

Autoreferat

na temat dorobku i osiągnięć naukowo-badawczych
w tym opis zasadniczego osiągnięcia naukowego –
jednotematycznego cyklu publikacji pod tytułem:
„Planowanie zadań manipulacji wykonywanych przez roboty”

dr inż. Wojciech Szynkiewicz

Politechnika Warszawska

Wydział Elektroniki i Technik Informatycznych

Instytut Automatyki i Informatyki Stosowanej

10.09.2015

Spis treści

	Strona
1 Ogólne informacje o habilitancie	4
1.1 Posiadane stopnie naukowe z podaniem nazwy, miejsca i roku ich uzyskania oraz tytuł rozprawy doktorskiej	4
1.2 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych	4
2 Wykaz publikacji będących podstawą wniosku habilitacyjnego	4
3 Opis zasadniczego osiągnięcia naukowego	6
3.1 Struktura osiągnięcia naukowego	6
3.2 Zakres i cel naukowy badań	6
3.3 Podsumowanie uzyskanych wyników	8
4 Omówienie uzyskanych wyników prac badawczych	8
4.1 Planowanie manipulacji dwuręcznej robotów	9
4.1.1 Umiejętności manipulacyjne [H6, H11]	9
4.1.2 Planowanie ścieżek dla dwóch robotów manipulujących pojedynczym obiektem [H7]	13
4.2 Planowanie rozmieszczenia robotów-podpór rekonfigurowalnego systemu mocowań [H1–H5, H8]	19
4.2.1 Budowa rekonfigurowalnych mocowań [H2, H5]	21
4.2.2 Planowanie rozmieszczenia rekonfigurowalnych agentów-podpór jako zadanie optymalizacji z ograniczeniami [H8, H9]	23
4.2.3 Planowanie rozmieszczania ruchomych podpór jako problem spełnienia ograniczeń [H1–H4]	25
4.3 Implementacje algorytmów planowania i sterowania z użyciem programowej struktury ramowej MRROC++ [H10, H11]	33
4.3.1 Struktura ramowa dla tworzenia oprogramowania sterującego robotami	33
4.3.2 Implementacja <i>planera</i> i układu sterowania dla układania kostki Rubika [H10, H11]	33
4.3.3 Planowanie ścieżki dla systemu z zamkniętym łańcuchem kinematycznym [H7]	34
4.3.4 Układ sterowania rekonfigurowalnych robotów mocujących [H5]	34
5 Omówienie pozostałej działalności naukowo-badawczej	37
5.1 Autorstwo lub współautorstwo publikacji naukowych innych niż przedstawione w punkcie 2	37
5.1.1 Synteza chwytów i ich realizacja za pomocą chwytaków wielopalczastych	37
5.1.2 Projektowanie i implementacja wysokopoziomowych sterowników o strukturze otwartej dla systemów jedno- i wielorobotowych	38
5.1.3 Planowanie trajektorii ruchu dla współpracujących manipulatorów robotów	39
5.1.4 Sterowanie i autonomiczna nawigacja kołowych robotów mobilnych	40

5.2	Czynny udział w konferencjach i seminariach naukowych	42
5.3	Kierowanie projektami badawczymi lub udział w takich projektach	46
5.3.1	Udział w projektach	46
5.3.2	Kierowanie projektami	48
5.4	Nagrody za osiągnięcia naukowe	49
5.5	Zbiorcze wskaźniki bibliometryczne	49

1 Ogólne informacje o habilitancie

1.1 Posiadane stopnie naukowe z podaniem nazwy, miejsca i roku ich uzyskania oraz tytuł rozprawy doktorskiej

- Doktor nauk technicznych w dziedzinie Automatyka i Robotyka. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych. Tytuł rozprawy: *Algorytmy planowania trajektorii w systemach wielorobotowych*, 1996.
- Magister inżynier w dziedzinie Automatyka i Robotyka. Politechnika Warszawska, Wydział Elektroniki. Tytuł rozprawy: *Sterowanie manipulatorem robota przy pomocy algorytmów ruchu ślizgowego*, 1985.

1.2 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

1985-obecnie	Politechnika Warszawska Wydział Elektroniki i Technik Informacyjnych Instytut Automatyki i Informatyki Stosowanej – adiunkt (od 1996) – pracownik naukowo-badawczy (od 1985)
1999-2003	Centrum Automatyki i Technik Informacyjno-Decyzyjnych (CATID) – zastępca dyrektora i sekretarz Rady Naukowej CATID

2 Wykaz publikacji będących podstawą wniosku habilitacyjnego

Poniższe publikacje wchodzą w skład osiągnięcia, o którym mowa w art. 16 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki.

Jest nim jednotematyczny cykl publikacji (obejmujący 11 pozycji) zatytułowany:

„Planowanie zadań manipulacji wykonywanych przez roboty”.

Publikacje wchodzące w skład zasadniczego osiągnięcia naukowego wnioskodawcy obejmują okres lat 2006–2014 i powstały po uzyskaniu stopnia naukowego doktora nauk technicznych. Pragnę zauważyć, że prace mają charakter zarówno autorski jak i współautorski. Jako członek zespołu naukowego kierowanego przez prof. Cezarego Zielińskiego specjalizuję się w zadaniach planowania działań autonomicznych robotów, wykonując prace badawcze w tym zakresie samodzielnie względnie kierując nimi.

Przy każdej pozycji podany jest *Impact Factor* (IF) czasopisma (jeśli czasopismo takowe ma) według Web of Science. Ilościowy oraz jakościowy udział współautorów w przygotowaniu publikacji określony i poświadczony podpisami został przedstawiony w załączniku Z5. Pozostały dorobek wnioskodawcy jest przedstawiony w osobnych załącznikach.

