczowe komponenty programowe zainstalowane na komputerze PC są następujące:

- Ubuntu 20.04,
- ROS Noetic:
 - Universal Robots ROS Driver [18],
 - MoveIt Motion Planning Framework,
- Matlab 2023b:
 - Robotics System Toolbox (RST),
 - Robotics System Toolbox Support Package for Universal Robots UR Series Manipulators (RSTSPURM) [17],
 - ROS Toolbox.

Manipulator UR3 jest przemysłowym robotem współpracującym, dlatego bardzo dobrze nadaje się do zadań, które dopuszczają obecność człowieka w przestrzeni zadaniowej, co jest istotne z punktu widzenia stanowiska dydaktycznego. Matlab jest uznanym środowiskiem do obliczeń naukowych i inżynierskich. RST i RSTSPURM udostępniają w Matlabie reprezentacje wirtualnego i rzeczywistego manipulatora UR3 w formie obiektów. Podobne skrypty Matlabu mogą być uruchomione zarówno dla wirtualnego jak i fizycznego manipulatora UR3. W konsekwencji, student ma możliwość przejścia ścieżki od analizy formalnej, przez numeryczną do eksperymentalnej, koncentrując swoją uwagę na problemie teoretycznym, co jest istotne na wczesnym etapie kształcenia z podstaw robotyki. UR ROS Driver wraz z RSTSPURM tworzą łącznie programowe pomiędzy Matlabem i sterownikiem CB3 manipulatora UR3.

Kamera nadgarstkowa Robotiq w manipulatorze UR3 jest niewystarczająca z punktu widzenia potrzeb kształcenia z podstaw teoretycznych robotyki w zakresie systemów wizyjnych robotów i wizyjnego sprzężenia zwrotnego. Zdecydowanie lepszym rozwiązaniem jest kamera RGB-D Intel RealSense D435. W podobne kamery są wyposażane manipulatory badawczo-dydaktyczne, np. Kinova Gen 3 czy Franka-Research 3. Powyżej wspomniana kamera nie jest obsługiwana przez Matlab 2023b w wersji pod Ubuntu 20.04, ale producent oferuje do niej API w C++ i Pythonie oraz wrapper w ROS. To, w połączeniu z UR ROS Driver i MoveIt, daje studentowi możliwość analizy numerycznej i eksperymentalnej rozmaitych zagadnień związanych z wizją w robotyce i wizyjnym sprzężeniem zwrotnym [6], [15]. Tym bardziej, że na etapie analizy numerycznej można wykorzystać wirtualną wersję UR3, dostępną z poziomu ROS w Gazebo.

Reasumując, jeżeli dane laboratorium posiada manipulator współpracujący UR3, zakupiony z myślą o kształceniu w zakresie zastosowań przemysłowych współcze-

snych robotów oraz środowisko do obliczeń naukowych i inżynierskich Matlab, powszechnie wykorzystywane na uczelniach, to niewielkim kosztem może wytworzyć platformę sprzętowo-programową o zupełnie nowych możliwościach, porównywalnych z tym co oferują drogie manipulatory badawczo-dydaktyczne.

3. Planowanie ruchu manipulatora

Planowanie ruchu należy do grupy ważniejszych zagadnień omawianych na kursach z podstaw teoretycznych robotyki. Zasadniczo proces planowania składa się z dwóch etapów [15], [7]:

1. wyznaczenia toru ruchu w przestrzeni zadaniowej pomiędzy pozycją bieżącą a docelową, z uwzględnieniem obecności przeszkód, i stowarzyszonego z nim toru ruchu w przestrzeni konfiguracyjnej;
2. wyznaczeniu trajektorii w przestrzeni fazowej, generującej zadany tor i spełniającej ograniczenia w zakresie prędkości i przyspieszeń.

Wyznaczenie analityczne toru ruchu i stowarzyszonej z nim trajektorii w przestrzeni fazowej jest możliwe wyłącznie dla prostych manipulatorów (np. planarny o 2 DOF). W przypadku manipulatorów typowych, o 6 DOF i więcej, analiza własności algorytmów planowania ruchu może mieć charakter wyłącznie numeryczny i eksperymentalny. Tylko w procesie badań eksperymentalnych student ma możliwość nabycia wiedzy i intuicji w zakresie zachowania się postury manipulatora w konfiguracjach oddalonych i bliskich osobliwym oraz wpływu kinematyki na możliwości robota w zakresie manipulacji.

W pracy [13] zostały przeprowadzone badania kilku metod planowania ruchu manipulatora na platformie sprzętowo-programowej przedstawionej w rozdziale 2. Do tego celu wybrano trzy metody planowania toru ruchu: jakobianową, R. H. Taylora i grafową (Rapidly-exploring Random Tree). Wyznaczanie trajektorii wzdłuż toru nie miało miejsca ponieważ RSTSPURM dopuszcza jedynie zadawanie punktów pośrednich (waypoints) w przestrzeni przegubowej lub przestrzeni roboczej.

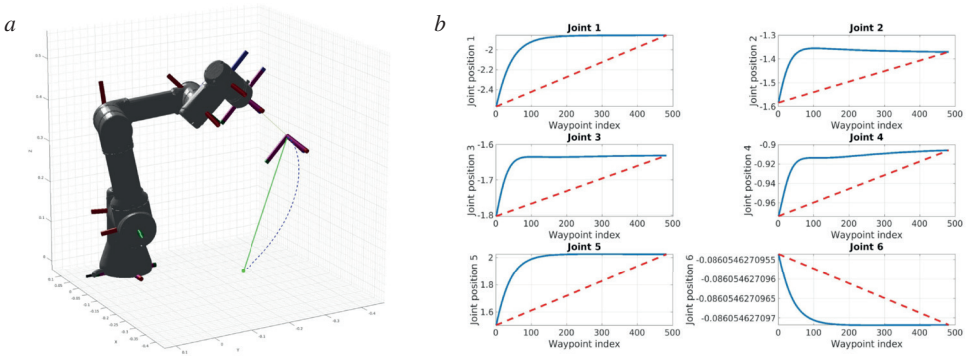
Z [13] wyłania się model pracy złożony z dwóch etapów: badania numeryczne i badania eksperymentalne.

3.1. Badania numeryczne

Na tym etapie ma miejsce rozwój, implementacja i testowanie algorytmów planowania toru ruchu na wirtualnym manipulatorze UR3. Implementacja ma formę m-plików, w których kod bazuje na operacjach macierzowych i funkcjach z RST, takich jak:

- `UR3virtual=loadrobot(...)` - obiekt reprezentujący wirtualny manipulator UR3,
- `UR3virtual.getTransform(...)` - kinematyka prosta UR3virtual,
- `UR3virtual.geometricJacobian(...)` - jakobian geometryczny UR3virtual
- `UR3virtual.show(...)` - wizualizacja wirtualnego UR3, UR3virtual, dla zadanej konfiguracji,
- `ik=inverseKinematics(UR3virtual, ...)` - zwraca obiekt z solverem kinematyki odwrotnej stowarzyszony z obiektem UR3virtual.

Ponieważ to są funkcje wysokiego poziomu, implementacje algorytmów planowania toru ruchu mają zwięzłą i czytelną formę, pomimo iż szczegółowa postać równań reprezentujących model manipulatora UR3 jest złożona. Student może skoncentrować się na implementowanym algorytmie bez nadmiernego angażowania się w kwestie



Rys. 1. Metoda Jakobianu transponowanego

a – tor ruchu w przestrzeni zadaniowej, animacja, b – tory ruchu w przestrzeni przegubowej [13]

rozwoju i uruchomienia oprogramowania.

Efektem pierwszego etapu są tory ruchu w przestrzeni przegubowej i przestrzeni zadaniowej oraz animacje postury, co ilustruje rysunek 1.

Rysunek 1 przedstawia wyniki badania, którego celem było zaplanowanie toru ruchu na bazie metody Jakobianu transponowanego od $x_0 = [-0.4 \ -0.1 \ -0.1]^T$ [m] do $x_k = [-0.1 \ -0.4 \ -0.3]^T$ [m]. Na tor ruchu składają się 482 konfiguracje. Wykresy pokazują, że cel został osiągnięty. Ponadto widać też, że w przypadku badanej metody uzyskany tor ruchu niekoniecznie ma kształt linii prostej. To spostrzeżenie ilustruje własność użytego algorytmu, którego pierwotnym zadaniem jest wyznaczenie trajektorii od pozycji początkowej do końcowej na skończonym przedziale czasu.

3.2. Badania eksperymentalne

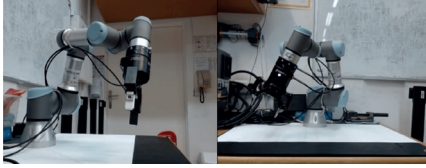
W tej fazie ma miejsce badanie zachowania rzeczywistego manipulatora UR3 poruszającego się wzdłuż torów ruchu uzyskanych w etapie I-wszym, omówionym w rozdziale 3.1. Implementacja procedury eksperymentu ma formę m-pliku, wykorzystującego kod z etapu I-wszego oraz funkcje z RSTSPURM, takie jak

- `UR3real=universalrobot(...)` lub `urROSnode(...)` – zwraca obiekt Matlaba reprezentujący rzeczywisty UR3,
- `UR3real.getJointConfiguration` – aktualna konfiguracja robota,
- `UR3real.sendJointConfiguration, UR3real.followTrajectory` – zadaje punkt, sekwencję punktów w przestrzeni przegubowej robotowi,
- `UR3real.sendCartesianPose, UR3real.followWaypoints` – zadaje punkt orientacyjny, sekwencję punktów orientacyjnych w przestrzeni zadaniowej robotowi.

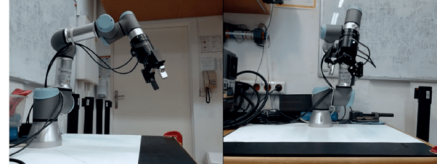
Taki m-plik jest nieduży gdyż używane w nim wyrażenia Matlaba mają charakter języka wysokiego poziomu. Podobnie jak w etapie I-wszym, student może się skoncentrować na eksperymencie, a nie na jego przygotowaniu.

Efektem jest ruch manipulatora i zarejestrowane trajektorie ruchu w przestrzeniach przegubowej i zadaniowej, co ilustruje rysunek 2. Warto podkreślić, że student ma możliwość obserwacji zachowania robota współpracującego, o 6 DOF, stosowanego w przemyśle, realizującego tory ruchu uzyskane w efekcie wdrożenia różnych algo-

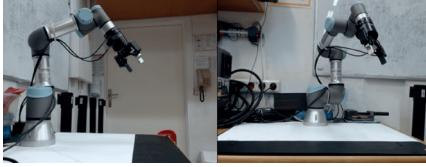
a



b



c



Rys. 2. Metoda Jakobianu transponowanego
a – stan początkowy, b – stan pośredni; a – stan końcowy [13]

Tab. 1. Nominalne parametry Denavita-Hartenberga manipulatora UR3

i	θ_i^0 [rad]	d_i [m]	a_i [m]	α_i [rad]
1	0	0.1519	0.0	$\pi/2$
2	0	0.0	-0.24365	0.0
3	0	0.0	-0.21325	0.0
4	0	0.11235	0.0	$\pi/2$
5	0	0.08535	0.0	$-\pi/2$
6	0	0.0819	0.0	0.0

rytmów teoretycznych, w dość krótkim czasie i dość niskim kosztem.

Powyżej przedstawiona procedura może znaleźć zastosowanie w procesie przekazywania studentom przez autorów tej pracy własnych wyników naukowych, o charakterze teoretycznym, np. z artykułów [16], [21], [20].

4. Kalibracja kinematyki manipulatora

Kalibracja kinematyki wymaga dogłębnego zrozumienia zagadnień z modelowania kinematyki manipulatora i umożliwia całościowe spojrzenie na tę tematykę. Z tego powodu walor edukacyjny zadania kalibracji kinematyki manipulatora jest znaczący.

Rozważmy manipulator o 6 DOF i jego kinematykę

$$y = f(q, \Phi), \quad (1)$$

gdzie f jest odwzorowaniem z rozmaitości przegubowej w rozmaitość zadaniową, q oznacza współrzędne wewnętrzne, y pozycję efektora a Φ wektor złożony z parametrów kinematyki (najczęściej w notacji Denavita-Hartenberga). Zwykle jest znana nominalna wartość tych parametrów Φ_{nom} , która nieco odbiega od faktycznych na skutek niedoskonałości w technologii wytwarzania. Przykładowo, nominalne parametry Denavita-Hartenberga dla manipulatora UR3 są zebrane w tabeli 1. Do oceny

estymat $\hat{\Phi}$ parametrów kinematyki stosuje się model błędu

$$\Delta y(\hat{\Phi}, q) := y - f(q, \hat{\Phi}). \quad (2)$$

W przypadku parametrów nominalnych $\Delta y(\Phi_{nom}, q) \neq 0$. Zadanie kalibracji kinematyki polega na wyznaczeniu wektora wartości $\hat{\Phi}^*$, dla którego $\Delta y(\Phi^*, q) = 0$ dla wszystkich par (y, q) dopuszczalnych przez manipulator [15]. W rzeczywistości należy wyznaczyć $\hat{\Phi}^*$, który minimalizuje wskaźnik jakości oparty na (2) (zwykle jest to forma kwadratowa względem $\hat{\Phi}$) dla sekwencji pomiarów (y_l, q_l) , $l = 1 \dots N$. Procedura kalibracji składa się z czterech głównych etapów [1]:

Modelowanie: wybór parametryzacji poszczególnych ogniw manipulatora, wyznaczenie kinematyki, jacobianu identyfikacyjnego i inwersu uogólnionego dla jacobianu identyfikacyjnego.

Akwizycja danych: wyznaczenie optymalnych konfiguracji przegubowych; przemieszczenie manipulatora do optymalnej konfiguracji przegubowej, pomiar rzeczywistych wartości konfiguracji przegubowych i pozycji efektora (zob. rys. 3).

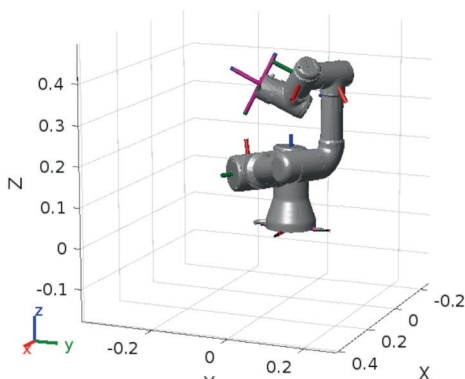
Identyfikacja: wyliczenie estymat parametrów kinematyki przez algorytm kalibracji.

Weryfikacja: analiza wartości błędów pozycji efektora dla wyliczonych estymat w oparciu o posiadane dane pomiarowe.

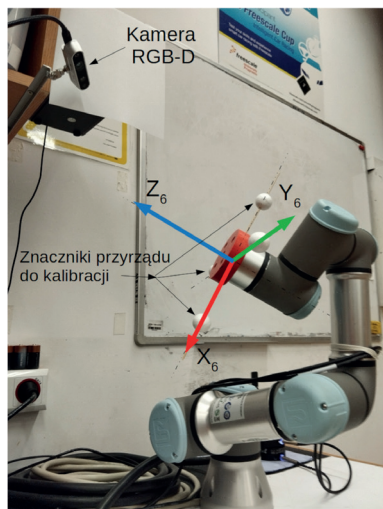
W przypadku drugiego etapu pewnym wyzwaniem jest pomiar y dla ustalonego q . O ile pomiar q jest prosty i może mieć formę odczytu aktualnych wartości zwracanych przez wbudowane enkodery w manipulatorze to pomiar y wymaga użycia specjalizowanych zewnętrznych systemów CMM (coordinate measuring machine). Z uwagi na wysoką cenę CCM, w pracy została dokonana adaptacja metody pomiarowej z [1], którą ilustruje rysunek 3.

W miejsce efektora, na końcu ogniwa nr 6, tam gdzie jest zlokalizowany układ współrzędnych $X_6Y_6Z_6$, zgodnie z konwencją Denavita-Hartenberga, jest zamontowany przyrząd w formie symetrycznego krzyża z czterema kolorowanymi znacznikami (zob. rys. 3(b)). Sposób montażu i zastosowana technologia pozwalają na precyzyjne wyznaczenie położenia tych znaczników w układzie współrzędnych efektora. Z drugiej strony, dzięki kamerze RGB-D, w połączeniu z interfejsem programowym *pyrealsense2* i biblioteką *OpenCV*, możliwe jest wyznaczenie położenia znaczników na przyrządzie w układzie współrzędnych kamery. Te dane umożliwiają wyznaczenie macierzy obrotów ${}^{RGB-D}R_6$ i wektora położenia ${}^{RGB-D}p_6$ dla układu $X_6Y_6Z_6$ w układzie współrzędnych kamery RGB-D. Macierz przejścia ${}^bT_{RGB-D}$ z układu współrzędnych kamery do bazowego układu współrzędnych jest znana, w konsekwencji możliwe jest wyliczenie macierzy obrotów bR_6 i wektora położenia bp_6 dla układu $X_6Y_6Z_6$ w bazowym układzie współrzędnych. Powyższe obliczenia są realizowane w ROS-wym węźle `\ee_pos` i publikowane w topiku o tej samej nazwie. Powyższa metoda pomiarowa nie jest precyzyjna, ale ma walor dydaktyczny przybliżając studentowi problem pomiaru pozycji efektora i jego znaczenie dla problemu kalibracji kinematyki manipulatora.

a



b



Rys. 3. Kalibracje kinematyki UR3, akwizycja danych

a – manipulator wirtualny, b – manipulator rzeczywisty z przyrządem do pomiaru pozycji $X_6Y_6Z_6$ zamontowanym w miejscu efektora

4.1. Badania numeryczne

Na tym etapie można zrelizować pełną procedurę kalibracji kinematyki UR3 na gruncie danych (y, q) uzyskanych z obiektu UR3virtual (por. rozdz. 3.1). **Etap modelowania** obejmuje: wyznaczenie kinematyki manipulatora na gruncie notacji Denavita-Hartenberga z modyfikacją Hayati'ego, $k(q, \Phi_H)$, modelu błędu, jacobianu kinematycznego $J(\hat{\Phi}_H, q) := \frac{\partial k}{\partial \Phi_H}(\hat{\Phi}_H, q)$. **Etap akwizycji danych** obejmuje wyznaczenie N optymalnych postur manipulatora UR3 z uwagi na indeks obserwowalności jacobianu kinematycznego i zebranie danych (y_l, q_l) , $l = 1 \dots N$ (rys. 3(a)). Tak zebrane dane są wykorzystane w **etapie identyfikacji** do wyznaczenia $\hat{\Phi}_H^*$ minimalizującego wskaźnik jakości oparty na modelu błędu $\Delta y(\hat{\Phi}_H)$. Minimalizacja odbywa się na gruncie algorytmu Gaussa-Newtona, zainicjowanego w Φ_{nom} , którego implementacja w Matlabie ma postać funkcji `lsqnonlin`. Alternatywnie, można użyć funkcji `fmincon`. Przykładowe wyniki są zebrane w tabeli. 2. Ostatni, **etap weryfikacji**, to wyznaczenie wartości $\Delta y(\hat{\Phi}_H^*, q)$ dla różnych q i ich ocena.