Wybrany cykl publikacji:

- [H1] W. Kasprzak, **W. Szynkiewicz**, D. Zlatanov, and T. Zielińska. A hierarchical CSP search for path planning of cooperating self-reconfigurable mobile fixtures. *Engineering Applications of Artificial Intelligence*, 34:85–98, Sep. 2014. (IF=2.207).
- [H2] **W. Szynkiewicz**. Action planning for multi-robot-based reconfigurable fixture. *International Journal of Ad Hoc and Ubiquitous Computing*, 17(2-3):73–81, 2014. (IF=0.554).
- [H3] T. Zielińska, W. Kasprzak, **W. Szynkiewicz**, and C. Zieliński. Path planning for robotized mobile supports. *Mechanism and Machine Theory*, 78:51–64, Aug. 2014. (IF=1.66).
- [H4] W. Kasprzak, D. Zlatanov, **W. Szynkiewicz**, and T. Zielińska. Task planning for cooperating self-reconfigurable mobile fixtures. *Journal of Advanced Manufacturing Technology*, 69(9-12):2555–2568, 2013. (IF=1.779).
- [H5] C. Zieliński, W. Kasprzak, T. Kornuta, **W. Szynkiewicz**, P. Trojanek, M. Wałęcki, T. Winiarski, and T. Zielińska. Control and Programming of a Multi-Robot-Based Reconfigurable Fixture. *Industrial Robot: An International Journal*, 40(4):329–336, 2013. (IF=0.622).
- [H6] **W. Szynkiewicz**. Skill-based bimanual manipulation planning. *Journal of Telecommunications and Information Technology*, (4):54–62, 2012.
- [H7] **W. Szynkiewicz** and J. Błaszczuk. Optimization-based approach to path planning for closed-chain robot systems. *International Journal of Applied Mathematics and Computer Science*, 21(4):659–670, 2011. (IF=0.487).
- [H8] **W. Szynkiewicz**, T. Zielińska, and W. Kasprzak. Robotized Machining of a Big Work Pieces: Localization of Supporting Heads. *Frontiers of Mechanical Engineering in China*, 5(4):357–369, 2010.
- [H9] **W. Szynkiewicz**. Planning system for multi-agent based reconfigurable fixtures. *Journal of Telecommunications and Information Technology*, (3):71–76, 2010.
- [H10] C. Zieliński, **W. Szynkiewicz**, T. Winiarski, M. Staniak, and T. Czajewski, W. Kornuta. Rubik’s cube as a benchmark validating MRROC++ as an implementation tool for service robot control systems. *Industrial Robot: An International Journal*, 34(5):368–375, 2007. (IF=0.4).
- [H11] **W. Szynkiewicz**, C. Zieliński, W. Czajewski, and T. Winiarski. Control architecture for sensor-based two-handed manipulation. In *CISM Courses and Lectures - RoManSy 16*, volume 487, pages 237–244. Springer, Wien, New York, 2006.

Łączny IF czasopism, w których opublikowano powyższe prace wynosi 7.749.

3 Opis zasadniczego osiągnięcia naukowego

3.1 Struktura osiągnięcia naukowego

Tematyka cyklu publikacji będących podstawą wniosku habilitacyjnego dotyczy metod planowania zadań manipulacji wykonywanych przez współpracujące ze sobą roboty. W ogólnym przypadku do manipulacji zalicza się następujące grupy czynności: *manipulację chwytną (zręczną)*, *unieruchamianie (mocowanie)* obiektów oraz *aktywną eksplorację/badanie* otoczenia. Podział ten, oczywiście, nie wyczerpuje wszystkich możliwych rodzajów manipulacji. Tematyka moich prac wchodzących w skład przedstawianego cyklu koncentruje się w dwóch pierwszych grupach czynności.

1. Przedmiotem badań przedstawionych w pracach [H6,H7,H10,H11] są zagadnienia związane z manipulacją dwuręczną robotów obiektem, zarówno sztywnym, jak i z wewnętrznymi stopniami swobody.
2. Prace [H1–H5, H8, H9] dotyczą planowania rozmieszczania oraz sterowania zespołem rekonfigurowalnych robotów-podpór wchodzących w skład elastycznego systemu mocującego. Zagadnienie planowania rozmieszczania elementów systemu mocowań (a więc jego rekonfiguracji) w celu unieruchamiania obiektu jest ściśle powiązane z syntezą chwytów dla robotów manipulacyjnych. Proponowane—w wyżej wymienionych pracach—ogólne podejście polega na sformułowaniu zadania planowania manipulacji jako problemu spełnienia ograniczeń (więzów). Do jego rozwiązania wykorzystano różne metody, w tym algorytmy programowania nieliniowego oraz techniki sztucznej inteligencji polegające na przeszukiwaniu drzew decyzyjnych.
3. Oprogramowanie układów sterowania dla systemów robotycznych realizujących przykładowe zadania manipulacji dwuręcznej oraz rekonfigurowalnych robotów-podpór mocujących detale poddawane obróbce, zostało zbudowane na bazie programowej struktury ramowej MRROC++ (*Multi-Robot Research-Oriented Control utilizing C++*) [H5, H10, H11]. Środowisko MRROC++ składa się z biblioteki modułów, wzorców ich wykorzystania oraz właściwej struktury ramowej, i umożliwia tworzenie wysokopoziomowych sterowników robotów.

Istotnym elementem prac badawczych była zarówno symulacyjna (numeryczna), jak i eksperymentalna (doświadczalna) weryfikacja poprawności i efektywności działania opracowanych metod planowania manipulacji.

3.2 Zakres i cel naukowy badań

Zadanie manipulacji obiektem przez roboty

Wykonanie wielu codziennych czynności wymaga manipulacji różnymi obiektami. Ludzie uczą się tej umiejętności przez wiele lat, poczynając od okresu niemowlęctwa, gdy dziecko próbuje dosięgnąć i schwycić różne przedmioty, kończąc na bardzo złożonych czynnościach manipulacyjnych wymagających użycia skomplikowanych narzędzi. Próba zrozumienia oraz dążenie do

odtworzenia, choćby w przybliżeniu, zdolności manipulacyjnych człowieka przez roboty jest jednym z podstawowych elementów, koniecznych do skonstruowania robotów wykonujących różnorodne czynności manipulacyjne w naturalnym środowisku człowieka. Jednakże, pomimo wielu prac badawczych i rozwoju technologii, nadal istnieje duża różnica pomiędzy zdolnościami manipulacyjnymi ludzi i autonomicznych robotów, szczególnie w niestrukturalnych środowiskach naturalnych.

W załączonych pracach jest rozważane zagadnienie planowania zadań manipulacji realizowanych przez roboty, w szczególności przez pary robotów względnie roboty dwuręczne. Opis dowolnego zadania manipulacji jest generalnie złożonym problemem. Obiekt w zadaniu manipulacji jest centralnym elementem systemu robotycznego, którego ruch i wywierane nań siły są przedmiotem analizy przy planowaniu manipulacji. Planowanie manipulacji wymaga znajomości modeli robota, obiektów manipulacji oraz środowiska (otoczenia). Sposób reprezentacji zadania musi być wystarczająco uniwersalny, aby zapewnić opis możliwie szerokiego zakresu zadań a jednocześnie umożliwiać jego rozwiązanie. Ponadto istnieje luka semantyczna między prostymi poleceniami wysokiego poziomu np. „zdejmij pokrywkę pudełka”, a ich opisem geometrycznym i kinematycznym na poziomie sterownika robota [3, 23].

Planowanie zadań i akcji

Ludzie wykonując złożone zadania manipulacyjne dzielą je na szereg prostszych czynności. Efekty takiego podziału można zaobserwować w postaci zmieniających się reakcji na bodźce w trakcie manipulacji [7, 12]. W koncepcji tzw. prymitywów motorycznych (*motor primitives*) również zakłada się, że człowiek (ale także zwierzęta) wykonuje wiele czynności składając je z prostych czynności z dyskretnego zbioru elementarnych zachowań [8, 20]. Prymitywy motoryczne są elastycznymi akcjami, które mogą być łatwo przystosowane do nowych sytuacji. Wystarczy zatem kilka prymitywów, aby możliwe było wykonanie różnych zadań. Istotne znaczenie dla sposobu określenia umiejętności manipulacyjnych człowieka ma sformułowana przez Jamesa J. Gibsona teoria afordancji (*affordances*) [10]. Jednym z jej kluczowych założeń jest sposób klasyfikacji obiektów manipulacji według czynności (akcji), które można z nimi wykonać.

Planowanie ścieżek ruchu dla zamkniętych łańcuchów kinematycznych

W przypadku, gdy chwytaki dwóch robotów (lub robota dwuramiennego) uchwycą i manipulują pojedynczym obiektem, powstaje wieloprzegubowy mechanizm zawierający zamknięty łańcuch kinematyczny. W obszarze robotyki można znaleźć więcej przykładów systemów, w których występują pętle kinematyczne. Roboty równoległe są strukturami mechanicznymi zawierającymi zamknięte łańcuchy [18]. W maszynach krocących zamknięty łańcuch tworzy się w chwili kontaktu nóg z podłożem [25]. Poza obszarem robotyki zamknięte łańcuchy są przedmiotem badań m.in. w grafice komputerowej, wirtualnym prototypowaniu, a także w chemii i biologii molekularnej [6, 13, 22]. Planowanie ruchu dla zamkniętego łańcucha jest wykonywane w przestrzeni konfiguracyjnej o dużym wymiarze i zazwyczaj przy skomplikowanych ograniczeniach. Więzy ruchu wynikające z warunku zamknięcia pętli kinematycznej można przedstawić w postaci układu nieliniowych równań wielomianowych. Rozwiązanie takiego układu równań jest trudnym zadaniem o dużej złożoności obliczeniowej. Nie opracowano, jak dotychczas, dostatecznie ogólnej i efektywnej metody rozwiązywania tego typu równań, aby mogła ona znaleźć

praktyczne zastosowanie w algorytmach planowaniu ruchu.

Planowanie rozmieszczenia i rekonfiguracji mobilnych robotów-podpór

Pomimo, iż dominującym obecnie obszarem prac badawczych w dziedzinie robotyki są roboty usługowe i terenowe, to w przemyśle nadal istnieją nierozwiązane istotne problemy wymagające zastosowania robotów. Jednym z takich przykładów jest przemysł lotniczy, gdzie przy obróbce mechanicznej elementów poszycia kadłubów samolotów są stosowane kosztowne formy mocujące (podpierające). Elementy te są wykonane z cienkich blach i mają stosunkowo znaczne rozmiary. Obecnie zazwyczaj dla każdego detalu jest stosowana odrębna forma podpierająca, precyzyjnie dopasowana do jego kształtów. Rzadziej stosowane są zestawy ręcznie konfigurowanych podpór. Pierwsze rozwiązanie wymaga wielu precyzyjnie dopasowanych form, a więc jest bardzo kosztowne, natomiast w drugim przypadku przy zmianie rodzaju obrabianego detalu każdorazowo trzeba precyzyjnie ustawić podpory, co jest czasochłonne, a ponieważ wykonywane jest na stanowisku obróbczym, czyni je przez ten czas bezproduktywnym.

3.3 Podsumowanie uzyskanych wyników

Proponowane—w pracach wchodzących w skład niniejszego cyklu—metody planowania planowania oraz systemy sterowania robotów umożliwiają przynajmniej częściowe rozwiązanie wyżej wymienionych problemów. Do istotniejszych osiągniętych wyników tych prac należy zaliczyć:

- autorską strategię planowania manipulacji polegającą na dekompozycji złożonego zadania na zestaw elementarnych umiejętności manipulacyjnych wraz z przedstawieniem jednolitego sposobu ich opisu (→ omówienie w pkt. 4.1.1);
- opracowanie autorskiej, uniwersalnej metody planowania ścieżek ruchu dla zamkniętych łańcuchów kinematycznych w manipulacji dwuręcznej robotów (→ omówienie w pkt. 4.1.2);
- opracowanie, implementację oraz weryfikację eksperymentalną systemu planowania i sterowania zespołem rekonfigurowalnych zrobotyzowanych podpór (→ omówienie w pkt. 4.2);
- opracowanie programowej struktury ramowej MRROC++ umożliwiającej tworzenie sterowników dla systemów jedno- i wielorobotowych wykonujących różnorodne zadania (→ omówienie w pkt. 4.3).