Podobnie jak w rozdziale 3.1 wdrożenie powyższych zadań ma formę LiveScriptu lub kilku m-plików niedużych rozmiarów dzięki wykorzystaniu funkcji Matlabu wysokiego poziomu. Dzięki temu student ma możliwość dość szybkiego opanowania całej procedury kalibracji i zbadania jej niuansów, pomimo złożoności samej procedury.

W pracy [1] na temat kalibracji kinematyki manipulatora ARIA o 6 DOF zaproponowaną procedurę kalibracji oprogramowano w formie Kinematics Calibration Tool-box pod Matlabą, o nieco innych i nieco bardziej rozbudowanych możliwościach.

Tab. 2. Wartości parametrów Denavita-Hartenberga-Hayatiego uzyskane z procesu kalibracji kinematyki dla wirtualnego manipulatora UR3

i	θ_i^0 [rad]	d_i [m]	a_i [m]	α_i [rad]	β_i [rad]
1	3.5114e-02	1.5018e-01	4.8255e-03	1.6379e+00	
2	7.1614e-02		-2.4142e-01	-3.9670e-02	-1.2095e-08
3	4.5762e-02		-2.1072e-01	-1.1098e-02	3.9726e-09
4	-4.1459e-02	1.1726e-01	1.5152e-03	1.6173e+00	
5	-7.8982e-02	8.4003e-02	-4.2731e-03	-1.5747e+00	
6	4.1204e-02	7.9370e-02	1.3163e-03	-4.5767e-02	

Tab. 3. Wartości parametrów Denavita-Hartenberga-Hayatiego uzyskane z procesu kalibracji kinematyki dla rzeczywistego manipulatora UR3

i	θ_i^0 [rad]	d_i [m]	a_i [m]	α_i [rad]	β_i [rad]
1	-0.29015	0.31456	0.033395	1.1108	
2	-0.34163		-0.01722	-0.22819	-0.14973
3	-0.39019		0.062349	0.36874	0.2289
4	-0.17341	0.072066	0.044173	1.5597	
5	0.2866	0.1237	-0.18683	-1.3828	
6	0.79507	0.05636	0.019897	-0.14973	

4.2. Badania eksperymentalne

W tej fazie realizowana jest procedura kalibracji kinematyki z rozdz. 4.1, przy czym w **etapie akwizycji danych** wartości (y, q) pochodzą z pomiaru wielkości fizycznych (rys. 3(b)). Dzięki badaniom na manipulatorze UR3 wirtualnym student może wytypować zadawane konfiguracje q_i , $i = 1 \dots, N$ dla rzeczywistego UR3 oraz liczbę pomiarów N . Z poziomu m-pliku (lub LiveScriptu) w Matlabie można odczytać konfiguracje q_i z manipulatora wywołaniem `UR3real.getJointConfiguration` (por. rozdz. 3.1), natomiast stowarzyszoną z q_i pozycję układu $X_6 Y_6 Z_6$ w bazowym układzie współrzędnych można odczytać z topika `\ee_pos` przy użyciu funkcji z ROS Toolbox w Matlabie. **Etapy identyfikacji i weryfikacji** będą identyczne jak w fazie badań numerycznych na wirtualnym UR3 w rozdziale 4.1. Przykładowe wyniki można zobaczyć w tabeli 3 (uzyskane wyniki są obarczone błędem wynikającym z niedokładności metody pomiaru, o czym wcześniej wspomniano, niemniej prace nad rozwiązaniem tego problemu są w toku).

W efekcie student ma możliwość względnie szybkiego poznania problemu, algorytmów, procedury i środowiska pomiarowego do kalibracji kinematyki manipulatora przemysłowego o 6 DOF.

5. Podsumowanie

W pracy przedstawiono propozycję stanowiska dydaktycznego wspomagającego nauczanie wybranych zagadnień podstaw teoretycznych robotyki. Stanowisko jest owocem integracji sprzętu i oprogramowania, które jest obecne w wielu krajowych laboratoriach robotycznych: manipulatora UR3, środowiska ROS i programu Matlab. Bez ponoszenia wysokich kosztów rozszerzono spektrum możliwości posiadanych zasobów laboratoryjnych o nowe obszary, które dają możliwość prowadzenia badań

eksperymentalnych algorytmów omawianych na kursach z podstaw teoretycznych robotyki na manipulatorze o 6 DOF.

Prezentowane w pracy środowisko eksperymentalne pozwala na szybkie przejście od analizy formalnej do badania eksperymentalnego bez konieczności wdrażania się w obsługę i eksploatację nowych narzędzi programowych i sprzętowych. Warto też zwrócić uwagę na fakt, że firma Universal Robots nieustannie wspiera rozwój UR ROS Driver, UR3 jest obecny w RST kolejnych wydaniach Matlaba. Można też zauważyć rozwój RSTSPURM z wydania na wydanie. To skłania do konkluzji, że zaproponowane w pracy rozwiązanie jest perspektywiczne w średnim horyzoncie czasowym.

Przypadki użycia, omawiane w pracy, dotyczą kinematyki manipulatora. Równie ważnym obszarem podstaw robotyki jest sterowanie dynamiką manipulatora oparte na jej modelu. Na chwilę obecną nie ma dobrych propozycji robotów wspomagających edukację w tym obszarze. Pewne możliwości oferują kosztowne roboty badawczo-dydaktyczne, jak Franka-Research 3 [8], choć w literaturze są prace prezentujące działania własne w tym zakresie, na przykład w [2] (gdzie ważnym elementem elementem platformy sprzętowo-programowej jest Simulink).

Stanowisko dydaktyczne, omawiane w pracy, jest elastyczne i może być wykorzystane na różne sposoby, w formie i zakresie które są dopasowane do potrzeb studentów, metod dydaktycznych prowadzącego, oraz spodziewanych efektów uczenia.

Literatura

- [1] Krzysztof Arent et al. Kinematic calibration of a lightweight manipulator for medical applications. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 2024, wolumen 72, numer 6, s. e151381.
- [2] Krzysztof Arent et al. Construction, Modelling and Identification of an Experimental and Educational 2DOF Manipulator for Investigation of Model Based Control Algorithms. In: 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR). *Proceedings*, 2018, s. 920–925.
- [3] ASTOR. *Robot ASTORINO*., 2025.
- [4] Alexandre Campeau-Lecours et al. Kinova Modular Robot Arms for Service Robotics Applications. *International Journal of Robotics Applications and Technologies*, 01, 2017, wolumen 5, s. 49–71.
- [5] José Alberto Naves Cocota, Hideo Silva Fujita, Itamar Jorge da Silva. A Low-Cost Robot Manipulator for Education. In: 2012 Technologies Applied to Electronics Teaching (TAE). *Proceedings*, 2012, s. 164–169.
- [6] Peter Corke. *Robotics, Vision and Control*. Springer Cham 2017.
- [7] Ignacy Dulęba. *Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych*. Akademicka Oficyna Wydawnicza EXIT 2001.
- [8] Sami Haddadin et al. The Franka Emika Robot: A Reference Platform for Robotics Research and Education. *IEEE Robotics & Automation Magazine*, 2022, wolumen 29, numer 2, s. 46–64.
- [9] David P. Miller, Illah R. Nourbakhsh, Roland Siegwart. *Robotics for Education*. Cham, Springer Berlin Heidelberg 2016, s. 2115–2134.
- [10] José Alberto Naves Cocota et al. Design and implementation of an educational robot manipulator. In: 2014 XI Tecnologías Aplicadas a la Enseñanza de la Electro-

- nica (Technologies Applied to Electronics Teaching) (TAEE). *Proceedings*, 2014, s. 1–6.
- [11] Norbert Prokopiuk, Piotr Falkowski. Applicability of Augmented and Virtual Reality for Education in Robotics. *Journal of Automation, Mobile Robotics and Intelligent Systems*, Aug., 2023, wolumen 16, numer 3, s. 65–74.
 - [12] Quanser. *QArm.*, 2025.
 - [13] Dawid Rogaliński. Experimental verification of motion planning algorithms on the ur3 manipulator with the application of matlab/robotics system toolbox. *Praca magisterska*, Wrocław University of Science and Technology, 2024.
 - [14] Alexander S. Barbosa et al. Design and Development of a Manipulator for Educational Robotics. In: 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). *Proceedings*, 2018, s. 554–561.
 - [15] Bruno Siciliano et al. *Robotics. Modelling, Planning and Control*. Springer-Verlag London 2009.
 - [16] Krzysztof Tchoń, Katarzyna Zadarnowska. Kinematic dexterity of mobile manipulators: an endogenous configuration space approach. *Robotica*, 2003, wolumen 21, numer 5, s. 521–530.
 - [17] Universal Robots. *Robotics System Toolbox Support Package for Universal Robots UR Series Manipulators.*, 2025.
 - [18] Universal Robots. *Universal_Robots_ROS_Driver.*, 2025.
 - [19] Dhaval R. Vyas, Anilkumar Markana, Nitin Padhiyar. Economic 6-dof robotic manipulator hardware design for research and education. *Materials Today: Proceedings*, 2022, wolumen 62, s. 7179–7184. International Conference on Additive Manufacturing and Advanced Materials (AM2).
 - [20] Katarzyna Zadarnowska, Adam G Ratajczak. *Robot motion and control*. Springer 2011, wolumen 422, rozdział Task-priority motion planning of wheeled mobile robots subject to slipping, s. 75–85. *Lecture Notes in Control and Information Sciences*, ISSN 0170-8643.
 - [21] Katarzyna Zadarnowska, Krzysztof Tchoń. A control theory framework for performance evaluation of mobile manipulators. *Robotica*, 2007.
 - [22] *Standardowy pakiet edukacyjny FANUC.*, 2025.
 - [23] Luka Čehovin Zajc, Anže Rezelj, Danijel Skočaj. Low-Cost Open-Source Robotic Platform for Education. *IEEE Transactions on Learning Technologies*, 2023, wolumen 16, numer 1, s. 18–25.

The use of an UR3 manipulator for teaching theoretical foundations of robotics

The paper is devoted to the use of the UR3 industrial collaborative manipulator in a non-standard application as a teaching tool to support the teaching of selected topics from the theoretical stances of robotics. The presentation and analysis of two use cases: motion planning of manipulators and calibration of manipulator kinematics show that the UR3 manipulator integrated by software with ROS, Matlab and RGB-D camera is a professional and effective teaching tool suitable for teaching robotics at an advanced level.

Przegląd bezzałogowych platform w zastosowaniach wojskowych

Jakub Malewicz¹, Krzysztof Górski²

Streszczenie

Niniejsza praca stanowi przegląd platform autonomicznych, wykorzystywanych przez siły zbrojne w różnych częściach świata. W artykule przedstawiono platformy lądowe, pływające i powietrzne. W zestawieniu uwzględniono kategoryzację platform pod względem przeznaczenia wojskowego. Nakreślono możliwości poszczególnych jednostek, ujętych w dokumencie, wskazano ich parametry techniczne, warunki pracy i główne obszary zastosowań.

1. Wprowadzenie

Historia konfliktów zbrojnych jest nierozzerwalnie związana z rozwojem technologii. Poziom technologiczny stron konfliktu nie jest czynnikiem decydującym i rozstrzygającym. Na wynik mają wpływ także poziom wykształcenia, morale, liczebność i sposób dowodzenia [1], [9]. Niezaprzeczalnie jednak przewaga na polu technicznym jest jednym z bardzo istotnych parametrów, które mogą przesądzić o wyniku konfliktu. Z uwagi na ten fakt armie świata przeznaczają coraz większe kwoty na rzecz rozwoju technologii wojskowych.

W zakresie technologicznym rynek cywilny i wojskowy od zawsze były ze sobą związane. Do II wojny światowej wojsko bardzo często korzystało z cywilnych wynalazków. Przykładami są telegraf, telefon, radio, kolej, samolot. Druga wojna światowa zmieniła kierunek transferu technologii, to znaczy sprzęt i wynalazki wytworzone na potrzeby armii trafiały później na rynek cywilny. Przykładami mogą tu być kuchenka mikrofalowa, klej cyjanoakrylowy czy nawigacja satelitarna.

Rozwój platform bezzałogowych jest silnie związany z rozwojem platform załogowych, operujących w danym środowisku. Jest także powiązany z poziomem technologicznym systemów sensoryki, mocy obliczeniowej komputerów, ich miniaturyzacji, ale także stopniem zaawansowania algorytmów i intensywnego, szczególnie w ostatnim czasie rozwoju sztucznej inteligencji.

Pierwsze platformy bezzałogowe na dużą skalę były stosowane już podczas II wojny światowej. Można tu wymienić na przykład zdalnie sterowanego Goliatha³

¹Akademia Wojsk Lądowych imienia Tadeusza Kościuszki, ul. Piotra Czajkowskiego 109, 51-147 Wrocław, jakub.malewicz@awl.edu.pl, awl.edu.pl

²Akademia Wojsk Lądowych imienia Tadeusza Kościuszki, ul. Piotra Czajkowskiego 109, 51-147 Wrocław, krzysztof.gorski@awl.edu.pl, awl.edu.pl

lub Borgwarda B IV⁴. Wyżej wymienione platformy były pozbawione jakiegokolwiek autonomii. Współczesne, dynamiczne pole walki stawia przed sprzętem znacznie większe wymagania. Ze względu na wielość odbieranych sygnałów i monitorowanych parametrów aparatura pokładowa musi przejść od operatora percepcję i analizę danych. Niektóre zdarzenia są na tyle dynamiczne, że człowiek nie jest w stanie na nie zareagować w odpowiednio krótkim czasie.

2. Klasyfikacja

Autonomiczne platformy wojskowe w nomenklaturze militarnej określane jako bezzałogowe systemy lub bezzałogowe statki mogą być wykorzystywane na lądzie. Wówczas mamy do czynienia z BSL (bezzałogowy system lądowy). Działają w powietrzu i wtedy jest to BSP (bezzałogowy system powietrzny). W środowisku wodnym występują BSM (bezzałogowe systemy morskie), niezależnie czy operują na wodach otwartych czy śródlądowych. Jedną z klasyfikacji jest zatem identyczna, jak klasyfikacja cywilnych robotów mobilnych, wskazana między innymi w [2] i podkategorii zdefiniowane w [3].

Nieco zmieniona jest nomenklatura autonomicznych platform w przypadku kategoryzacji pod kątem zastosowania. Wówczas, w przypadku jednostek wojskowych, mówi się o wsparciu, które w znakomitej większości sprowadza się do platform transportowych. Drugą kategorią to rozpoznanie, odpowiadające cywilnym robotom inspekcyjnym. Pozostałe typy zostałyby w sferze cywilnej umieszczone w grupie robotów specjalnych, a są to urządzenia bojowe oraz platformy zwalczające inne systemy bezzałogowe.

Istnieje szereg innych sposobów podziału wojskowych systemów bezzałogowych w zależności od środowiska działania czy przeznaczenia. Można tu wymienić na przykład klasyfikację stosowaną przez NATO dla BSP [4], [6], [7], wiążącą masę i długość lotu, umieszczoną w tabeli 1 lub inne stosowane na przykład w pracy [5].

Tab. 1. Klasyfikacja uproszczona BSP według NATO z 2009 roku

Klasa	Masa	Czas lotu	Zastosowanie
Klasa I	Do 150kg	Do 6 godzin	Wsparcie na poziomie taktycznym, na szczeblu drużyny, plutonu, kompanii
Klasa II	150kg – 600kg	Do 24 godzin	Wsparcie na poziomie taktycznym, na poziomie batalionu, brygady
Klasa III	Powyżej 600kg	Do 40 godzin	Wsparcie na poziomie operacyjnym i strategicznym

³ Mały pojazd o napędzie gąsienicowym, wypełniony materiałem wybuchowym, przeznaczony do likwidacji zatorów drogowych, bunkrów, zdalnie sterowany poprzez ciągnięty za sobą kabel.

⁴ Ciężki pojazd gąsienicowy, przeznaczony do przenoszenia i posadowienia ładunków wybuchowych.

3. Przykłady platform bezzałogowych

Jak nakreślono w rozdziale 2, autonomiczne platformy wojskowe, mogą być klasyfikowane według różnych kryteriów. W niniejszym rozdziale zostaną podane przykłady systemów operujących w środowiskach lądowym, morskim i powietrznym, z uwzględnieniem różnego ich przeznaczenia.

3.1. Bezzałogowe systemy lądowe

Wśród systemów działających na lądzie można wyróżnić platformy wsparcia. Jednym z bardziej znanych rozwiązań tego typu jest BigDog, zbudowany przez Boston Dynamics. Na rysunku 1 przedstawiono wyżej wymieniony system, a w tabeli 2 umieszczono podstawowe parametry jednostki. W założeniu, ten kroczący robot ma operować w trudnym terenie i stanowić wsparcie na poziomie drużyny lub plutonu, przenosząc zaopatrzenie, w szczególności żywność i amunicję.



Rys. 1. Robot BigDog operujący w terenie leśnym

Tab. 2. Podstawowe parametry platformy BigDog

Parametr	Wartość
Kraj pochodzenia	Stany Zjednoczone
Producent	Boston Dynamics
Długość	91cm
Wysokość	76cm
Masa	109kg
Prędkość	6,4km/h
Masa ładunku	154kg
Zasięg	20,6kg
Pokonywanie wzniesień	Do 35 stopni

Innym przykładem platformy lądowej jest SUGV. Jego nazwa to akronim: Small Unmanned Ground Vehicle. Jest to pojazd opracowany przez amerykańską firmę iRobot, znaną z cywilnego rynku odkurzaczy autonomicznych. Jego niewielka masa

(13kg) pozwala na przenoszenie przez jednego żołnierza. Przeznaczeniem urządzenia jest rozpoznanie, ale też rozminowywanie i neutralizacja prowizorycznych ładunków wybuchowych. Dodatkowo może być wykorzystany do oznaczania celów przy pomocy lasera pokładowego. Na rysunku 2 przedstawiono jedną z wersji wykonania tego robota. Cechą wyróżniającą tę platformę jest zdolność do wytrzymywania przeciążeń rzędu 400g (dla porównania piloci myśliwców w trakcie walki powietrznej są w stanie wytrzymać przeciążenia dochodzące nawet do 20g).



Rys. 2. Platforma SUGV z systemem wizyjnym i laserem

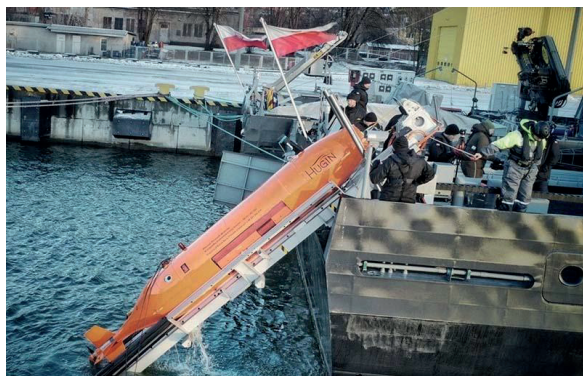
Wśród wojskowych platform specjalnych można wyróżnić tak zwaną amunicję krążącą. Jej przykładem jest platforma GNOM (rys. 3), opracowana przez polską firmę MACRO-SYSTEM. Uzbrojony w ładunek o masie 7 kilogramów i rozpędzający się do prędkości 80 kilometrów na godzinę robot jest w stanie bardzo szybko zneutralizować pojazdy przeciwnika.



Rys. 3. Amunicja krążąca GNOM

3.2. Bezzałogowe systemy morskie

Systemy bezzałogowe, działające w środowisku wodnym, operują zarówno na powierzchni, jak i pod nią. Modele realizujące misje rozpoznawcze mogą monitorować infrastrukturę krytyczną, wejścia do portów, wykrywać miny lub wrogie jednostki. Przykładem takiego systemu jest Hugin (rys. 4, tab. 3). Wersje bojowe są przeznaczone do niszczenia celów, głównie jednostek pływających, ale są także egzemplarze wyposażone w wyrzutnie rakiet, pozwalające niszczyć cele w powietrzu. Przykładem takiej jednostki jest Magura V5 (rys. 5, tab. 4), wykorzystywana przez wojska ukraińskie.



Rys. 4. Jednostka podwodna Hugin

Tab. 3. Parametry aparatu Hugin

Parametr	Wartość
Kraj pochodzenia	Norwegia
Producent	Kongsberg
Długość	5,2m
Średnica	0,75m
Masa	1000 - 1500kg
Prędkość	6węzłów
Głębokość	1000m
Czas pracy	30h



Rys. 5. Jednostka nawodna Magura V5

Tab. 4. Parametry aparatu Magura V5

Parametr	Wartość
Kraj pochodzenia	Ukraina
Producent	SpetsTechnoExport
Długość	5,5m
Średnica	0,75m
Wyporność	1000kg
Prędkość	42węzły
Zasięg	800km
Czas pracy	60h

3.3. Bezzałogowe systemy powietrzne

Najbardziej rozpoznawalną i najbardziej liczną kategorią systemów bezzałogowych są platformy powietrzne. Wynika to z faktu ich dużej użyteczności, rozległego obszaru działania, znacznie mniejszych ograniczeń, ale i mniejszej liczby zagrożeń, niż te, które czekają na aparaty w środowisku morskim i na lądzie. Ponadto jednostki powietrzne mogą operować nad lądem i wodą, dzięki czemu są bardziej uniwersalne, zdecydowanie trudniej dostępne dla żołnierzy przeciwnika, a ze względu na pułap operowania mają zdecydowanie najlepsze pole widzenia.

Podobnie, jak w przypadku rozwoju lotnictwa, tak i w przypadku aparatów powietrznych pierwsze konstrukcje były stosowane w celach rozpoznawczych. Można tu wymienić od największych: Global Hawk (rys. 7, tab. 5), poprzez średnie, FlyEye (rys. 8, tab. 6), aż po najmniejsze Black Hornet Nano (rys. 9, tab. 7).

**Rys. 7.** Bezzałogowy system rozpoznania Global Hawk

Tab. 5. Parametry aparatu Global Hawk

Parametr	Wartość
Kraj pochodzenia	Stany Zjednoczone
Producent	Northrop Grumman
Rozpiętość	35,5m
Długość	13,4m
Wysokość	4,6m
Masa własna	3850kg
Masa użyteczna	10400kg
Prędkość przelotowa	650km/h
Pułap	20km
Zasięg	20000km
Długość lotu	36h



Rys. 8. Bezzałogowy system rozpoznania FlyEye

Tab. 6. Parametry aparatu FlyEye

Parametr	Wartość
Kraj pochodzenia	Polska
Producent	WB Electronics
Rozpiętość	3,6m
Długość	1,8m
Masa startowa	12kg
Prędkość przelotowa	120km/h
Pułap	5000m
Długość lotu	2 – 4h



Rys. 9. Bezzałogowy system rozpoznania Black Hornet Nano

Tab. 7. Parametry aparatu Black Hornet Nano

Parametr	Wartość
Kraj pochodzenia	Wielka Brytania, Norwegia
Producent	Prox Dynamics, Marlborough Communications
Średnica wirnika	120mm
Masa własna	16g
Pułap	10 – 30m
Zasięg	1000m
Długość lotu	30min.

W dalszej kolejności wykształciły się drony bojowe takie, jak turecki Bayraktar TB2 (rys. 10, tab. 8), którego poprzednikiem był aparat zbudowany w celach rozpoznawczych. Osobną kategorię stanowią jednostki, będące amunicją krążącą, takie jak Shahed 136 (rys. 11, tab. 9), które są urządzeniami jednokrotnego użytku.



Rys. 10. Bezzałogowa jednostka bojowa Bayraktar TB2

Tab. 8. Parametry aparatu Bayraktar TB2

Parametr	Wartość
Kraj pochodzenia	Turcja
Producent	Baykar Makina
Masa startowa	650kg
Prędkość	220km/h
Pułap	8200m
Zasięg	150km
Długość lotu	27h



Rys. 11. Amunicja krążąca Shahed 136

Tab. 9. Parametry aparatu Shahed 136

Parametr	Wartość
Kraj pochodzenia	Iran
Producent	Shahed Aviation Industries
Długość	3,5m
Rozpiętość	2,5m
Masa startowa	200kg
Prędkość	185km/h
Zasięg	1800 – 2500km

Tematyka systemów powietrznych jest niezwykle szeroka i ze względu na obszerność tematyki może stanowić podstawę do osobnego opracowania, jak na przykład [8].

4. Podsumowanie

W niniejszej pracy zaprezentowano klasyfikację systemów bezałogowych stosowanych w siłach zbrojnych. Przedstawiono kilka przykładów poszczególnych typów aparatów wraz z ich parametrami. Należy wyraźnie podkreślić, że jest to tylko niewielki fragment istniejących na świecie rozwiązań i niniejsza praca nie wyczerpuje tematu. W opracowaniu starano się zobrazować rozległość tej tematyki i wskazano obszary działań i kierunki rozwoju.

Przedstawione w dokumencie rozwiązania, nie tylko przeszły chrzest bojowy, ale także udowodniły swoją wartość na współczesnym polu walki, zarówno w zakończonych konfliktach i operacjach wojskowych, jak i trwających wojnach.

Z całą pewnością należy stwierdzić, że systemy bezzałogowe stanowią przyszłość wyposażenia nowoczesnych armii, a ich liczba, różnorodność oraz obszar zastosowań będzie tylko wzrastał.

LITERATURA

- [1] S. Tzu. *Sztuka Wojny*. Warszawa, Biblioteka Filozofów 2009, tom 13.
- [2] J. Szrek. *Inspekcyjne roboty mobilne*, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej 2023, s. 12-14.
- [3] R. Siegwart, I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*, London, The MIT Press Cambridge 2004.
- [4] ATP-3.3.7. *Guidance for the training of unmanned aircraft systems (UAS) operators*, Edition B, Version 1, NATO Standardization Agency, 2014, s. 1-4.
- [5] S. Agostino, M. Mammone, M. Nelson, T. Zhou, *Classification of Unmanned Aerial Vehicles Aeronautical Engineering*, Adelaide, The University of Adelaide 2007, Mechanical Engineering 3016.
- [6] D. Kompała, *Systemy bezzałogowych statków powietrznych jako główny środek rozpoznania powietrznego przyszłości*, Warszawa, Zeszyty Naukowe Wydziału Zarządzania i Dowodzenia Akademii Sztuki Wojennej 2015, 2(14)/2015.
- [7] K. Balcer, *Klasyfikacja bezzałogowych statków powietrznych w odniesieniu do specyfikacji oraz przepisów prawa polskiego i międzynarodowego*, Bydgoszcz, Wydawnictwa Uczelniane Politechniki Bydgoskiej im. Jana i Jędrzeja Śniadeckich 2021, Developments in Mechanical Engineering s. 5-15.
- [8] M. J. Dougherty, *DRONES. An Illustrated Guide to the Unmanned Aircraft That Filling Our Skies*, New York, Sterling Publishing 2015.
- [9] Wojna a technologie,
<https://bezpiecznik.org/2018/04/28/wojna-a-technologie/>

Overview of Unmanned Platforms in Military Applications

This paper is a review of autonomous platforms used by armed forces in various parts of the world. The paper presents land, maritime and air platforms. The list includes a categorization of platforms in terms of military purpose. The capabilities of individual units included in the document are outlined, their technical parameters, operating conditions and main areas of application are indicated.

ROBOTY MEDYCZNE I REHABILITACYJNE

Doświadczalna weryfikacja dokładności procedury pomiaru LLD i LOD za pomocą ramienia pomiarowego NaviFast6D

*Agnieszka Kobierska¹, Patryk Skwierczyński¹, Kacper Sęk¹,
Piotr Kacprzak¹, Leszek Podśędkowski¹*

Streszczenie

Praca prezentuje opis weryfikacji dokładności procedury pomiaru zmiany LLD (Leg Length Discrepancy) i LOD (Leg Offset Discrepancy) za pomocą ramienia pomiarowego NaviFast6D opracowanego na PŁ. Parametry te docelowo będą mierzone w trakcie operacji endoprotezoplastyki stawu biodrowego. Pokazano zbudowane stanowisko testowe i przeprowadzono testy dokładności pomiaru w warunkach laboratoryjnych zbliżonych do warunków na sali operacyjnej. Pomiar podzielono na etapy zgodne z procedurą operacji chirurgicznej. Wyznaczenie dokładności pomiaru ramieniem NaviFast6D jest istotne ze względu na określenie, jak bardzo wynik pomiaru może odbiegać do rzeczywistej wartości mierzonej wielkości. Dzięki wykorzystaniu ramienia lekarze będą w stanie zmniejszyć liczbę potrzebnych reoperacji wynikających z wydłużenia nogi pacjenta po operacji stawu biodrowego.

1. Wstęp

Wraz ze wzrostem liczby pacjentów z problemami stawu biodrowego, operacje jego wymiany stają się coraz powszechniejsze. Towarzyszy temu rozwój technologii chirurgicznych, zwłaszcza robotyki, która znacząco poprawia precyzję zabiegów ortopedycznych. Wysoka dokładność narzędzi robotycznych ma kluczowe znaczenie dla bezpieczeństwa pacjentów i skuteczności operacji. Celem niniejszego artykułu jest przedstawienie procedury badania dokładności i powtarzalności pomiaru przyrostowego za pomocą nowo opracowanego w Instytucie Obrabiarek i TBM Politechniki Łódzkiej ramienia pomiarowego NaviFast6D, w warunkach laboratoryjnych zbliżonych do warunków panujących podczas operacji. Powtarzalność określana jest na podstawie odchylenia standardowego a dokładność na podstawie średniej wartości uzyskanego błędu.

1.1. Analiza literatury

Wymiana stawu biodrowego jest powszechnie stosowanym zabiegiem w celu umożliwienia poruszania się i złagodzeniu bólu pacjenta. Kluczowym parametrem oceny sukcesu operacji jest różnica w długościach nóg. Badania wskazują, że nawet niewielkie różnice w długościach nóg mogą powodować dyskomfort [2] [3] [4].

¹ Politechnika Łódzka, Instytut Obrabiarek i TBM

Obecnie nie ma naukowego konsensusu w kwestii definicji istotnej nierówności pooperacyjnej. Niektórzy badacze akceptują różnice rzędu 2 cm inni mniejsze [5]. Jednym z głównych celów obecnie jest zwiększenie dokładności wykonywania operacji wymiany stawu biodrowego i w tym celu opracowano urządzenia wspomagające chirurgów w czasie operacji. Obiecującą technologią w tej dziedzinie są roboty chirurgiczne [6], ale pomimo zadowalających wyników badań dokładności [7] [8] tych urządzeń, są one obecnie drogie i mało dostępne. Inne metody opierają się na wspomaganiu chirurga poprzez wskazanie jak powinna być umieszczona proteza, dostępnymi na rynku urządzeniami tego typu są OrthoPilot oraz OrthAlign.

OrthoPilot wykorzystuje dane z dwóch kamer, interpretowane przez algorytmy i wyświetlane na ekranie. Do działania wymaga nadajnika umieszczonego w miednicy pacjenta, który umożliwia pomiar offsetu, długości nogi i kąta nachylenia panewki. Dokładność systemu to 0,2 mm [9].

W warunkach laboratoryjnych przebadano różnice we wskazaniach położenia punktów w przestrzeni pomiędzy OrthoPilotem i systemu wykorzystującego 12 kamer firmy VICON z oprogramowaniem Vicon Nexus version 1.4.116 with Bodybuilder software version 3.55 [10]. Zbieżność pomiędzy OrthoPilot i VICON została zmierzona przy użyciu skalibrowanego modelu fantomu miednicy z naniesionymi landmarkami. Wyniki tych pomiarów wskazują, że błąd odchylenia względem drugiego urządzenia wynosił około 1%. Kolejne badanie sprawdzające dokładność systemu OrthoPilot przeprowadzono przy użyciu stołów mikrometrycznych przesuwających się względem siebie [11]. Metoda wykorzystana w tym badaniu symulowała zmiany w długości nogi i offsetu pacjenta. Dokładność w przypadku długości wynosiła 0,6 mm, a w przypadku umiejscowienia panewki błąd wyniósł 0,66 °.

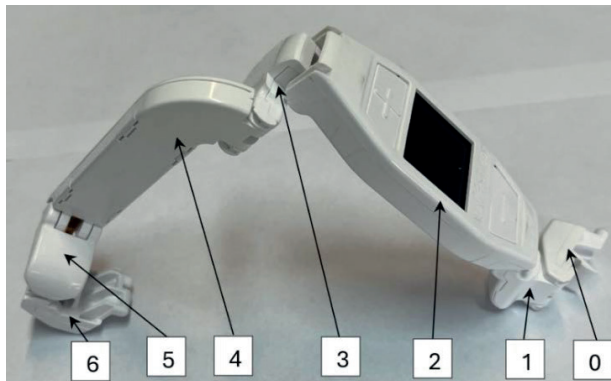
OrthAlign działa w oparciu o czujnik laserowy. Jak podaje producent, przy pomocy tego urządzenia uzyskano 95% skuteczność w umieszczeniu panewki biodra w docelowym obszarze oraz uzyskano zmniejszenie różnicy w długościach nóg po operacji [12].

Innym sposobem nawigacji, przedstawionym w badaniu „A Hybrid CT-Free Navigation System for Total Hip Arthroplasty” [13] obejmującym przezskórną digitalizację punktową i dwupłaszczyznową rekonstrukcję punktów orientacyjnych przy użyciu wielu zarejestrowanych obrazów fluoroskopowych. Wykazano znaczną poprawę dokładności wykonanych operacji. Dla urządzenia OrthoPilot i OrthAlign przeprowadzono dodatkowo metaanalizy badań klinicznych. W przypadku systemu OrthoPilot udowodniono, że jego zastosowanie skutkuje zmniejszeniem prawdopodobieństwa wystąpienia błędu medycznego [15] [16]. Dla urządzenia OrthAlign dokładność wynosi 3° w procesie umieszczania panewki [17].