4 Omówienie uzyskanych wyników prac badawczych

Przedstawione poniżej komentarze dotyczą najistotniejszych wyników omawianych w pracach [H1–H11] jednotematycznego cyklu publikacji będącego podstawą wniosku habilitacyjnego. Poza cytowaniem stosownych prac należących do jednotematycznego cyklu będę również odwołał się do moich prac [Hu1, Hu2, Hu3, Hu4, Hu5, Hu6, Hu7, Hu8, Hu9, Hu10, Hu11], autorskich bądź współautorskich, powiązanych tematycznie z pracami cyklu. Uzyskane wyniki będę omawiał przyjmując konwencję tematyczną. W pierwszej kolejności przedstawię wyniki prac dotyczących planowania i realizacji manipulacji dwuręcznej (pkt. 4.1), w tym planowania ścieżek dla mechanizmów wieloprzegubowych zawierających zamknięte łańcuchy kinematyczne (pkt. 4.1.2).

Następnie omówię najważniejsze wyniki prac dotyczących planowania rozmieszczania rekonfigurowalnych robotów-podpór wchodzących w skład elastycznego systemu mocowań (pkt. 4.2). Na koniec przedstawię strukturę ramową MRROC++ i tworzenie sterowników z jej użyciem oraz opis implementacji algorytmów planowania i realizacji zadań manipulacji (pkt. 4.3).

4.1 Planowanie manipulacji dwuręcznej robotów

W obliczu szybkiego rozwoju robotów usługowych, a w szczególności robotów humanoidalnych i manipulatorów mobilnych, które wykonują czynności wymagające manipulacji przedmiotami codziennego użytku, konieczne jest opracowanie efektywnych algorytmów planowania manipulacji. Algorytm planowania manipulacji jest odpowiedzialny za generację sekwencji akcji przeprowadzającej robota i obiekty, którymi on manipuluje ze stanu początkowego do zadanego stanu końcowego.

W pracach [H6, H7, H10, H11] jest rozważane zagadnienie planowania manipulacji dla robotów wykonujących czynności wymagające ich ścisłej współpracy. Wspólna manipulacja pojedynczym obiektem przez dwa manipulatory lub robota dwuręcznego jest typowym przykładem takiego zadania [Hu3]. Drugim przykładem manipulacji obiektem z wewnętrznymi stopniami swobody jest układanie kostki Rubika przez dwa roboty [Hu4, Hu9, Hu10]. Dla szerokiej klasy zadań usługowych zastosowanie robotów dwuręcznych może przyspieszyć wykonanie lub być wręcz niezbędne do realizacji zadania. Wykorzystanie dwóch rąk poszerza klasę obiektów, którymi można manipulować, o przedmioty długie, elastyczne, o nieregularnych kształtach lub złożone, z wewnętrznymi stopniami swobody (np. kostka Rubika, słoik z pokrywką, nożyczki). Dla ściśle współpracujących robotów, gdy wymagana jest ścisła synchronizacja ruchów robotów, już znalezienie dopuszczalnych trajektorii ruchu oraz chwytów jest złożonym problemem obliczeniowym.

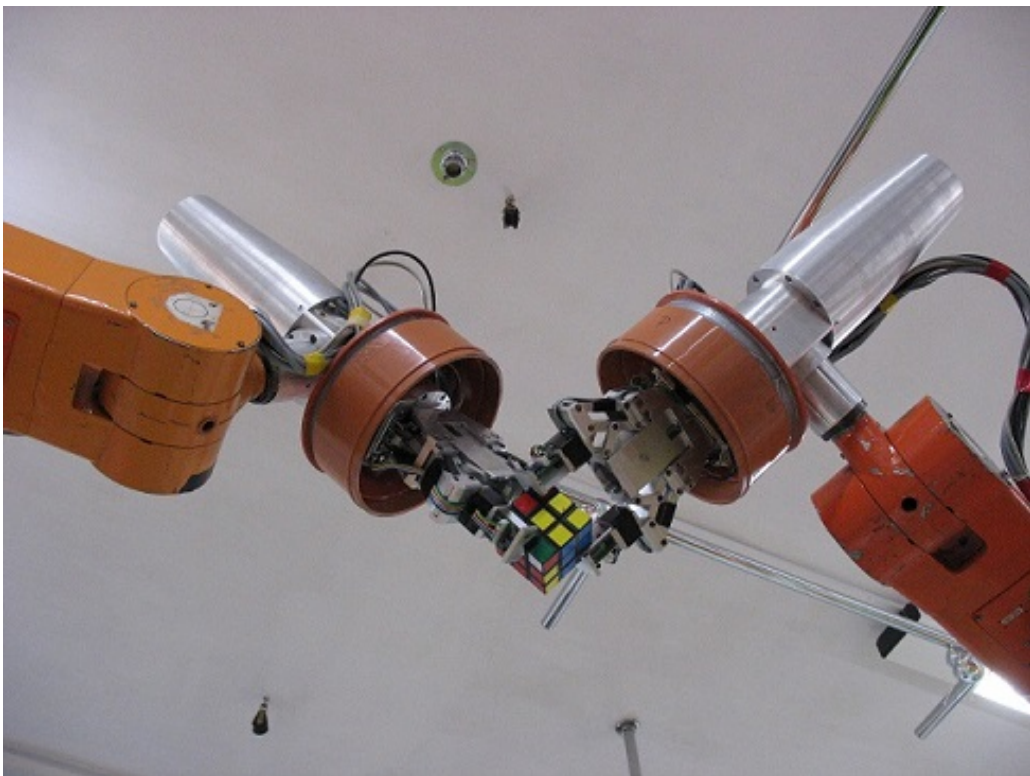
4.1.1 Umiejętności manipulacyjne [H6, H11]

W artykułach [H6, H11] przedstawiono koncepcję planowania manipulacji polegającą na dekompozycji złożonego zadania na zestaw (sekwencyjnych i/lub równoległych) prostych czynności (np. ruchów, chwytów) zwanych elementarnymi umiejętnościami manipulacyjnymi (*basic manipulation skills*). Wstępna wersja tego podejścia została zaprezentowana w pracy [Hu4]. Umiejętności manipulacyjne mogą być utożsamiane z pewną klasą zachowań. W paradygmacie sterowania behawioralnego zachowania określają czynności robota.