W dziedzinie robotyki przemysłowej i ramion przegubowych, do określania położenia i orientacji końcówki sondy stosuje się powszechnie notację Denavita-Hartenberga (D-H). Uwzględnia ona zarówno geometrię, jak i relacje kątowe między przegubami [1]. Jest szczególnie przydatna w przypadku ramion pomiarowych lub robotów z przegubową konstrukcją. W kontekście robotów asystujących przy wymianie stawu biodrowego, takich jak OrthoAlign [12], również stosuje się notację

D-H. Innymi rozwiązaniami w tej dziedzinie są m.in. OrthoPilot [9], oparte na systemie wizyjnym.

Alternatywne podejście do określania położenia operowanego biodra reprezentuje ramię pomiarowe NaviFast 6D opracowane w Instytucie Obrabiarek i TBM Politechniki Łódzkiej. Urządzenie NaviFast 6D (rys. 1) to ramię pomiarowe o krótkim zasięgu, opracowane z myślą o chirurgach, służące do pomiaru położenia kości udowej względem kości biodrowej podczas operacji endoprotezoplastyki.



Rys. 1 Ramię pomiarowe NaviFast 6D

NaviFast 6D w obecnej wersji składa się z 7 segmentów (rys. 1) połączonych przegubami obrotowymi. Każdy segment zawiera trójosiowy akcelerometr MEMS, który przesyła dane do modułu obliczeniowego w segmencie 2. Segmenty skrajne (0 i 6) mocowane są do kości udowej i talerza biodrowego za pomocą markerów w śrubach chirurgicznych. Kąty θ_i pomiędzy poszczególnymi segmentami wyznaczano na podstawie przekształcania zmierzonego przez akcelerometry wektora przyspieszenia grawitacyjnego E_i przez macierz rotacji R_{i-1}^i .

$$E_{i-1} = R_{i-1}^i * E_i \quad (1)$$

$$\theta_i = \text{atan2}(e_{y_{i-1}}e_{x_i} + e_{x_{i-1}}(-c\alpha_i e_{y_i} + s\alpha_i e_{z_i}), e_{x_{i-1}}e_{x_i} - e_{y_{i-1}}(-c\alpha_i e_{y_i} + s\alpha_i e_{z_i})) \quad (2)$$

gdzie e_{x_i} , e_{y_i} , e_{z_i} – przekształcony do układu $OXYZ_i$ odczyt z akcelerometrów trzech składowych wektora grawitacji, $c\alpha$, $s\alpha$ – cosinus i sinus kąta skręcenia członu. Tak wyznaczone kąty (równania 1, 2) posłużyły do wyliczania macierzy jednorodnych T_{i-1}^i a w konsekwencji macierzy T_0^6 określającej położenie i orientację końca ramienia w stosunku do jego początku. Kluczowym elementem w procesie pomiaru jest kalibracja ramienia, w tym celu zostało zaprojektowane specjalne urządzenie oparte na dwóch platformach Stewart'a [18]. Inne metody kalibracji tego typu urządzeń nie dają dostatecznie dużej dokładności, niezbędnej w przedstawionym zastosowaniu [19] [20] [21]. NaviFast 6D umożliwia pomiar zarówno pozycji liniowych, jak i kątowych we wszystkich stopniach swobody. Pomiary te są wykonywane za pomocą czujników MEMS, służących do pomiaru przyspieszeń [14]. Jak wskazały wcześniejsze badania przeprowadzone na tym urządzeniu, przy

pomocy akcelerometrów, można uzyskać powtarzalny pomiar z odchyleniem standardowym dla tego urządzenia wynoszącym poniżej 1 mm.

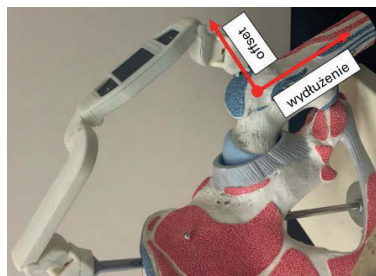
1.2. Operacja wymiany stawu biodrowego z użyciem NaviFast6D

W pierwszym kroku wymiany stawu biodrowego, za pomocą ramienia NaviFast 6D, chirurg wkłada w kości pacjenta śruby kaniulowane i mocuje do nich uchwyty pomiarowe. Następnie ustawia nogę w ustawieniu odpowiadającemu pozycji stojącej (etap I) i rejestruje dane, na podstawie których wyznacza pierwotne położenie kości udowej względem biodrowej – jako punkt odniesienia do późniejszych pomiarów.

W trakcie operacji po etapie I ramię jest demontowane z kości udowej i chirurg przeprowadza całą operację wstawienia implantu, najpierw panewki stawu biodrowego a potem próbnego implantu trzpienia i główki kości udowej. Na koniec tej procedury chirurg ponownie montuje ramię pomiarowe NaviFast 6D do pacjenta poprzez wstawienie kołków uchwytów ramienia pomiarowego w śruby kaniulowane znajdujące się w kości udowej i kości biodrowej.

W następnym kroku chirurg porusza kończyną we wszystkich stopniach swobody przez około 20 sekund (etap II). Ruchy powinny być powolne, tak aby przyspieszenia członów ramienia pomiarowego wynikające z tych ruchów były pomijalnie małe w stosunku do mierzonego przez akcelerometry przyspieszenia grawitacyjnego. Na podstawie zmierzonych położzeń i orientacji program wyznacza, jakie poprawki ma nanieść na pomiar, jeżeli orientacja końcowa (w etapie IV) nogi będzie niezgodna z początkową. W kolejnym kroku chirurg odłącza ostatni segment ramienia pomiarowego NaviFast 6D od kości biodrowej i ustawia czwarty segment ramienia wzdłuż kości udowej (etap III) tak by powierzchnia tego segmentu była prostopadła do kierunku offsetu. Ramię pomiarowe przelicza orientację tylko czwartego segmentu względem podstawy związanej z kością udową i określa układ współrzędnych w którym będą podawane wyniki. To bardzo istotny punkt procedury, gdyż dopiero w tym momencie system obliczeniowy znajdujący się w ramieniu dowiadyuje się jak jest ustawiona kość udowa względem ramienia.

Następnie chirurg ustawia kość udową w położeniu odpowiadającym ustawieniu nogi podczas postawy stojącej (identycznym jak w etapie I) i dokonuje pomiaru wtórnego (etap IV).



Rys. 2 Kierunki pomiarowe - offset, wydłużenie na modelu stawu biodrowego

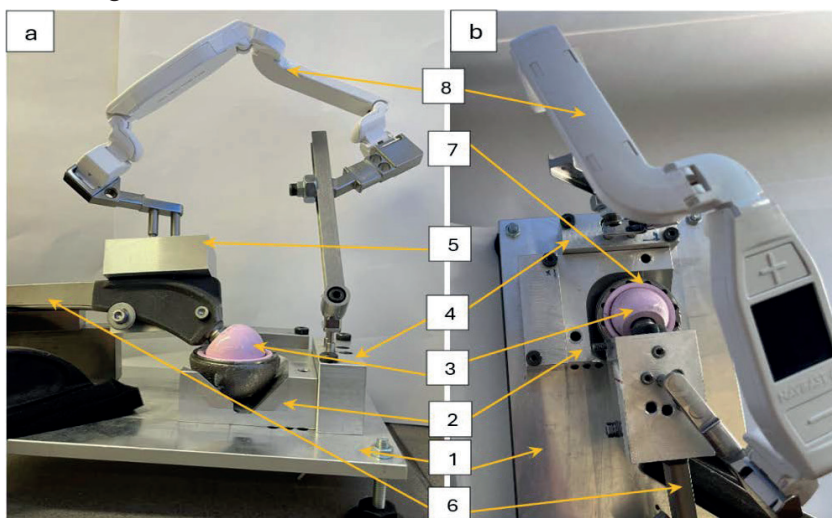
Wyliczana różnica pomiędzy uzyskanym położeniem a położeniem zmierzonym w etapie I jest rzutowana na kierunki zdefiniowane w etapie III. Z tych 3 kierunków pokazywany na ekranie jest jedynie kierunek wydłużenia i offsetu (rys.2), natomiast kierunek trzeci w przód i w tył pacjenta jest pomijany jako nieistotny w procedurze chirurgicznej.

Badanie dokładności procedury pomiaru realizowano w etapach zgodnie z procedurą wykonywaną pomiaru w trakcie operacji chirurgicznej.

2. Badanie dokładności ramienia NaviFast 6D

2.1. Stanowisko badawcze

Stanowisko badawcze (rys. 3) umożliwiło precyzyjne pomiary różnicowe oraz symulację wszystkich etapów operacji. Cała endoproteza, w tym panewka, trzon i główka, są standardowo stosowanymi elementami w procedurze wymiany stawu biodrowego.

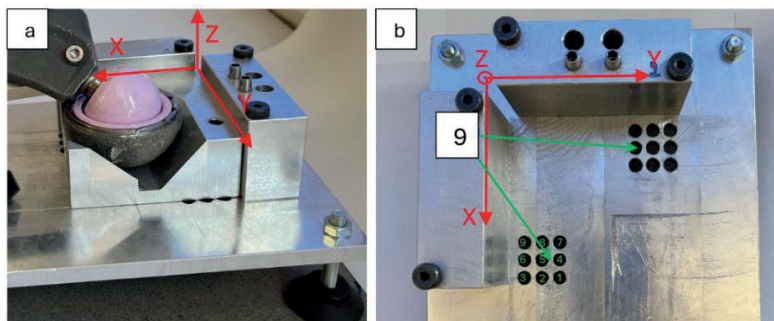


Rys. 3 Stanowisko badawcze: a- widok z przodu, b- widok z góry

W skład konstrukcji stanowiska badawczego (rys.3) wchodzi podstawa (1) będąca płytą aluminiową na której zamocowano blok oporowy (4), imitujący kości miednicy pacjenta, umożliwiające mocowanie ramienia NaviFast6D (8), koszyk (2) pozwalający na mocowanie panewki (7) oraz przemieszczanie jej pomiędzy kolejnymi pozycjami na płycie. Stanowisko posiada elementy imitujące kość udową w postaci głowy endoprotezy (3) oraz trzpienia endoprotezy (6) z korpusem (5), który pozwala na zamocowanie ramienia NaviFast6D do trzpienia endoprotezy.

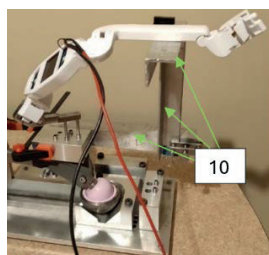
Konstrukcja stanowiska umożliwia symulowanie ruchu panewki w trzech płaszczyznach (rys.4 a). Ruch wzdłuż osi pionowej odpowiada za zmianę offsetu,

natomiast przemieszczenie wzdłuż dłuższej krawędzi stanowiska zmienia wydłużenie nogi.



Rys. 4 Stanowisko badawcze: a- płaszczyzny ruchu panewki, b- otwory mocujące koszyk

Ruch w poziomie umożliwia 9 par mocowań w postaci precyzyjnie nawierconych otworów kalibrowanych (9) w podstawie (rys.3 b). Przesunięcie pionowe zostało uzyskane przy pomocy płytki o znanej grubości umieszczonej pod koszykiem z panewką. Stanowisko uzupełniono także o dodatkowe kątowniki (10) wspomagające pomiar kierunku (rys. 5).



Rys. 5 Pomiar kierunku– etap III operacji z wykorzystaniem NaviFast 6D

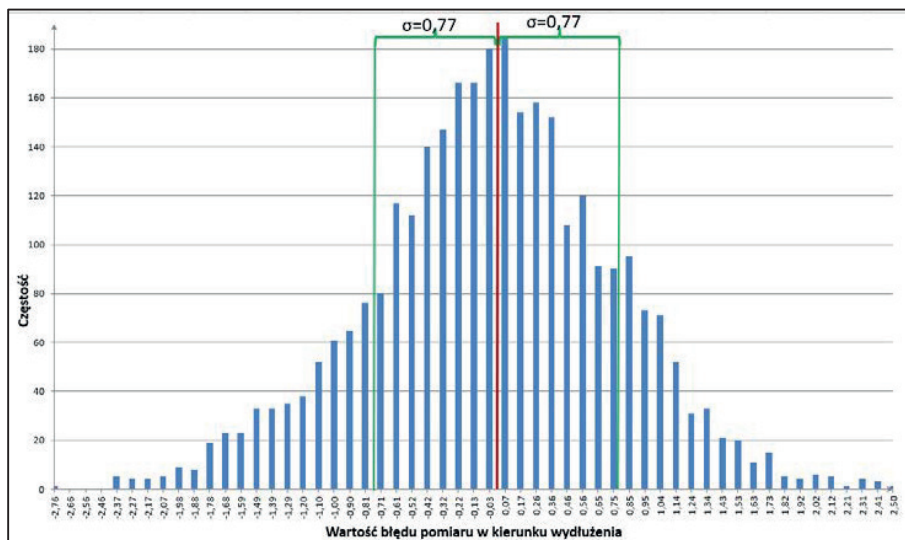
Nadmiarowe mocowania endoprotezy oraz samego ramienia NaviFast6D imitują różne rodzaje przemieszczeń, które mogą wystąpić podczas operacji, umożliwiając zbadanie ich wpływu na pomiary dokonywane przez urządzenie. W związku z procedurą pomiarową zaistniała konieczność ustalenia równoległości segmentu 5 urządzenia NaviFast6D oraz kości biodrowej (symulowanej przez szczękę mocującą trzon endoprotezy). W tym celu stanowisko uzupełniono o 3 elementy, używane w III etapie procedury pomiarowej (pomiarze kierunku). Profile kątowe (10) zostały zamocowane w trakcie pomiarów do szczęki za pomocą zacisków i ustawione prostopadłe względem siebie przy pomocy narzędzi wzorcowych – kątowników. Szczęką natomiast została spozycjonowana – również za pomocą kątowników - równolegle do podstawy stanowiska. Dzięki temu uzyskano równoległość szczęki względem podstawy oraz równoległość segmentu 5 urządzenia Navifast6D względem szczęki, a co za tym idzie również względem podstawy. Działanie to

pozwoili ustalić zgodność kierunków stanowiska i ramienia pomiarowego NaviFast 6D, tak by mierzone wartości odpowiadały zadawanym zmianom.

2.2. Metodyka badań

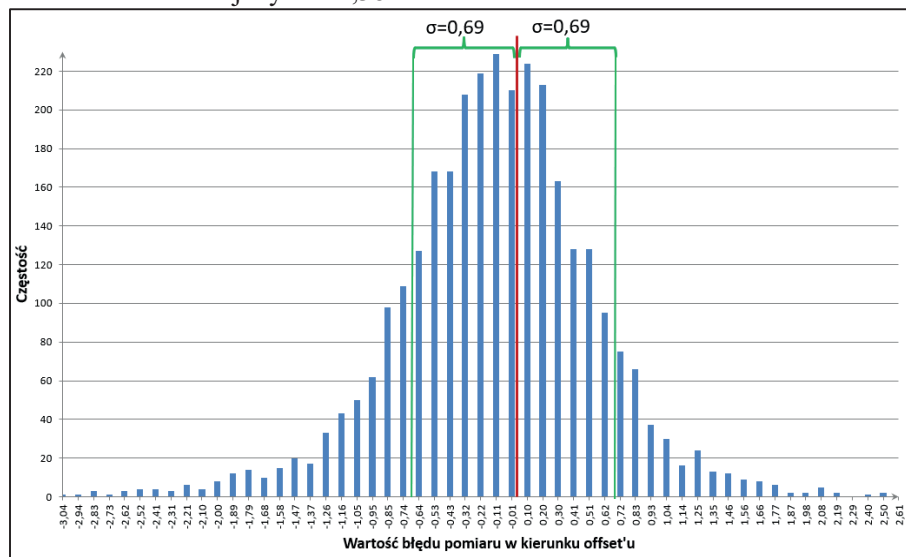
Badania wykonano na specjalnie zaprojektowanym stanowisku umożliwiającym umiejscowienie ramienia w 9 różnych pozycjach w równych odstępach od siebie określając położenie i orientację końca ramienia w stosunku do jego początku w każdej pozycji. W toku badań przebadano 30 ramion uzyskując dane do analizy statycznej. Dla każdego ramienia zostało wykonane 5 pomiarów pierwotnych (etap I) dla pozycji koszyka 1, 3, 5, 7, 9 (rys.4b). Ostatnie pomiary pierwotne były wykonywane w pozycji 5, względem której zostały wykonane pomiary wtórne (etap IV). Pomiar kierunku kości udowej (etap III) był powtarzany 3-krotnie dla każdego ramienia. W pozycjach 1, 3, 5, 7, 9 zostały wykonane jednokrotne pomiary do uzyskania danych korygujących (etap II). Wykonano również 8 pomiarów wtórnych (etap IV): 4 dla główki metalowej i 4 dla główki porcelanowej. Pomiary etapu IV dla każdej główki zostały wykonane z uwzględnieniem niedokładności, które ma za zadanie kompensować program NaviFast6D.

Na podstawie danych z ramion (pliki tekstowe) wykonano w Matlabie obliczenia pomiarów etapów II, III i IV dla pozycji 1, 3, 5, 7, 9. Uzyskano łącznie 3112 pomiarów, które porównano z wartościami oczekiwanymi, a różnice poddano analizie statystycznej. Wartością oczekiwaną μ pomiarów w prezentowanym przypadku pomiaru różnicy pomiędzy pomiarem ramieniem a wartością zadaną powinna wynosić 0, co oznaczono na wykresach (rys.6, rys. 7) linią czerwoną.



Rys. 6 Histogram błędu pomiaru w kierunku wydłużenia

Średnia wartość błędu offsetu wyniosła $-0,17$ a odchylenie standardowe σ uzyskano na poziomie $0,69$ mm. Otrzymane wyniki pomiarów dla przedziału ufności 1σ (oznaczone dwoma liniami zielonymi $\sigma=0,69$) zawierają $73,55\%$ pomiarów. Dla ramienia pomiarowego NaviFast6D, zgodnie z rozkładem Gaussa dla poziomu ufności wynoszącym $95,4\%$, dokładność pomiaru offset'u uzyskana w opisaney procedurze badawczej wynosi $1,38$ mm.



Rys. 7 Histogram błędu pomiaru w kierunku offset'u

Średnia wartość błędu pomiaru wydłużenia wyniosła $-0,07$ a odchylenie standardowe σ uzyskano na poziomie $0,77$ mm. Otrzymane wyniki pomiarów dla przedziału ufności 1σ (oznaczone dwoma liniami zielonymi $\sigma=0,77$) zawierają $69,25\%$ pomiarów. Dla ramienia pomiarowego NaviFast6D, zgodnie z rozkładem Gaussa dla poziomu ufności wynoszącym $95,4\%$, dokładność pomiaru wydłużenia uzyskana w opisaney procedurze badawczej wynosi $1,54$ mm.

3. Wnioski

Wyniki badań ukazują, że ramię NaviFast6D umożliwia wykonanie operacji z większą dokładnością niż przy użyciu dotychczasowych metod, dzięki czemu lekarze będą w stanie zmniejszyć liczbę potrzebnych reoperacji wynikających z wydłużenia nogi pacjenta po operacji. Dotychczas stosowane techniki charakteryzowały się błędem na poziomie ± 6 mm. $87,5\%$ pomiarów z wykorzystaniem ramienia NaviFast6D mieści się w przedziale od -1 mm do 1 mm zmiany wydłużenia nogi, co sugeruje potencjał urządzenia w precyzyjnym wykonywaniu operacji. Wyniki te wskazują na większą dokładność w porównaniu do konkurencyjnych urządzeń [10] [11]. Zauważono, że wynik pomiaru nie jest zależny od położenia ani sposobu wykonania pomiaru w etapie II.

Co więcej, stwierdzono, że ramię NaviFast6D skutecznie kompensuje błędy wynikające z niewłaściwego ustawienia oraz pochylenia podczas pomiaru wtórnego. W toku przeprowadzonych badań wykazano, że pojedynczy pomiar w etapie I może prowadzić do powstania nieusuwalnych błędów w ocenie wydłużenia kończyny dolnej. W związku z tym planowane jest rozszerzenie zastosowania etapu II – zarówno przed, jak i po implantacji – co może przyczynić się do zwiększenia precyzji pomiarowej oraz umożliwić śródoperacyjną ocenę dokładności. Kolejnym etapem rozwoju systemu NaviFast6D są badania kliniczne. Jak każde nowoopracowane urządzenie, NaviFast6D wymaga przeprowadzenia badań klinicznych w celu potwierdzenia jego skuteczności, dokładności oraz niezawodności w warunkach rzeczywistego zabiegu chirurgicznego, co stanowi niezbędny etap procesu walidacji przed szerokim wdrożeniem do praktyki klinicznej.

Literatura

- [1] Dynamika robotów, E. Jezierski, WNT, 2006.
- [2] W. J. Maloney et al. „Leg length discrepancy after total hip arthroplasty,” 2004.
- [3] S. Desai et al. „Leg length discrepancy after total hip arthroplasty: a review of literature” *Musculoskeletal Medicine*, 2013.
- [4] M. S. Austin et al. „Stability and Leg Length Equality in Total Hip Arthroplasty” *The Journal of Arthroplasty*, 2003.
- [5] W. D. Abraham et al. „Leg Length Discrepancy in Total Hip Arthroplasty” *Orthopedic Clinics of North America*, 1992.
- [6] S. Banerjee et al. „Robot-assisted total hip arthroplasty” *Expert Review of Medical Devices*, 2015.
- [7] D. H. Nawabi et al. „Haptically guided robotic technology in total hip arthroplasty: a cadaveric investigation” *Proc Inst Mech Eng H.*, 23 03 2013 .
- [8] M Van den Bempt et al. „Toward a higher accuracy in orthognathic surgery by using intraoperative computer navigation, (...)” *Journal of Cranio-Maxillo-Facial Surgery*, 2018.
- [9] Braun Sharing Expertise, AESCULAP® OrthoPilot® Elite.
- [10] S. M.A. Arachchi et al. „Validation of a CT-free navigation system for the measurement of native” doi.org/10.1504/IJBET.2015.069401
- [11] J. Parvizi et al. „A new mini-navigation tool allows accurate,” *Medical Devices: Evidence and Research*, 2018.
- [12] www.orthalign.com/wp-content/uploads/2021/07/001273-Rev-A.pdf.
- [13] G Zheng et al. „A Hybrid CT-Free Navigation System for Total Hip Arthroplasty” *Computer Aided Surgery*, 14 5 2002.
- [14] A. Kobierska et al. „Modelling and testing the measurement repeatability of linear and angular positions (...)” DOI:10.1016/j.measurement.2022.111960
- [15] R. Lartizien et al. „Improvement in accuracy of maxillary repositioning of Le Fort I osteotomy with OrthopilotTM Navigation System: evaluation of 30 patients” *British Journal of Oral and Maxillofacial Surgery*, 2020.

- [16] J. Jia et al. „Clinical efficacy of orthopilot navigation system versus conventional manual of total hip arthroplasty” 2019.
- [17] F. Sasazaw et al. „The Simple and Accurate Cup Setting with Portable Navigation System in Total Hip Arthroplasty” Case Study6/2017,orthalign.com.
- [18] A. Bryszewski et al. „Pneumatics can be precise - Methodology of building a stand with high” Precision Engineering, 2023.
- [19] A. Kobierska et al. „Estimation of Orientation and Position of the Manipulator Using the Extended Kalman Filter Based on Measurements From the Accelerometers” JAMRIS, DOI: 10.14313/JAMRIS/2-2019/15.
- [20] P. Rakowski et al., „Accuracy and repeatability tests on 6D measurement arm” Mechanics and Mechanical Engineering, 2017.
- [21] Ming Bai et al., „Calibration method based on models and least-squares support (...)” IEEE Access,DOI:10.1109/ACCESS.2021.3115949
- [22] A. Quast et al. „Traditional face-bow transfer” Oral and Maxillofacial Surgery, 2018.

Experimental verification of the accuracy of the LLD and LOD measurement procedure using the NaviFast6D measuring arm

The paper presents a procedure for verifying the accuracy of LLD (Leg Length Discrepancy) and LOD (Leg Offset Discrepancy) measurements using the NaviFast 6D measuring arm. These parameters are intended for use during hip arthroplasty surgery. A custom test stand was constructed, and measurement accuracy tests were conducted under laboratory conditions simulating those in the operating room. The measurement process was divided into stages corresponding to the surgical procedure. Determining the accuracy of the NaviFast 6D arm is essential to understand how much the measurement results deviate from the actual values. The use of this arm may help surgeons reduce the number of reoperations caused by leg length discrepancies following hip replacement surgery.

A vision based module for implementing interactive mirror-like therapy using Stretchbox device*

Igor Zubrycki¹, Kamila Kuczmarska¹

Abstract

This paper introduces a vision-based control module for the Stretchbox hand rehabilitation device, enabling interactive, mirror-like therapy sessions. Leveraging a standard tablet camera enhanced with a fisheye lens and the MediaPipe Hands framework, the system tracks the user's unaffected hand to guide the motion of the affected hand via the Stretchbox's elastic band actuation. We detail the implementation, including a 1-Degree-of-Freedom (1-DOF) control signal generation based on identifying and tracking the hand movement axis with the highest variance, and its mapping to the device's motors. Preliminary experimental verification with healthy volunteers shows the system successfully replicates primary hand movements (flexion/extension, abduction/adduction) with a measurable latency. This approach enhances user engagement compared to previous passive modes, utilizing accessible web technologies (JavaScript, Web Bluetooth) for easy deployment without additional hardware costs or complex setup.

1. Introduction and motivation

The Stretchbox device, a prototype developed at Lodz University of Technology, is a simple mechanism capable of driving multiple hand movements by stretching a thin rubber band. While not commercially available, it is currently being evaluated in hospitals in Lodz and Poznan. Intended for post-stroke rehabilitation, it enables finger extension, finger adduction (spreading), and wrist exercises using a pair of motors. Although there is a single primary direction of stretch, the band's flexibility allows for more complex movements, such as finger adduction.

Recovery of finger and wrist extension is crucial, as these movements strongly influence patients' quality of life. The device was designed to address limited access to hand rehabilitation therapies. The previous version of the device, described in [12], featured a single setup where the therapist controlled the amount of stretch and movement by setting maximum and minimum levels; movement then occurred at a predetermined speed. Safety and other control aspects were managed by a behavior tree high-level controller. In this form, the device was used in a pilot study and demonstrated improvements in finger range of motion and functional outcomes.

*Supported by National Centre for Research and Development, grant No. LIDER/50/0203/L-11/19/NCBR/2020. Authors would like to thank Witold Sierocki for design of tablet fish lens holder.

¹Lodz University of Technology, Institute of Automatic Control, Stefanowskiego 18, 90-537 Lodz, Poland

igor.zubrycki@p.lodz.pl

Research indicates that active patient participation, focus, and motivation significantly influence the rate of recovery as it highly depends on motor learning or neuroplasticity in general: the brain's ability to reorganize and form new connections to recover function [11]. This aspect was not leveraged in the previous generation of the Stretchbox device. In this work, we introduce a camera-based extension enabling mirror-therapy-like rehabilitation. Classical mirror therapy, introduced by Ramachandran et. al, uses a mirror placed between the healthy and affected hands, creating the illusion that the affected hand moves through patient intention [6]. It has been successfully used with stroke patients, and meta-analyses based on randomized controlled trials have demonstrated improvements in activities of daily living scores, Fugl-Meyer scores, and Box-and-Block test performance [7, 3]. Notably, Hsieh's study indicates that bimanual mirror therapy, where patients actively move the affected hand, yields better results compared to unimanual therapy, where patients only move the unaffected hand and observe the mirror.

Mirror therapy can be enhanced by technology. In their scoping review, this enhanced approach is referred to as *second-generation mirror therapy*, which incorporates virtual environments or robotic systems [1]. The authors list several potential benefits of this approach, including promoting neuroplasticity through multisensory feedback, increasing the range and complexity of tasks, enhancing motivation and engagement, and providing more intense and repetitive training. However, they also note significant costs associated with robotic devices and the requirement for highly qualified operators.

There are multiple technical challenges involved in developing an active mirror-therapy device. The human hand has numerous degrees of freedom, so driving its motion with traditional robotic mechanisms requires either multiple motors (or multiple linkage mechanisms) per finger or intelligently designed passive elements to create underactuated yet functional movements via coupled joint motions or passive adaptability. An alternative approach involves soft robotic actuators, which offer inherent compliance. Pneumatic soft actuators, for example, can bend or straighten to induce finger flexion or extension. These actuators are commonly implemented as wearable gloves or exoskeletons.

Another significant challenge is accurately capturing the pose of the healthy hand to mirror its movements. Here, important factors include the precision of pose measurement and the speed of data acquisition, as well as the overall teleoperation loop latency. Historically, solutions relied on passive exoskeletons with encoders to measure joint motion or sensor-equipped gloves that detect finger flexion. Vision-based systems offer an alternative solution; however, 3D-vision-based devices such as Kinect or Leap Motion typically require substantial computational power and more complex setups compared to simpler RGB (monocular) camera-based systems.

The rapid advancement in deep learning has led to the emergence of high-quality and lightweight hand-tracking solutions, notably MediaPipe Hands from Google. This solution was trained on a dataset comprising 16,000 real-world images and a synthetic dataset of 100,000 examples [10]. Initially developed to simplify the creation of gesture-controlled and augmented reality applications, MediaPipe Hands achieves excellent efficiency, with an average latency of 17.12 milliseconds per frame for hand landmark detection on a Pixel 6 phone [2]. Its availability across multiple platforms,

including mobile and web applications, makes it highly accessible and suitable for integration into therapeutic devices [2].

Any technical solution for post-stroke hand rehabilitation must accommodate users' constrained resources and environments. Hand therapy sessions are typically scheduled in 30–60 minute blocks, leaving minimal therapist time for device setup or calibration, so systems must be operable with minimal clinician oversight [8, 5]. Moreover, informal caregivers shoulder substantial emotional and physical burdens that limit their capacity to maintain and operate complex equipment [4].

Mirror therapy in its simplest form requires only a low-cost mirror box (i \$20), whereas VR-enhanced setups cost approximately US \$252, and robotic mirror-therapy devices can exceed US \$25 000 [1]. Spatial constraints further dictate a small device footprint: hand rehabilitation often occurs in activities-of-daily-living (ADL) rooms or at the patient's bedside, and vision-based systems like Microsoft Kinect demand roughly 2–3 m of clear line-of-sight and controlled lighting, which may be impractical in typical clinical or home settings [9].

2. Implementation of Hand-Tracking-Based Capability

To leverage the benefits of active patient participation highlighted in the previous section and address the limitations of the original Stretchbox controller, we developed a novel control modality based on real-time hand tracking. The primary goal was to create a system capable of delivering active, mirror-therapy-like rehabilitation, where the movements of the patient's unaffected hand, captured by a camera, directly drive the Stretchbox actuation for the affected hand.

This development was guided by several key requirements and constraints. Firstly, the system needed to technically support the primary therapeutic hand movements facilitated by the Stretchbox: finger extension/flexion and abduction/adduction. Secondly, to maintain the device's accessibility and low cost, the implementation had to rely on the existing computational hardware, namely the tablet used for the device's Human-Machine Interface (HMI). Thirdly, the entire setup, including the camera and the device, needed to remain compact enough to fit comfortably on a standard therapy training table. Finally, the chosen solution needed to be adaptable for future extensions and relatively easy to develop and maintain, favouring readily available and well-supported technologies.

Considering these factors, we opted for a RGB-vision-based approach using the tablet's built-in camera, enhanced with a fisheye lens attachment to allow close-range hand observation. This setup processes the video stream using the MediaPipe Hands framework to extract kinematic information from the healthy "leader" hand. This information is then translated into control commands for the Stretchbox's elastic band actuation system, described below, to mirror the intended movement in the affected "follower" hand. The setup is presented in fig. 1. It consists of existing Stretchbox device and Android tablet. The tablet has a 3D printed attachment with a fisheye lens camera which considerably extends its field of view.

2.1. Stretchbox Actuation and Control Principle

The actuation system of "Stretchbox" consists of a flat elastic band driven by two independently controlled motorised rollers housed within a frame (see fig. 2a and

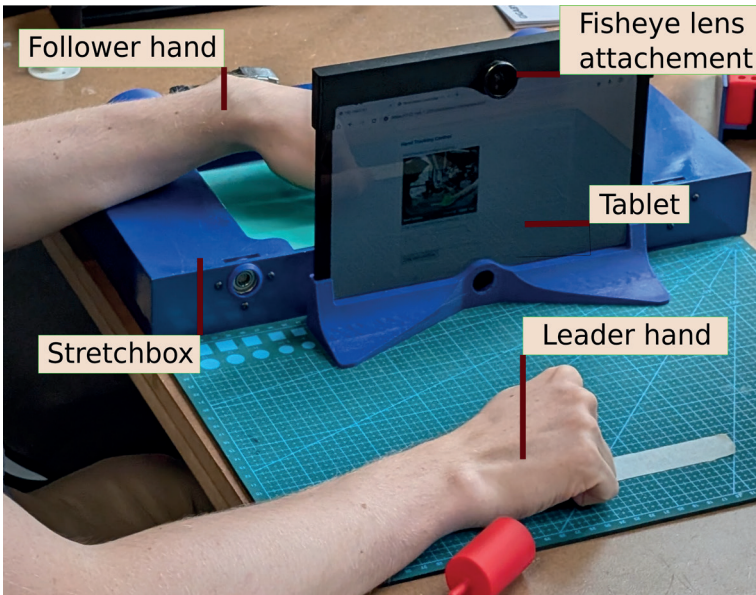


Fig. 1: A setup for a mirror-like therapy. Tablet with the HMI is pointing its camera at the leader hand. Camera is modified with a fisheye lens to allow for much shorter distance for observation

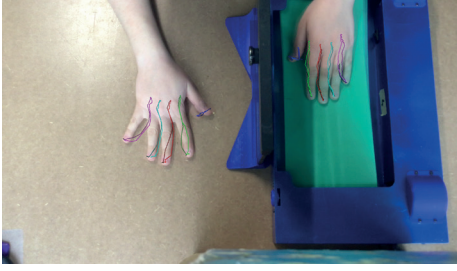
fig. 2b for illustration context). By rotating the rollers, two independent mechanical variables of the band segment interacting with the user's hand can be commanded:

1. **Band translation, Δx :** Obtained by turning one roller or both in the same direction. This shifts the band laterally (e.g., proximally/distally along the fingers) with negligible change in its length between attachment points.
2. **Band elongation/tension, ΔL :** Obtained by turning the rollers in opposite directions. This increases the total length of the band segment between the rollers and therefore its tension.