Planowanie złożonych zadań manipulacji wymaga połączenia planowania symbolicznego i geometrycznego. Zgodnie z proponowaną koncepcją system planowania ma wielowarstwową strukturę hybrydową składającą się z warstwy symbolicznego planowania zadania, warstwy planowania geometrycznego (planowania ruchu, syntezy chwytów) oraz warstwy pośredniej, w której skład wchodzi umiejętność manipulacyjna. Zadanie jest dekomponowane na sekwencję umiejętności manipulacyjnych, których wykonanie prowadzi do osiągnięcia określonego celu. Umiejętność manipulacyjna jest pewną abstrakcją zbioru podstawowych czynności, czyli elementarnych umiejętności manipulacyjnych. Każda elementarna umiejętność jest realizowana za pomocą jednej strategii (prawa) sterowania i ma zestaw parametrów, których wartości są obliczane dla planu wyznaczonego dla konkretnego zadania. Elementarne umiejętności są atomowymi akcjami dla warstwy planowania symbolicznego, czyli elementarnymi symbolami, które

są użyte do opisu akcji i zadań. Zastosowanie umiejętności ma na celu zwiększenie autonomiczności, elastyczności i odporności na błędy modelowania oraz zmienność i niepewność w zadaniu. Elementarne umiejętności manipulacyjne służą do osiągnięcia konkretnego celu poprzez realizację takich czynności jak generacja trajektorii ruchu, uchwycenie i zwolnienie obiektu, itp. Każda elementarna umiejętność jest implementowana jako parametryzowalny sterownik (program sterujący) z bezpośrednim dostępem do receptorów i efektorów. W ogólnym przypadku, każda elementarna umiejętność jest predefiniowana (zaprogramowana) lub może być nauczona. W rozwiązaniu przedstawionym w pracy [H6] wszystkie umiejętności elementarne zostały wcześniej zdefiniowane i zaimplementowane w postaci odpowiednich programów sterujących.

Proponowane podejście zostało zastosowane i zweryfikowane w zadaniu układania kostki Rubika będącego przykładem złożonej manipulacji wykonywanej przez dwuręcznego robota usługowego [H6]. Jako fizyczny model takiego robota wykorzystano dwa współpracujące manipulatory IRp-6 (każdy z dodanym 6-tym stopniem swobody), wyposażone w specjalizowane chwytaki [Hu4] (rys. 1). Ułożenie kostki Rubika wymaga wykonania szeregu czynności takich



Rys. 1: Manipulacja kostką Rubika

jak: lokalizacja kostki, określenie jej aktualnego stanu na podstawie informacji wizyjnej, obliczenie sekwencji obrotów ścianek kostki prowadzących do stanu docelowego, a następnie wykonanie sekwencji odpowiednich chwytów i ruchów w celu jej ułożenia.

Podsumowanie uzyskanych wyników:

1. Dekompozycja problemu – definicja elementarnych umiejętności robota – symbolicznych akcji.
2. Strategie przeszukiwania przestrzeni stanów – IDA* dla znalezienia rozwiązania (planu układania kostki Rubika).

3. Wyrażenie elementarnych symbolicznych akcji jako zadań sterowania pozycyjnego i pozycyjno-siłowego.

Opis szczegółowy

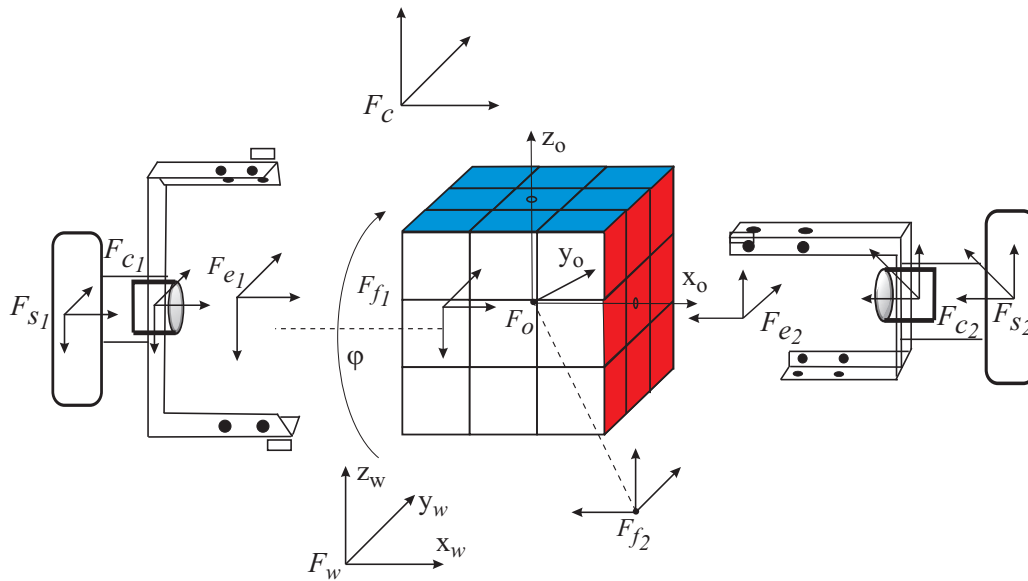
Standardowa kostka Rubika $3 \times 3 \times 3$ jest zbudowana z 27 małych sześciątów (w tym 26 widocznych). Stan kostki jednoznacznie określa podanie położenia i orientacji 20 sześciątów. Liczba stanów osiągalnych z każdego stanu wynosi około $n \approx 4.32 \times 10^{19}$. Dla znalezienia optymalnego (lub nawet sub-optymalnego) rozwiązania tego zadania jest konieczne zastosowanie efektywnego algorytmu przeszukiwania drzewa stanów. W tym celu wykorzystaliśmy dwa algorytmy. W pierwszym z nich, Iterative-Deepening-A* (IDA*), wykorzystuje się heurystyczną estymatę kosztu dojścia do stanu docelowego bazując na obliczonej wcześniej bazie wzorców [14]. Jeśli rozwiązanie optymalne wymaga mniej niż 16 ruchów, algorytm ten umożliwia jego obliczenie w akceptowalnym czasie. Natomiast, gdy rozwiązanie wymaga więcej niż 16 ruchów stosowany jest dwufazowy algorytm Kociemby, który oblicza rozwiązanie suboptymalne, ale w znacznie krótszym czasie.