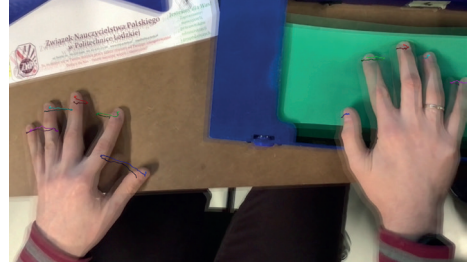
Although the hardware offers two degrees of freedom (P_L, P_R motor positions, controlling Δx and ΔL), many functional hand rehabilitation exercises involve coordinated movements along specific paths that can be effectively parameterized by a single degree of freedom (1DOF). For example:

- **Finger extension/flexion:** To train opening and closing of the hand, the primary requirement is to vary the band's lateral position (Δx) while keeping its tension (ΔL) relatively constant. As the band slides distally over the dorsum of the fingers, it assists extension; reversing the motion assists flexion. The resulting fingertip motion is predominantly linear along the finger axis (fig. 2a).
- **Finger abduction/adduction:** To train spreading and pinching, the rollers are driven primarily in opposite directions to increase band tension (ΔL) while minimizing net translation ($\Delta x \approx 0$). The resulting quasi-uniform radial force allows each fingertip to travel along its natural diverging/converging arc (fig. 2b).

Based on the observation that these key therapeutic motions can be driven by a single control parameter, we established a teleoperation scheme where the user's hand



a – Finger extension/flexion driven primarily by band translation (Δx). Trajectories show predominantly proximal/distal motion



b – Finger abduction/adduction driven primarily by band elongation (ΔL). Trajectories show diverging/converging arcs

Fig. 2: Illustration of Stretchbox actuation modes and resulting fingertip trajectories (overlaid) for different therapeutic movements. The colors of the trajectories represent different fingers, with lighter colors showing diverging and darker converging stages of motion

motion, captured via a standard webcam, generates this 1DOF control signal. For the system, the same tablet used for the control interface can run the hand tracking. The tablet's camera view can be augmented with a fisheye lens attachment to allow observation of the hand from a short distance, accommodating typical therapy setups.

2.2. Teleoperation Scheme: From Hand Motion to Motor Commands

The teleoperation scheme maps the detected hand motion to coordinated commands for the two Stretchbox motors (P_L, P_R). This involves deriving a single normalized control signal, $u(t)$, from the hand tracking data and using it to interpolate between therapist-defined start and end points for the exercise.

Control Signal Generation ($u(t)$). The scheme uses the MediaPipe Hands framework [10] to process the camera stream and extract hand landmarks. A single control signal representing the progression of the therapeutic motion is generated in two phases:

1. **Learning Phase:** The user performs the target therapeutic hand motion (e.g., full flexion to full extension, or full adduction to full abduction) for a brief period (e.g., 5 seconds). During this phase:
 - MediaPipe detects 21 key 3D hand landmarks, providing their world coordinates $\mathbf{p}_k(t) \in \mathbb{R}^3$ for $k \in \{0, \dots, 20\}$ at time t .
 - The system calculates the Euclidean distances $d_{ij}(t) = \|\mathbf{p}_j(t) - \mathbf{p}_i(t)\|_2$ between all pairs of landmarks (i, j) where $i < j$.
 - The system identifies the landmark pair (i^*, j^*) whose distance exhibits the maximum variance over the learning period:

$$(i^*, j^*) = \operatorname{argmax}_{i, j: i < j} \{\operatorname{Var}[d_{ij}(t)]\}$$

. This distance $d_{i^*j^*}(t)$ corresponds to the dimension of greatest movement during the demonstrated exercise and serves as the raw 1DOF input signal.

2. **Tracking Phase:** After the learning phase identifies the key distance $d_{i^*j^*}(t)$, the system continuously tracks it during teleoperation. This raw distance is normalized based on its observed range to create a stable control signal:
 - The system records the minimum (d_{min}) and maximum (d_{max}) values attained by the tracked distance $d_{i^*j^*}(t)$ since the learning phase concluded. Let $d_{min} = \min_{\tau \geq t_{learn_end}} \{d_{i^*j^*}(\tau)\}$ and $d_{max} = \max_{\tau \geq t_{learn_end}} \{d_{i^*j^*}(\tau)\}$.
 - A normalized control signal $u(t) \in [0, 1]$ is calculated:

$$u(t) = \text{clamp} \left(\frac{d_{i^*j^*}(t) - d_{min}}{d_{max} - d_{min}}, 0, 1 \right)$$

, where $\text{clamp}(v, v_{min}, v_{max}) = \max(v_{min}, \min(v, v_{max}))$.

Mapping to Motor Commands. A critical part of the setup involves the therapist defining the exact start and end configurations of the Stretchbox for the intended exercise using the interface buttons. This establishes the boundary conditions for the movement:

- Start Configuration: Motors at positions $(P_{L,min}, P_{R,min})$
- End Configuration: Motors at positions $(P_{L,max}, P_{R,max})$

These pre-set configurations determine the specific trajectory in the 2D motor command space (P_L, P_R) that corresponds to the 1D therapeutic task (e.g., defining the range and combination of Δx and ΔL).

During teleoperation, the normalized hand tracking signal $u(t)$ linearly interpolates the motor commands along this defined trajectory. The target motor positions $P_L(t)$ and $P_R(t)$ at time t are computed as:

$$\begin{aligned} P_L(t) &= P_{L,min} + u(t) \cdot (P_{L,max} - P_{L,min}) \\ P_R(t) &= P_{R,min} + u(t) \cdot (P_{R,max} - P_{R,min}). \end{aligned}$$

These target positions $(P_L(t), P_R(t))$ are then sent to the Stretchbox device via Bluetooth Low Energy (BLE). This allows the user's single, most prominent hand movement feature (captured as $u(t)$) to intuitively drive the device through the precise range and type of motion (flexion/extension or abduction/adduction) pre-defined by the therapist.

2.3. Software Implementation

The described teleoperation system is realized as a JavaScript-based web application, streamlining deployment and accessibility. It leverages the MediaPipe library [10] for robust real-time hand tracking from the webcam feed and integrates OpenCV.js to correct for fisheye lens distortion. The core hand tracking functionality, including the 1-DOF signal generation, is encapsulated within a self-contained JavaScript module. This module also provides a compact graphical interface displaying the camera view, detected landmarks, and tracking status indicators (see fig. 3). Crucially, the application utilizes the Web Bluetooth API for direct communication with the Stretchbox device, eliminating the need for any software installation and allowing operation directly within compatible web browsers.

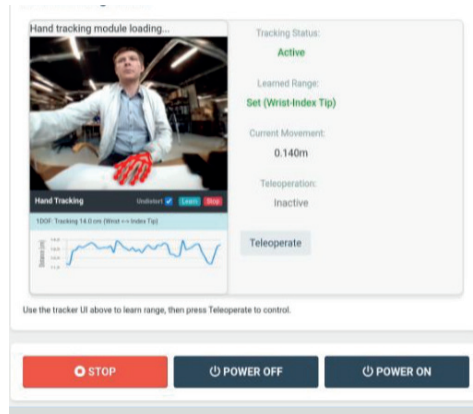


Fig. 3: Screenshot of the interaction module in the tablet application

3. Experimental Verification

To assess the performance of the vision-based teleoperation system, preliminary experiments were conducted focusing on motion replication accuracy and system latency.

3.1. Experimental Setup and Procedure

Experiments involved healthy volunteers ($N=4$) from the Skaner Robotics Group at Lodz University of Technology. Each participant sat at a table with the Stretchbox setup (see fig. 1). They were instructed to place one of their hands (follower hand) passively within the Stretchbox band, while using their other hand (leader hand) positioned in view of the tablet's fisheye camera (approx. 30-40 cm distance) to control the device. Before each trial type, the participant performed the corresponding movement (e.g., maximal finger extension/flexion or abduction/adduction) for 5 seconds to allow the system's 1-DOF learning phase to identify the primary motion axis. Subsequently, participants performed trials involving:

1. **Latency Test:** Starting from a stationary pose, perform a rapid, large-amplitude movement (extension or abduction).
2. **Range of Motion (RoM) Test:** Move the leader hand smoothly through its full comfortable range for either finger extension/flexion or abduction/adduction.

All sessions were recorded using an external camera at approximately 30 fps.

3.2. Data Processing and Metrics

Video recordings were first stabilized using `ffmpeg` filtering to compensate for minor camera shake. Subsequently, the MediaPipe Hands framework [10] was applied offline to extract 2D screen coordinates of the 21 landmarks for both the leader and follower hands in each frame. Frame-by-frame kinematic analysis was performed using custom scripts and kinematic analysis software to quantify performance based on the following metrics:

Time Delay (T_{delay}): Calculated from the latency test trials as the time difference between the onset of leader hand motion ($f_{leader,start}$) and the corresponding onset of follower hand motion ($f_{follower,start}$):

$$T_{delay} = (f_{follower,start} - f_{leader,start})/fps, \quad (1)$$

where fps is the video frame rate. Motion onset was determined as the frame where displacement relative to the initial stationary pose exceeded a predefined threshold.

Range of Motion Difference (ΔRoM): Calculated from the RoM test trials to evaluate tracking fidelity. The general definitions are:

$$RoM = val_{max} - val_{min}, \quad (2)$$

$$\Delta RoM = RoM_{leader} - RoM_{follower}, \quad (3)$$

, where val represents the specific metric being evaluated. The primary metrics used for RoM comparison were:

- **Finger Extension ($d_{ext, mid}$):** The change in distance between the middle finger base (MCP, landmark 9) and the finger tip (landmark 12), used as a proxy for comparing the extent of finger extension/flexion.

$$\Delta RoM_{ext} = RoM_{leader}(d_{ext, mid}) - RoM_{follower}(d_{ext, mid}), \quad (4)$$

- **Finger Abduction ($d_{span, I-P}$):** The span distance between the tips of the index finger (landmark 8) and the pinky finger (landmark 20), measured using 3D world coordinates, representing the overall finger spread.

$$\Delta RoM_{abd, span} = RoM_{leader}(d_{span, I-P}) - RoM_{follower}(d_{span, I-P}). \quad (5)$$

Visualization Metric: For qualitative visualization purposes only, particularly in Fig. 4 (bottom row), the sum of the middle finger's Proximal Interphalangeal (PIP) and Distal Interphalangeal (DIP) joint angles ($Angle_{sum, mid}$) was also calculated using a 2D projection of world coordinates (landmarks 9, 10, 11, 12), based on the method in the analysis script:

$$Angle_{sum, mid}(t) = Angle_{PIP}(t) + Angle_{DIP}(t). \quad (6)$$

This metric was not used for the quantitative ΔRoM comparison but helps visualize the flexion/extension pattern.

3.3. Results

Preliminary experiments quantified the system's performance. For the finger extension/flexion mode, analysis (fig. 4) indicated successful tracking of the leader hand's motion profile. The average difference in range of motion for finger extension (ΔRoM_{ext} based on $d_{ext, mid}$, (4)) across $n=4$ trials was 1.475 ± 0.515 cm (mean $\pm$ 95% CI). For the finger abduction/adduction mode, the average difference in range of motion for finger span ($\Delta RoM_{abd, span}$ based on $d_{span, I-P}$, (5)) across $n=4$ trials was

2.535 ± 1.631 cm (mean $\pm$ 95% CI). These results suggest reasonable fidelity in replicating the overall extent of the primary movements. A consistent visual offset (approx. 1.4 cm in the representative extension trial shown in fig. 4) was noted between absolute leader/follower positions, potentially due to calibration or setup variations. Movement speeds appeared comparable during execution phases.

However, a significant time delay (T_{delay}) of approximately 0.46 ± 0.05 seconds (mean $\pm$ SD, N=3 participants) was measured ((1), fig. 4). While finger spreading ((5)) was replicated reasonably well in terms of RoM, qualitative analysis indicated less accurate mirroring of thumb abduction, suggesting limitations related to the device’s single-band actuation mechanism for independent thumb control.

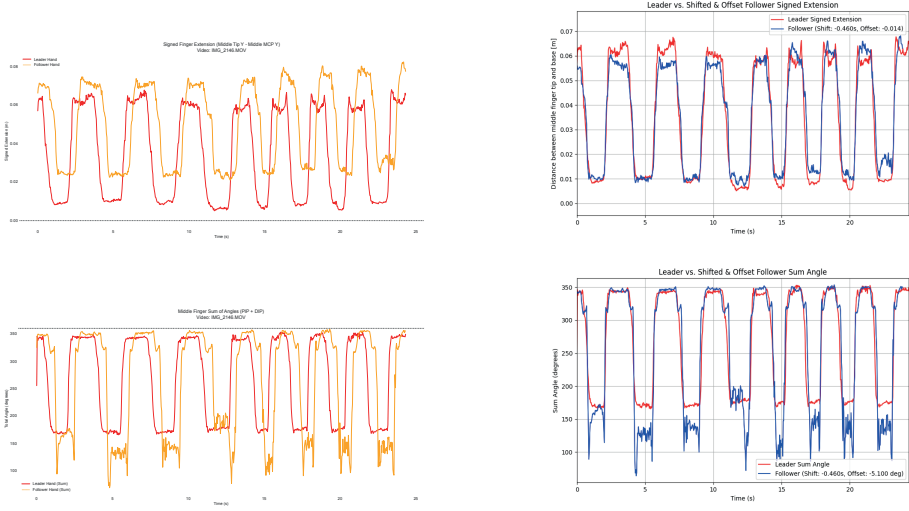


Fig. 4: Tracking of the leader hand’s profile, measured through separate camera setup. Top-left: Middle finger base-tip distance ($d_{ext, mid}$) for leader (red) and follower (yellow). Top-right: Comparison on the shape of trajectories (time and distance offsets removed) red: leader hand, blue: follower. Bottom-left: Sum of middle finger joint angles ($Angle_{sum, mid}$, calculated for visualization). Bottom-right: Follower vs. leader angle sum. ($T_{delay} \approx 0.46$ s; $\Delta RoM_{ext} \approx 1.48 \pm 0.52$ cm)

4. Discussion and conclusions

The vision-based module successfully enables interactive, mirror-like control for the Stretchbox, leveraging standard tablet hardware and accessible web technologies (MediaPipe, Web Bluetooth) [10]. This promotes active patient participation, crucial for neuroplasticity [11], without increasing hardware cost or complexity, addressing usability constraints in rehabilitation [8, 4]. While preliminary tests show good replication of primary movement RoM (except fine thumb abduction, likely due to mechanical limits), a notable latency (≈ 0.46 s) exists. This delay may limit its effectiveness for inducing a pure mirror illusion [1] but offers enhanced interactivity over previous passive modes [12] and could be acceptable for active-assistive guidance. The web-

based platform provides significant advantages for future extensibility compared to prior implementations [12].

In conclusion, we developed and demonstrated an accessible vision-based interactive control system for the Stretchbox device, enhancing its potential for active patient engagement using readily available web technology. Future work will prioritize clinical validation to assess therapeutic efficacy [7, 3], significant system latency reduction for improved responsiveness, and exploration of adaptive control strategies or refined movement mapping algorithms to enhance performance and broaden the scope of addressable exercises.

References

- [1] N. Darbois, A. Guillaud, N. Pinsault. Do robotics and virtual reality add real progress to mirror therapy rehabilitation? a scoping review. *Rehabilitation Research and Practice*, 2018, Vol. 2018, pp. 6412318. Epub 2018 Aug 19.
- [2] Google AI. Hand Landmarker — MediaPipe — Google AI Edge. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker, 2024. Accessed: 2024-08-01.
- [3] Y.-L. Hsieh et al. Effects of mirror therapy on motor and functional recovery of the upper extremity in subacute stroke: Systematic review and meta-analysis. *PM&R*, 2025, Vol. n/a, No. n/a.
- [4] J. Oliva-Moreno et al. Determinants of informal care, burden, and risk of burnout in caregivers of stroke survivors: The conoces study. *Stroke*, 2018, Vol. 49, pp. 140–146.
- [5] *Physical Therapy: Summary of Recommendations by Source*. Red. National Library of Medicine. National Center for Biotechnology Information 2015.
- [6] V. S. Ramachandran, E. L. Altschuler. The use of visual feedback, in particular mirror visual feedback, in restoring brain function. *Brain*, 2009, Vol. 132, No. Pt 7, pp. 1693–1710.
- [7] I. D. Saragih et al. Effects of mirror therapy on upper limb motor function of patients with stroke: A systematic review and meta-analysis of randomized controlled trials. *Clinical Rehabilitation*, 2025, Vol. 39, No. 1, pp. 23–34. Epub 2025 Jan 21.
- [8] Wayne UNC Health Care. Hand therapy, 2025.
- [9] D. Webster, C. Ozkan. Systematic review of kinect applications in elderly care and stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2014, Vol. 11, pp. 108.
- [10] F. Zhang et al. Mediapipe hands: On-device real-time hand tracking, 2020.
- [11] V. Zotey et al. Adaptive neuroplasticity in brain injury recovery: Strategies and insights. *Cureus*, September, 2023, Vol. 15, No. 9, pp. e45873.
- [12] I. Zubrycki. Behavior tree based controller for a soft rehabilitation device stretch-box. In: 2024 13th International Workshop on Robot Motion and Control (RoMoCo). *Proceedings*, 2024, pp. 168–173.

Układ sterowania systemem wspomagającym biopsję

Adam Kurnicki¹, Bartłomiej Stańczyk²

Streszczenie

W artykule przedstawiono układ sterowania systemem wykorzystywanym do wspomagania badania biopsyjnego. System ten został opracowany w ramach projektu BONITO "Konfigurowalne ramię do stabilizacji trajektorii igły biopsyjnej". Na początku artykułu pokazano strukturę całego systemu i omówiono zasadnicze elementy jego układu sterowania. Następnie przedstawiono problematykę związaną z wyznaczaniem sygnałów zadanych pozycji i orientacji w algorytmie sterowania. Na koniec omówiono niektóre wyniki jego ewaluacji.