Zaimplementowana wersja algorytmu IDA* pozwala na znalezienie rozwiązania optymalnego w czasie od kilku do kilkunastu sekund (dla obliczeń wykonywanych na komputerze typu PC z procesorem 3 GHz i pojemnością pamięci operacyjnej 512 MB) dla przypadków wymagających co najwyżej 16 ruchów. Jeśli rozwiązanie wymaga większej liczby ruchów niż 16, wówczas poszukiwanie rozwiązania optymalnego jest zbyt czasochłonne i jest wyznaczane rozwiązanie suboptymalne, które zazwyczaj nie przekracza 20–21 ruchów. W wyniku działania algorytmu, dla danego stanu kostki, wyznacza się sekwencję obrotów warstw kostki wymaganych do jej ułożenia.

Do wykonania zadania układania kostki Rubika zdefiniowano zestaw elementarnych umiejętności: *Localize* – lokalizacji kostki, *Reach* – generacji ścieżki podejścia chwytaków w celu uchwycenia kostki, *Grasp* – uchwycenia kostki (przez każdy z chwytaków), *Turn* – obrotu ścianki za pomocą jednego z chwytaków, *Release* – zwolnienia chwytu. Właściwa czynność obracania warstw kostki jest przykładem skoordynowanej manipulacji dwuręcznej pojedynczym obiektem z wewnętrznymi stopniami swobody. Plan wykonania zadania został zaprogramowany w postaci sekwencji elementarnych umiejętności manipulacyjnych.

Do zgrubnej lokalizacji kostki została zdefiniowana umiejętność *Localize*, w której źródłem informacji jest stacjonarna kamera CCD [Hu1]. Na podstawie analizy obrazu jest estymowana bieżąca pozycja układu kostki F_o względem układu związanego z stacjonarną kamerą F_c (rys.2). Dzięki znajomości pozycji układu kamery w globalnym układzie odniesienia F_w (kalibracja układu kamera-robot) można obliczyć pozycję kostki w układzie F_w . Wstępna lokalizacja kostki obserwowanej przez zewnętrzną kamerę pozwala na wykonanie ruchu chwytaka w pobliżu kostki z dokładnością od kilku do kilkunastu milimetrów (w zależności m.in. od jakości obrazu oraz odległości i kąta obserwacji kostki). Układy współrzędnych F_{e_i} , $i = 1, 2$ są związane z chwytakami, a układy F_{s_i} i F_{c_i} odpowiednio z czujnikami sił/momentów zamontowanymi w końcówkach manipulatorów oraz kamer umieszczonych w chwytakach. Wzajemne pozycje układów są opisane za pomocą macierzy przekształcenia jednorodnego ${}^i_jT \in \mathbb{SE}(3)$ o wymiarze (4×4) , gdzie $\mathbb{SE}(3)$ oznacza specjalną grupę euklidesową ruchów ciała sztywnego [19].

Po zgrubnej lokalizacji kostki w globalnym układzie współrzędnych, w umiejętności *Reach*



Rys. 2: Układy współrzędnych w zadaniu układania kostki Rubika

jest obliczana bezpieczna (bezkolizyjna) ścieżka dojścia do obiektu, przy założeniu znajomości położenia i kształtu przeszkód. Obraz z kamer umieszczonych w chwytakach jest wykorzystywany w umiejętności *Reach* (realizującej serwomechanizm wizyjny) do obliczania bieżącej pozycji kostki, a następnie do ustawienia chwytaka względem kostki w pozycji umożliwiającej poprawny chwyt. Położenie i orientacja chwytaka są na bieżąco aktualizowane i w każdej iteracji chwytak zbliża się do kostki zachowując odpowiednią orientację (prostopadle do ścianki i tak, aby szczęki chwytaka znalazły się wzdłuż krawędzi kostki). Błędy estymacji położenia i orientacji kostki nie przekraczają pojedynczych milimetrów i stopni, co jest w zupełności wystarczające do naprowadzenia chwytaka w taki sposób, aby mógł uchwycić kostkę. Istotnym zadaniem jest wybór takiej konfiguracji obu ramion, kiedy tworzą one wraz z uchwyconą kostką zamknięty łańcuch kinematyczny, która umożliwia wykonanie całego obrotu o zadany kąt bez konieczności zmiany uchwytu (przechwytywania).

Umiejętność *Grasp* realizuje algorytm syntezy chwytu oraz wykonanie chwytu. Przy syntezy chwytu wykorzystuje się znajomość modelu geometrycznego chwytaka. W rozważanym przypadku, ze względu na specyficzną konstrukcję chwytaków (chwytaki dwupalczaste o odpowiednio ukształtowanych szczękach równoległych) oraz znajomość modelu obiektu (kształt i rozmiary kostki są znane), problem syntezy chwytu sprowadza się do wyboru właściwej chwytaki – miejsca uchwycenia kostki. Dla każdej z 6 ścian jest potencjalnie możliwych 8 konfiguracji chwytów (określonych przez pozycję układu chwytaka w układzie kostki). Do opisu każdego chwytu są definiowane dwa układy współrzędnych: układ F_{f_1} dla uchwycenia przez chwytak pojedynczej warstwy kostki oraz F_{f_2} dla uchwycenia dwóch warstw. Zadane pozycje chwytu w układzie F_o pojedynczej warstwy i dwóch warstw są określone odpowiednio za pomocą macierzy ${}^o_{f_1}T$ oraz ${}^o_{f_2}T$. Zatem synteza chwytu nominalnego polega na określeniu pozycji układów F_{f_i} , $i = 1, 2$ względem globalnego układu współrzędnych F_w .

$${}^w_{f_1}T = {}^w_oT {}^o_{f_1}T; \quad {}^w_{f_2}T = {}^w_oT {}^o_{f_2}T \quad (1)$$

Warunki dopuszczalności chwytów mają charakter czysto geometryczny i są określone następu-

jąco:

$${}^w T_{f_1} = {}^w T_{b_1} {}^{b_1} T_{e_1}(\mathbf{q}_1) {}^{e_1} T_{f_1}^o T; \quad {}^w T_{f_2} = {}^w T_{b_2} {}^{b_2} T_{e_2}(\mathbf{q}_2) {}^{e_2} T_{f_2}^o T, \quad (2)$$

albo

$${}^w T_{f_1} = {}^w T_{b_2} {}^{b_2} T_{e_2}(\mathbf{q}_2) {}^{e_2} T_{f_1}^o T; \quad {}^w T_{f_2} = {}^w T_{b_1} {}^{b_1} T_{e_1}(\mathbf{q}_1) {}^{e_1} T_{f_2}^o T, \quad (3)$$

gdzie ${}^w T_{b_i}$, $i = 1, 2$ macierz określająca pozycję układu bazowego F_{b_i} robota w układzie F_w . Macierze ${}^{b_i} T_{e_i}(\mathbf{q}_i)$, $i = 1, 2$ reprezentują kinematykę prostą i -tego ramienia, a \mathbf{q}_i wektor jego współrzędnych konfiguracyjnych.