1. Wstęp

Biopsja jest jedną z najczęściej stosowanych metod diagnostycznych, pozwalającą na pobranie próbek tkanek do analizy histopatologicznej. Chociaż tradycyjne techniki ręcznego wykonywania biopsji przez lekarza diagnostę są skuteczne, to niosą ze sobą ryzyko błędów wynikających z ograniczonej precyzji oraz możliwości uszkodzenia zdrowych tkanek [10]. Przeprowadzenie takiego badania jest też zwykle czasochłonne, szczególnie w przypadku konieczności poboru materiału ze stosunkowo niewielkich rozmiarów guzów i związanej z tym konieczności wielokrotnego powtarzania czynności. Ze względu na fakt starzenia się społeczeństwa i związane z tym zwiększające się zapotrzebowanie na przeprowadzanie tego typu badań, przy jednoczesnym niedoborze personelu medycznego, w ostatnich latach coraz większe znaczenie zyskują stale rozwijające się systemy robotyczne wspomagające wykonywanie biopsji. Zapewniają one wyższą precyzję, bezpieczeństwo i skracają czas wykonania zabiegu [2]. Istotnym elementem takich systemów jest ich układ sterowania, który m.in. wspomaga diagnostę w określeniu miejsca i innych parametrów poboru próbek oraz umożliwia realizację precyzyjnego ich pobrania.

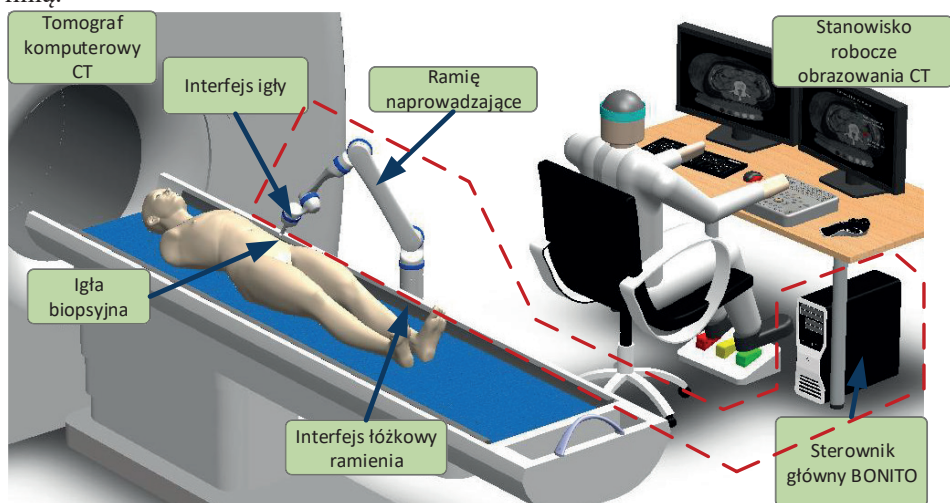
W literaturze naukowej można znaleźć wiele badań poświęconych rozwojowi manipulatorów robotycznych do celów medycznych. Użycie chociażby da Vinci Surgical System, w chirurgii, stanowiło fundament do dalszego rozwoju technologii manipulacji w medycynie. Biopsje igłowe wykonywane przy użyciu robota można wykonywać w różnych narządach, m.in. w mózgu, piersiach, płucach, prostaty czy

¹ Katedra Automatyki i Metrologii, Wydz. Elektrotechniki i Informatyki, Politechnika Lubelska, ul. Nadbystrzycka 38a, 20-618 Lublin, e-mail: a.kurnicki@pollub.pl

² ACCREA Medical Robotics, ul. Hiacyntowa 20, 20-143 Lublin, e-mail: b.stanczyk@accrea.com

wątrobie [9]. Podobnie jak w przypadku tradycyjnej procedury, biopsje robotyczne wykonywane są pod kontrolą obrazową, przy użyciu takich metod, jak: tomografia komputerowa (CT) [2, 7, 8], ultrasonografia (USG) lub obrazowanie metodą rezonansu magnetycznego (MRI). Manipulatory dedykowane do biopsji rozwijane są dla dwóch zastosowań: wewnątrz urządzenia obrazującego [3, 7, 8] oraz poza nim [9] tj. nie biorą udziału w obrazowaniu razem z pacjentem. Te pierwsze zastosowania podlegają ograniczeniom co do rozmiaru oraz materiału, z którego są wykonywane. Jednakże wyposażone w odpowiednie systemy sterowania mogą realizować operację wprowadzania igły zdalnie a nawet autonomicznie lub półautonomicznie [3, 10]. Natomiast te drugie realizują nawigację do miejsca wkłucia po analizie wcześniej wykonanych obrazów. Ich wykonanie oraz funkcjonowanie ich układu sterowania wymaga znacznie większej precyzji. Ogromne znaczenie mają tutaj systemy rejestracji obrazów [6], które często wykorzystują do tego celu algorytmy AI.

Firma ACCREA Medical Robotics, realizując projekt “Konfigurowalne ramię do stabilizacji trajektorii igły biopsyjnej” w ramach Poddziałania 1.1.1 Programu Operacyjnego Inteligentny Rozwój 2014-2020, współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego, opracowała system BONITO, dedykowany do biopsji zmian nowotworowych w tkankach brzusznych i piersiowych (np. wątroby, płuc). Ogólny zarys systemu BONITO przedstawia rys. 1. Jego główne elementy składowe znajdują się w obszarze obrysowanym czerwoną, przerywaną linią.



Rys. 1. Ogólny zarys systemu BONITO

Lekarz diagnosta w oparciu o specjalne oprogramowanie do przetwarzania obrazów medycznych, zaimplementowane na sterowniku głównym systemu BONITO, dokonuje wskazania miejsca i kierunku wbicia igły/-ieł biopsyjnej/-ych w unieruchomione ciało pacjenta, a następnie uruchamia procedurę planowania i wykonania badania. Efektem wykonanie tej procedury jest przemieszczanie

końcówki interfejsu igłowego manipulatora 7 DoF, według zaplanowanej trajektorii, do miejsc, w których lekarz dokonuje: wprowadzenia igły w ciało pacjenta i poboru tkanek. Manipulator jest mocowany do łóżka pacjenta za pomocą specjalnego interfejsu połączonego na sztywno z sekcją czterech znaczników rejestrowanych w obrazach tomografu komputerowego. Znaczniki te, na drodze wspomnianego wyżej przetwarzania obrazów CT, służą do precyzyjnego spozycjonowania ciała pacjenta względem manipulatora.

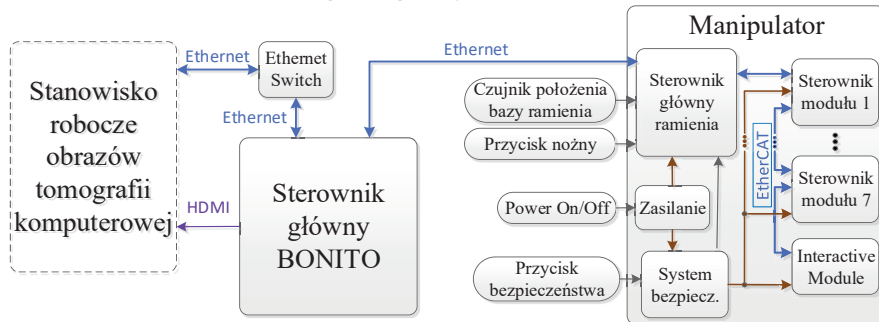
W kolejnych rozdziałach artykułu zaprezentowany zostanie układ sterowania a w szczególności algorytm sterowania implementowany w sterowniku głównym. Podjęta zostanie również problematyka generacji trajektorii ruchu manipulatora z naciskiem na problematykę wyznaczania orientacji zadanej dla końcówki manipulatora.

2. System BONITO i jego układ sterowania

2.1. Struktura systemu

System BONITO, którego architektura sprzętowa została przedstawiona na rys. 2, składa się z dwóch zasadniczych elementów:

- manipulatora (ramienia naprowadzającego) ze specjalistycznymi interfejsami,
- sterownika głównego (komputera PC z systemem Ubuntu), stanowiącego główną część stanowiska roboczego diagnosty.



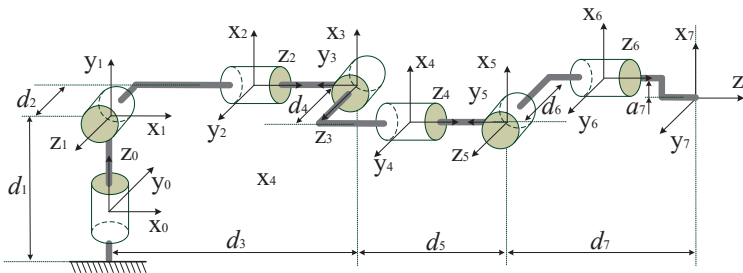
Rys. 2. Architektura sprzętowa systemu BONITO

Zgodnie z powyższą architekturą, dane obrazowe z tomografu CT (lub rezonatora magnetycznego) są przesyłane do sterownika głównego w standardzie DICOM poprzez sieć Ethernet. Bezpośrednie łącze ethernetowe wykorzystywane jest również do przesyłania komend sterujących do sterownika manipulatora ze sterownika głównego oraz odbierania przez niego informacji o aktualnym stanie pracy i konfiguracji kinematycznej manipulatora. Ramię montowane jest po jednej stronie łóżka tomografu, w jego środkowej części, za pomocą interfejsu mechanicznego, umożliwiającego jego ręczne przesuwanie wzdłuż łóżka, tak aby umożliwić

wykonanie obrazowania bez konieczności demontażu manipulatora. Zamontowany tam absolutny enkoder liniowy, umożliwia identyfikację aktualnej pozycji bazy ramienia względem interfejsu. Z interfejsem połączone jest kilka sekcji znaczników (markerów), które muszą być widoczne i rozróżnialne w obrazie CT pacjenta. Z bazą manipulatora połączone są elektrycznie (kablowo): przycisk awaryjny i przełącznik nożny (pedał), które fizycznie są ulokowane na stanowisku roboczym diagnosty. Stanowią one element systemu bezpieczeństwa, gdzie naciśnięciem przełącznika nożnego, diagnosta zezwala na ruch ramienia naprowadzającego.

2.2. Ramię naprowadzające

Ramieniem naprowadzającym jest manipulator o siedmiu stopniach swobody, którego struktura kinematyczna (przedstawiona na rys. 3) oraz rozmiar, zbliżone są do ramienia ludzkiego.



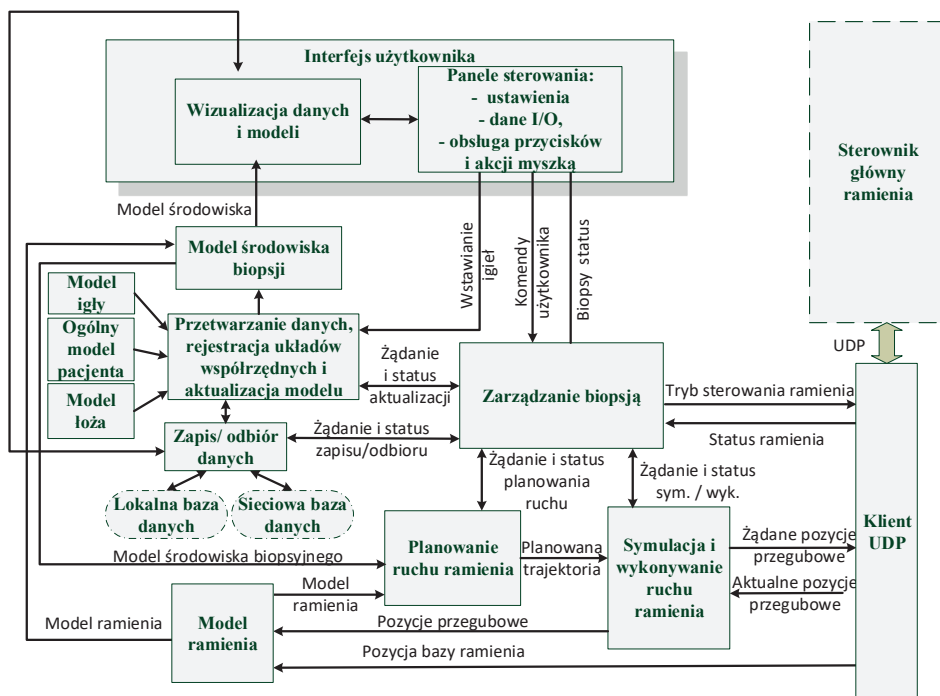
Rys. 3. Struktura kinematyczna manipulatora naprowadzającego BONITO

Konstrukcja manipulatora jest relatywnie sztywna. Zbudowany on został głównie z materiałów aluminiowo-stalowych. Moduły napędowe jego przegubów stanowią silniki BLDC firmy Maxon wraz z przekładniami harmonicznymi firmy Harmonic Drive. Ze względu na stosunkowo duże wymagania co do precyzji pozycjonowania poszczególnych modułów, sięgające $0,05^\circ$, do pomiaru położenia kąтового przegubów użyto enkodera absolutnego firmy iC-Haus, który poddany procedurze kompensacji błędów pomiaru, opisanej w [5], spełnia je. Dodatkowo całe ramię podlega wykonaniu poprodukcyjnej kalibracji kinematycznej [1].

Manipulator posiada rozproszoną strukturę układu sterowania. Sterownik główny manipulatora (Master), zbudowany jest w oparciu o komputer Raspberry Pi 3 z systemem Ubuntu Preempt RT, umożliwiającym zarządzanie pracą całego ramienia w czasie rzeczywistym. Poprzez sieć EtherCAT przesyła on komendy do sterowników modułów napędowych Slave oraz odbiera informacje dotyczące aktualnego stanu pracy (status, pozycja i prędkość kątowa, generowany moment napędowy, itp.). Algorytm sterowania implementowany na mikrokontrolerze STM32, poza zarządzaniem pracą modułu, realizuje kaskadowy algorytm sterowania napędem: pozycja-prędkość-moment. Algorytmy sterowania Slave-ów jak i Mastera wykonywane są z częstotliwością 1kHz. Szczegółowy opis obu algorytmów, jak również budowy modułów napędowych, można znaleźć w pracy [4].

2.3. Główny algorytm sterowania

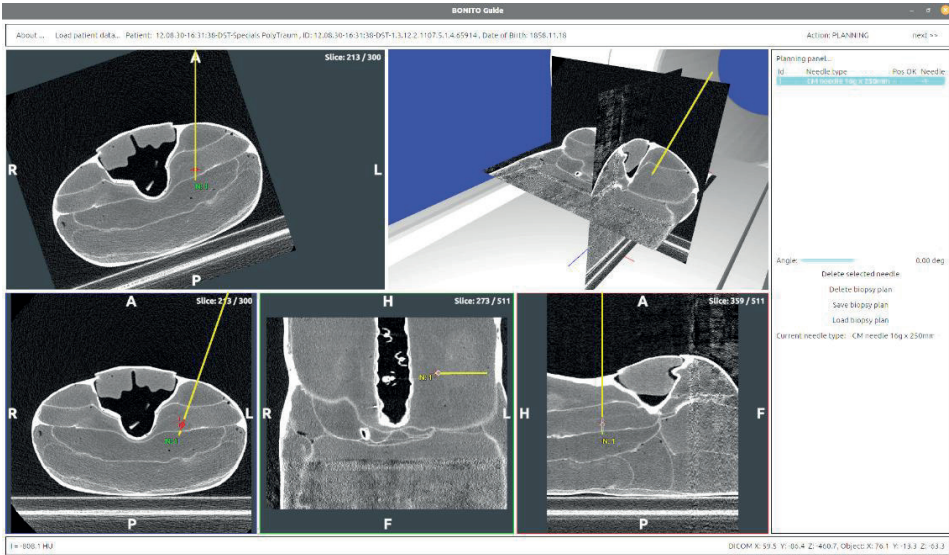
Dla całego systemu sterowania BONITO opracowano architekturę oprogramowania, której schemat blokowy prezentuje rys. 4. Wszystkie elementy tej architektury (z wyjątkiem oprogramowania sterującego bezpośrednio ramieniem) są implementowane fizycznie na komputerze stanowiącym sterownik główny BONITO.



Rys. 4. Architektura programowa systemu BONITO

Centralnym blokiem w architekturze programowej systemu BONITO jest blok *Zarządzanie biopsją*. Stanowi on maszynę stanów FSM zarządzającą sterownikiem głównym BONITO, w celu realizacji kolejnych zadań przewidzianych w diagramie sekwencji badania biopsyjnego, takich jak: pobranie i obróbka danych obrazowych pacjenta, określenie współrzędnych punktów: wbicia i docelowego igły biopsyjnej, odparkowanie i ubranie w rękaw ochronny ramienia, planowanie trajektorii ramienia, symulacja i wykonanie trajektorii do pozycji docelowej, zaplanowanie ruchu i przemieszczenie ramienia do pozycji gotowości, zdjęcie rękawa ochronnego oraz zaparkowanie ramienia. Przejścia pomiędzy kolejnymi stanami FSM są wyzwalane głównie akcjami podejmowanymi przez lekarza diagnostę za pośrednictwem interfejsu użytkownika i przełącznika nożnego, ale są również zależne od aktualnego statusu pracy manipulatora czy też innych bloków jak chociażby *Planowanie ruchu ramienia* czy *Symulacji i wykonanie ruchu ramienia*.