Uchwycenie kostki przez dwa chwytaki pozwala na rozpoczęcie właściwej manipulacji realizowanej przez umiejętność *Turn*. Wyznaczona sekwencja obrotów warstw kostki może być opisana w układzie F_o związanym z kostką, jako ciąg kolejnych obrotów wokół wersorów $\hat{\mathbf{x}}_o, \hat{\mathbf{y}}_o, \hat{\mathbf{z}}_o$ osi tego układu. Macierz obrotu jest obliczana na podstawie reprezentacji oś-kąt ${}^o \mathbf{R}(\mathbf{u}, \varphi) = \mathbf{Rot}(\mathbf{u}, \varphi)$, gdzie $\mathbf{u} \in \{\hat{\mathbf{x}}_o, \hat{\mathbf{y}}_o, \hat{\mathbf{z}}_o\}$, a $\varphi \in \{-\frac{\pi}{2}, -\pi, \frac{\pi}{2}, \pi\}$. Dlatego też, zadana ścieżka ruchu dla chwytaka może być opisana w układzie kostki jako obrót wokół jednej z osi tego układu o zadany kąt φ .

Przejsie do wykonania obrotu kolejną ścianką wymaga zmiany chwytu, to z kolei wymaga zwolnienia chwytu – umiejętność *Release* i generacji bezkolizyjnej trajektorii przechwytywania. Ułożenie kostki wymaga zatem wykonania n -razy sekwencji *Grasp* \rightarrow *Turn* \rightarrow *Release* (n jest liczbą ruchów potrzebnych do ułożenia kostki).

Ze względu na nieuniknione błędy wynikające m.in. z niedokładnej kalibracji oraz zgrubnej znajomości pozycji kostki, przy realizacji kolejnych faz manipulacji niezbędna jest modyfikacja nominalnych trajektorii ruchu i chwytów na podstawie bieżących odczytów z czujników. W umiejętności *Turn* wykorzystuje się informację o sile oddziaływań między kostką a chwytakami w celu unikania zakleszczeń. Tylko jeden z chwytaków obraca ściankę, drugi wykonuje ruchy kompensacyjne zapobiegające zakleszczeniom kostki. Podczas obracania warstw kostki jest realizowane sterowanie pozycyjno-siłowe, z wykorzystywaniem odczytów z czujników sił i momentów sił zamontowanych w nadgarstkach [Hu9]. Do opisu zadania sterowania pozycyjno-siłowego wybrano formalizm TFF (*Task Frame Formalizm*) [2].

4.1.2 Planowanie ścieżek dla dwóch robotów manipulujących pojedynczym obiektem [H7]

W pracy [H7] do planowania ścieżek ruchu dla dwóch robotów manipulujących pojedynczym obiektem zaproponowano podejście polegające na spełnieniu ograniczeń, bazujące na kinematyce systemu robotycznego o strukturze pętli kinematycznej. Jego idea polega na poszukiwaniu funkcji, w tym przypadku ścieżki, która spełnia ograniczenia i minimalizuje zadany wskaźnik jakości. Naturalną konsekwencją takiego podejścia jest sformułowanie zadania poszukiwania dopuszczalnych ścieżek ruchu w postaci zadania wariacyjnego. Polega ono na poszukiwaniu minimum warunkowego pewnego funkcjonału, przy czym na funkcje ze zbioru, na którym jest określony funkcjonał są nałożone warunki zwane więzami. Przy formułowaniu tego zadania istotnym założeniem jest osiągalność konfiguracji początkowej i końcowej (ustalone końce krzywej dopuszczalnej). Do rozwiązania tego zadania wykorzystano zmodyfikowaną metodę Ritza,

jedną z tzw. metod bezpośrednich rachunku wariacyjnego [4]. Podstawową ideą metod bezpośrednich jest to, że zadanie wariacyjne rozpatruje się jako graniczny przypadek zadania na ekstremum funkcji o skończonej liczbie zmiennych. Problem minimalizacji warunkowej funkcjonału jest sprowadzany do zadania programowania nieliniowego (*Non-Linear Programming* – NLP).

Ze względu na stosunkowo dużą złożoność obliczeniową zadania programowania nieliniowego pojawia się konieczność stosowania efektywnych obliczeniowo algorytmów rozwiązywania zadań optymalizacji nieliniowej. Do numerycznego rozwiązania zadania programowania nieliniowego wykorzystano algorytm IPOPT (bazujący na prymalno-dualnej metodzie punktu wewnętrznego dla zadań nieliniowych), będący obecnie jedną z wiodących technik optymalizacji nieliniowej dla zadań wielkiej skali [21].

Istotną zaletą proponowanego podejścia jest stosowalność zarówno dla układów z niedomiarem stopni swobody, jak też dla układów redundantnych [Hu5]. Podstawowym problemem przy planowaniu ruchu dla systemów z niedomiarem stopni swobody jest znalezienie dopuszczalnej ścieżki ruchu, łączącej konfigurację początkową z zadaną konfiguracją końcową. W przypadku układów redundantnych, tzn. z nadmiarem stopni swobody, mamy do czynienia niejako z problemem odwrotnym, polegającym mianowicie na wyborze, spośród wielu możliwych, ścieżki spełniającej ograniczenia i minimalizującej dany wskaźnik jakości.

Podsumowanie uzyskanych wyników:

1. Identyfikacja problemu i formalizacja jego opisu (opis kinematyczny układu z zamkniętą pętlą, ograniczenia).
2. Podejście wariacyjne do problemu planowania ścieżki (poszukiwanie minimum warunkowego funkcjonału).
3. Aproksymacja skończenie-wymiarowa zadania wariacyjnego i rozwiązanie zadania programowania nieliniowego.