Interfejs użytkownika stanowi szereg ekranów wyposażonych w przyciski funkcyjne. Umożliwia on (za pośrednictwem bloku *Zapis/odbiór danych*) przeglądanie systemu plików i automatyczny pobór serii obrazów pochodzących z urządzenia obrazowania medycznego. Dodatkowo diagnosta ma możliwość importu danych do wewnętrznej bazy danych, zarówno z systemu plików, jak i z zewnętrznego serwera PACS (ang. picture archiving and communication system). Po załadowaniu danych i rekonstrukcji wolumenu w bloku *Przetwarzanie danych, rejestracja układów współrzędnych i aktualizacji modelu*, wyświetla przekroje wykonane w 3 prostopadłych kierunkach oraz trójwymiarowy obraz modelu środowiska biopsji (patrz rys. 5). Model środowiska zawiera trójwymiarowe modele wolumenu pacjenta, łoża, igieł biopsyjnych oraz manipulatora naprowadzającego.



Rys. 5. Widok okna definiowania pozycji i kierunku wbicia igieł interfejsu użytkownika

Przyciski funkcyjne pozwalają, w zależności od aktualnego stanu algorytmu *Zarządzania biopsją*, na przełączanie okien i wywoływanie akcji takich jak planowanie, symulacja i wykonanie trajektorii ramienia. Z użyciem myszy i przycisków klawiaturowych diagnosta ustala punkty reprezentujące miejsce wbicia igły O_e i punkt docelowy poboru tkanek O_t . Dodatkową funkcjonalnością interfejsu użytkownika jest możliwość wyświetlania płaszczyzny przekroju równoległej do zaplanowanej igły. Użytkownik może obracać obraz wokół osi tej igły, aby wizualnie ocenić, czy istnieje ryzyko uszkodzenia wrażliwych organów.

Ze względu na fakt, iż elementy obrazowania medycznego w tym współrzędne x_e i x_t punktów O_e i O_t wyrażane są w układzie współrzędnych wolumenu (przestrzeni obrazowej pacjenta) O_s , natomiast operacje wykonywane przez ramię w układzie bazowym manipulatora O_{mo} , należy dokonać ich transformacji wg. zależności:

$$x_{mo} = T_{mo}^w * T_w^s * x_s, \quad (1)$$

gdzie: $\mathbf{x}_{mo}$ - wektor współrzędnych punktu wyrażony w bazowym układzie manipulatora O_{mo} , T_{mo}^w - macierz transformacji z układu interfejsu manipulatora O_w do układu O_{mo} , T_w^s - macierz transformacji z układu obrazowania pacjenta O_s do układu O_w , $\mathbf{x}_s$ - wektor współrzędnych punktu wyrażony w układzie O_s . Macierz T_{mo}^w reprezentuje jedynie translację, wzdłuż łoża pacjenta, o odległość mierzoną enkoderem liniowym interfejsu mocującego manipulator do łoża. Ponieważ macierz transformacji T_w^s nie jest znana apriori, wyznaczana jest na drodze optymalizacji, biorąc pod uwagę, otrzymane na podstawie przetwarzania obrazów tomograficznych DICOM, współrzędne grupy czterech znaczników zarejestrowanych w obrazach CT.

Model ramienia, którego bazowy opis stworzony został w formacie URDF, jest wykorzystywany wraz z pozostałymi elementami modelu środowiska biopsji, przez algorytm detekcji kolizji do planowaniu trajektorii ruchu ramienia.

Algorytm planowania ruchu ramienia realizowany jest w przedstawionej na rys. 4 architekturze oprogramowania, przez blok o nazwie *Planowanie ruchu ramienia*.

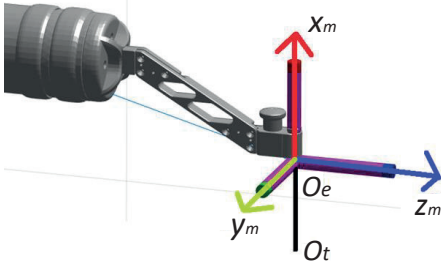
Algorytm ten realizuje, na żądanie, wyznaczanie trajektorii ruchu do:

- pozycji montażu / demontażu rękawa ochronnego,
- punktu wbicia igły - operacja złożona z dwóch etapów: najpierw dojazd do pozycji znajdującej się 10 cm nad miejscem wkłucia, a następnie dojazd po kierunku wprowadzania igły, bez zmiany orientacji, do punktu wbicia O_e ,
- pozycji gotowości po wbiciu igły - operacja złożona z dwóch etapów: najpierw odjazd po kierunku wbicia igły do pozycji 10 cm nad miejscem wkłucia, a następnie do pozycji gotowości.

O ile konfiguracja kinematyczna ramienia dla pozycji montażu i gotowości jest z góry określona, o tyle „poza” reprezentującą konfigurację wbijania igły musi zostać wyznaczona, a sam sposób jej wyznaczania został opisany w kolejnym rozdziale. Algorytm planowania trajektorii wykorzystuje bibliotekę OMPL, która stosuje algorytmy oparte na próbkowaniu, tj. losowym generowaniu stanów (pozycji przegubowych) robota. Jego celem działania jest odnalezienie ścieżki w wolnej przestrzeni stanów, dzięki której możliwe jest przejście pomiędzy pozą startową i końcową, unikając kolizji z otoczeniem (pacjentem, stołem) oraz samokolizji. Efektem wynikowym jest ciąg wartości kątowych dla poszczególnych napędów przegubowych w funkcji kolejnych chwil czasowych. Tak zaplanowane pozycje trajektorii ruchu są następnie za pośrednictwem bloku *Symulacja i wykonanie ruch ramienia* przekazywane poprzez zadawanie kolejnych pozycji przegubowych (w rygorze czasowym) dla celów wizualizacji symulacji ruchu ramienia albo do manipulatora celem wykonania rzeczywistego przemieszczenia. Wymiana danych ze sterownikiem głównym odbywa się poprzez bezpośrednie łącze komunikacyjne oparte o specjalnie zdefiniowane struktury danych (ramki) wyposażone w znaczniki czasu, których nośnikiem jest protokół komunikacyjny UDP IPv4. Na sterowniku Master ramienia uruchomiony jest server UDP, natomiast po stronie oprogramowania sterownika głównego BONITO, za pośrednictwem *Klienta UDP*, inicjowane jest nadawanie i odbiór danych (takich jak: zadane wartości pozycji przegubowych i trybu pracy ramienia, aktualny status pracy ramienia i aktualne pozycje przegubowe ramienia oraz aktualna wartość pozycji instalacji bazy).

3. Wyznaczanie wartości zadanych pozycji i orientacji

Algorytm planowania ruchu ramienia do miejsca biopsji wymaga podania docelowej konfiguracji kinematycznej (pozy) manipulatora. W tym celu najpierw wyznaczany jest kierunek wbijania igły, reprezentowany przez oś x_m (oznaczona jako x_7 na rys. 3) układu współrzędnych końcówki manipulatora O_m , który przedstawiono na rys. 6. Współrzędne wektora x_m obliczane są jako norma różnicy pomiędzy współrzędnymi punktów O_t (końcówki igły) i O_e (miejsca wbicia igły) wyrażonymi w bazowym układzie współrzędnych manipulatora O_{mo} .



Rys. 6. Układ współrzędnych końcówki manipulatora wraz z osią symetrii igły

Wyznaczenie wartości zadanej pozycji sprowadza się właściwie do przyjęcia za nie współrzędnych powstałych w wyniku przesunięcia współrzędnych punktu O_e o ustaloną wartość (np. 3-5mm) w kierunku osi x_m lub po prostu przyjęcia za nie współrzędnych punktu O_e (bez odsuwania od ciała pacjenta).

Wyznaczenie zadanej orientacji w postaci macierzy rotacji R_m poza jej pierwszą kolumną, która stanowi kolejne elementy wektora x_m :

$$x_m = [R_{11}, R_{21}, R_{31}]. \quad (2)$$

jest operacją bardziej złożoną. Ta złożoność wynika z faktu, iż na poprawność przeprowadzenia badania nie ma wpływu kąt obrotu igły wokół jej osi symetrii x_m i nie jest on tym samym definiowany przez lekarza. Niedeterministyczny charakter orientacji dwóch pozostałych osi końcówki manipulatora powoduje, że pozostałe dwie kolumny macierzy orientacji (wartości kąta obrotu α końcówki manipulatora wokół osi x_m) mogą przyjąć nieskończenie wiele kombinacji wartości. Należy więc spośród nich wybrać taką, która będzie najkorzystniejsza z punktu widzenia realizowalności, ergonomii i bezpieczeństwa badanego.

Pierwszym założeniem przyjętym w celu ograniczenia liczby przypadków jest nieujemność współczynnika R_{31} macierzy rotacji ($R_{31} \geq 0$). Oznacza to, że igła nigdy nie może być skierowana do góry (w kierunku od łóża do ciała pacjenta), co oczywiście powodowałoby najczęściej kolizję z ciałem pacjenta lub łóżem.

Uogólniając kształt ciała pacjenta, w okolicy brzuszno-piersiowej, do postaci półcylicylnicznej oraz wiedząc, że oś x_{mo} bazowego układu współrzędnych manipulatora jest skierowana prostopadłe do łóża (tj. osi symetrii tego półcylicindra), a przy tym równoległa do płaszczyzny łóża, można stwierdzić arbitralnie, iż

najlepszy pod względem ergonomicznym dostęp dla lekarza do wprowadzenia igły uzyskany zostanie, w bardzo dużej liczbie przypadków, gdy oś $\mathbf{z}_m$ końcówki manipulatora będzie prostopadła do płaszczyzny utworzonej przez osie $\mathbf{x}_{mo}$ i $\mathbf{x}_m$. W takiej sytuacji współrzędne osi $\mathbf{z}_m$ są wynikiem iloczynu wektorowego $\mathbf{x}_{mo} \times \mathbf{x}_m$, a po normalizacji będą stanowiły trzecią kolumnę macierzy rotacji R_m . Tak więc wartość współczynnika R_{13} , ze względu na prostopadłość osi $\mathbf{z}_m$ i $\mathbf{x}_{mo}$, wyniesie 0, a wartości współczynników R_{23} i R_{33} wyznaczone winny być następująco:

$$R_{23} = \frac{-R_{31}}{\sqrt{R_{21}^2 + R_{31}^2}}, \quad R_{33} = \frac{R_{21}}{\sqrt{R_{21}^2 + R_{31}^2}}. \quad (3)$$

Jeżeli igła jest skierowana równolegle do płaszczyzny stołu, tj. $R_{31} \approx 0$ i $R_{21} \approx 0$, wtedy współczynnikom z zależności (3) należy przypisać wartości odpowiednio: $R_{23} = -1$, $R_{33} = 0$. Ze względu na ortogonalność układu współrzędnych, współczynniki drugiej kolumny macierzy R_m , a tym samym osi $\mathbf{y}_m$, stanowią elementy normy z iloczynu wektorowego $\mathbf{z}_m \times \mathbf{x}_m$. Dodatkowo, gdy kąt odchylenia pomiędzy osiami $\mathbf{z}_m$ i $\mathbf{z}_{mo}$ jest większy od 45° , wówczas układ współrzędnych końcówki jest obracany o kąt o wartość z zakresu 0 do 90° , która jest dobierana proporcjonalnie do wielkości kąta odchylenia z zakresu 45 do 90° .

W celu dodatkowej poprawy ergonomii pracy lekarza i bezpieczeństwa badanego przeprowadzono, dla 50 przypadków wyboru miejsca i kierunku wprowadzania igły, eksperymenty optymalizujące poszukiwaną wartość kąta obrotu α , przyjmując za wartość początkową orientację wyznaczoną ww. metodą. Funkcja celu składała się z sumy dwóch podfunkcji. Pierwsza, związana z ergonomią pracy lekarza, stanowiła minimalną odległość elementu ramienia od osi symetrii stołka (wewnątrz którego ten element się znalazł) o promieniu podstawy 0.35 m i wysokości 0.7 m, którego wierzchołek umiejscowiony był w miejscu wbicia igły a jego oś symetrii skierowana była na rękkość mostka diagnosty wprowadzającego igłę. Druga podfunkcja stanowiła minimalną odległość elementów ramienia od ciała pacjenta, gdy ta odległość była mniejsza od 0.2 m. Wartości zwracane przez obie funkcje poddano normalizacji, poszukując kąta α , dla którego przyjmą one maksimum. W większości przypadków otrzymana wartość optymalnego kąta α nie odbiegała znacząco od 0 ($\pm 5^\circ$), a w innych przypadkach wartość funkcji celu nie była znacząco większa od tej dla $\alpha=0$. Potwierdza to, iż użycie ww. metody arbitralnej jest wystarczające, szczególnie w kontekście skrócenia czasu planowania ruchu z przekraczającego minutę (przy zastosowaniu optymalizacji) do pojedynczych sekund.

4. Podsumowanie

Przeprowadzone testy ewaluacyjne systemu BONITO dowiodły, że ma on szereg zalet. Dzięki znacznikom i rejestracji układów współrzędnych, znalezione miejsce biopsji można precyzyjnie oznaczyć a następnie nakierować na nie interfejs igłowy. Umożliwia to wprowadzenie igły bez pomyłki, w odpowiednie miejsce i pod

odpowiednim kątem, minimalizując ryzyko uszkodzenia tkanek oraz wielokrotnego powtarzania badania. Jednakże jest to silnie uzależnione od dokładności funkcjonowania poszczególnych elementów systemu. Kolejnym problemem wpływającym negatywnie na wyniki ewaluacji są wykonywane przez pacjenta ruchy pomimo jego unieruchomienia do łóża (np. oddychanie), które przez algorytm sterowania powinny być korygowane.

Literatura

- [1] K. Arent et al. Kinematic calibration of a lightweight manipulator for medical applications. *Bulletin of the Polish Academy of Sciences Technical Sciences*. vol. 27, no 6, 2024, s. 1-13.
- [2] W. J. Heerink, et al. Robotic versus freehand feedle positioning in CT-guided ablation of liver tumors: a randomized controlled trial. *Radiology*. vol. 290, 2019, s. 826–832.
- [3] A. Kuntz, et al. Autonomous medical needle steering in vivo. *Science Robotics*. vol. 8, Issue 82, 2023, s. 1-13.
- [4] A. Kurnicki, B. Stańczyk. Development of a modular light-weight manipulator for human-robot interaction in medical applications. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*. vol. 10, nr 4, 2020, s. 33-37.
- [5] A. Kurnicki, B. Stańczyk, B. Kania. Identyfikacja i kompensacja błędów pomiaru pozycji przegubowej w modułach napędowych ARIA. *Przegląd Elektrotechniczny*. vol. 100, nr 12, 2024, s. 93-96.
- [6] J. Liu, et al. Image registration in medical robotics and intelligent systems: fundamentals and applications. *Advanced Intelligent Systems*. vol. 1, Issue 6, 2019, s. 1-16.
- [7] T. Sato, et al. Robot-assisted CT-guided biopsy with an artificial intelligence-based needle-path generator: An experimental evaluation using a phantom model. *Journal of Vascular and Interventional Radiology*, 2025, s. 1-8.
- [8] D. Schreiber, et al. An open-source 7-axis, robotic platform to enable dexterous procedures within CT scanners. *IEEE Int Conf Intell Robot Syst*. 2019, s. 386–93.
- [9] F.J. Siepel, et al. Needle and Biopsy Robots: a Review. *Current Robotics Report*, vol. 2, 2021, s. 73–84.
- [10] B. Su, et al. Biopsy needle system with a steerable concentric tube and online monitoring of electrical resistivity and insertion forces. *IEEE Transactions on Biomedical Engineering*. vol. 68, 2021, s.1702–1713.

Control of a Biopsy Assistance System

The article presents the control system of a device used to assist in biopsy examination. This system was developed as part of the BONITO project, titled “Configurable Arm for Trajectory Stabilization of Needle Biopsy”. The article begins by outlining the structure of the entire system and discussing the key elements of its control system. It then addresses the issues related to determination of reference signals for position and orientation of manipulator end-effector in the control algorithm. Finally, the article discusses some of the results from its evaluation.

Indeks autorów

Arent, Krzysztof	205	Mazur, Alicja	105
Bartkowiak, Patryk	95	Michalek, Maciej	75
Chrobot, Joanna	39	Pazderski, Dariusz	95
Dindorf, Ryszard	185	Penar, Paweł	63
Domski, Wojciech	39	Podśędkowski, Lerszek	229
Dulęba, Ignacy	149	Przybylski, Mikołaj	75
Dutta, Vibekananda	117, 127	Rogaliński, Dawid	205
Dyba, Filip	105	Sęk, Kacper	229
Ekemeyong, Esther	117	Siemiątkowska, Barbara	29
Górski, Krzysztof	217	Singh, Gurjinder	15
Granosik, Grzegorz	137	Skwierczyński, Patryk	229
Grzegorzewski, Jacek	171	Sobański, Rafał	75
Herman, Przemysław	195	Stańczyk, Bartłomiej	249
Joon, Arpit	159	Strzałka, Jakub	137
Kacprzak, Piotr	229	Szeremeta, Mateusz	53, 63
Kamiński, Jakub	127	Szuster, Marcin	53, 63
Kobierska, Agnieszka	229	Takosoglu, Jakub	185
Kowalczyk, Wojciech	159	Woś, Piotr	185
Kreczmer, Bogdan	5	Zadarnowska, Katarzyna	205
Kuczmarska, Kamila	239	Zielińska, Teresa	117, 127
Kurnicki, Adam	249	Zieliński, Cezary	85
Kusz, Jakub	205	Zubrycki, Igor	239
Malewicz, Jakub	205, 217	Żebrecki, Julian	29
Małkowski, Mateusz	29		



**Wydawnictwa Politechniki Wrocławskiej
są do nabycia w sprzedaży wysyłkowej:
zamawianie.ksiazek@pwr.edu.pl
www.ksiegarnia.pwr.edu.pl**

ISBN 978-83-7493-910-2