Uzyskane wyniki pokazują, iż proponowane podejście do planowania dopuszczalnych ścieżek ruchu dla układów zawierających zamknięte łańcuchy kinematyczne może być z powodzeniem zastosowane w rzeczywistych zadaniach. Może być ono stosowane dla mechanizmów o dowolnej liczbie stopni swobody i dowolnej liczbie pętli kinematycznych. Nie jest wymagane rozwiązywanie odwrotnego zadania kinematyki dla złożonych mechanizmów zawierających zamknięte łańcuchy — w algorytmie jest wykorzystywane tylko proste zadanie kinematyki manipulatora. Obliczone ścieżki są dopuszczalne i dostatecznie gładkie.

Porównując opracowane przeze mnie podejście z powszechnie stosowanymi metodami bazującymi na losowym próbkowaniu, należy podkreślić, że w paradygmacie leżącym u podstaw tych metod nie zakłada się żadnej formy optymalizacji poszukiwanego rozwiązania. Znalezione ścieżki zazwyczaj nie są dostatecznie gładkie, cechują się gwałtownymi zmianami kierunku i często konieczne jest dodatkowe wygładzanie ścieżek. W odróżnieniu, ścieżki obliczane przez proponowany algorytm są gładkie oraz „bliskie” ścieżkom pożądanym, które mogą być dobierane stosownie do potrzeb [H7].

Opis szczegółowy**Przestrzeń konfiguracyjna systemu z zamkniętym łańcuchem kinematycznym**

Rozważany jest mechanizm $\mathcal{M} = (\mathcal{L}, \mathcal{J})$ składający się z sztywnych członów $L_i \in \mathcal{L}$ połączonych sztywnymi przegubami $J_k \in \mathcal{J}$, zawierający zamknięty łańcuch kinematyczny. Przestrzeń robocza \mathcal{W} mechanizmu \mathcal{M} jest pewnym podzbiorem przestrzeni fizycznej, którą można utożsamić z przestrzenią euklidesową \mathbb{R}^3 , czyli $\mathcal{W} \subset \mathbb{R}^3$. Domknięty i ograniczony podzbiór $\mathcal{O} \subset \mathbb{R}^3$ oznaczający obiekt jest wyróżnionym elementem tego mechanizmu. Używając standardowych metod parametryzacji dla \mathcal{M} , możemy wyrazić konfigurację mechanizmu \mathcal{M} jako wektor $\mathbf{q} = [q_1, \dots, q_n]^T$. Przy czym, w ogólnym przypadku, poszczególne zmienne konfiguracyjne mogą odpowiadać zarówno przegubom aktywnym (napędzanym), jak też biernym, np. gdy obiekt manipulacji nie jest pojedynczym ciałem sztywnym, lecz składa się dwóch lub więcej elementów połączonych przegubami (przykładem takiego obiektu jest kostka Rubika). Zbiór wszystkich punktów \mathbf{q} tworzy przestrzeń konfiguracyjną \mathbb{C} , zatem $\mathcal{M}(\mathbf{q})$ oznacza transformację \mathcal{M} do konfiguracji \mathbf{q} . Rozmaitość konfiguracyjna mechanizmu zawierającego pętle kinematyczne nie jest gładką rozmaitością, tak jak to ma miejsce w przypadku swobodnego ciała sztywnego. Niezależne ograniczenia konfiguracyjne wynikające z warunku zamknięcia pętli mają postać więzów holonomicznych i wyznaczają rozmaitość konfiguracyjną

$$\mathbb{C}_{clo} = \{\mathbf{q} : \mathbf{q} \in \mathbb{C} \wedge \mathbf{f}(\mathbf{q}) = \{f_1(\mathbf{q}) = 0, \dots, f_m(\mathbf{q}) = 0\}\} \quad (4)$$

Powyższe więzy holonomiczne można przekształcić do postaci wielomianowej, wówczas przestrzeń \mathbb{C}_{clo} może być przedstawiona jako rozmaitość *semi-algebraiczna* [15].

Dopuszczalne konfiguracje \mathbf{q} muszą spełniać także warunki bezkolizyjności z przeszkodami w przestrzeni roboczej i braku kolizji między poszczególnymi członami mechanizmu \mathcal{M} . Jeśli jako $\mathcal{B}_i \subset \mathcal{W}$, $i = 1, \dots, n_b$ oznaczymy zbiór przeszkód w przestrzeni roboczej, to *przestrzeń bezkolizyjną* (wolną) można zdefiniować jako

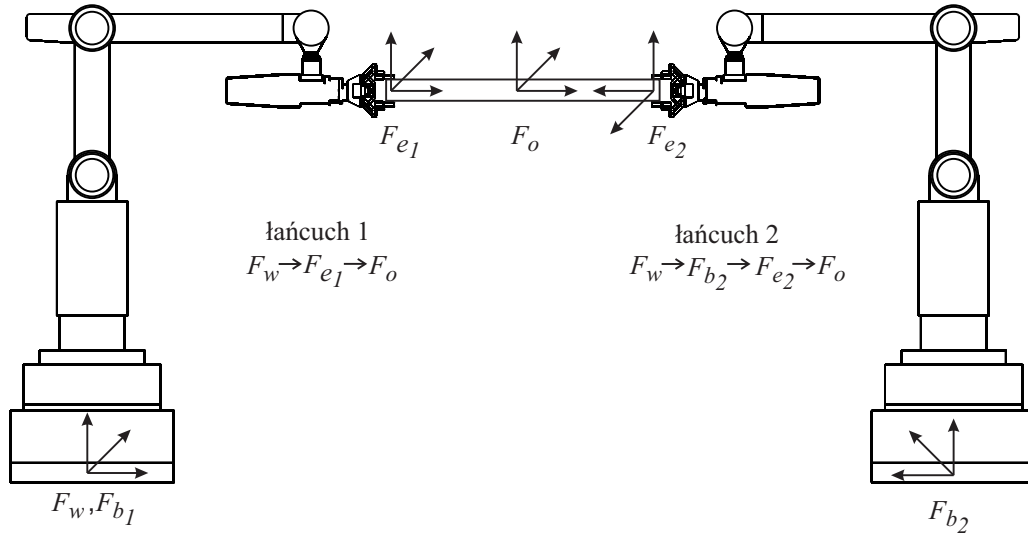
$$\mathbb{C}_{free} = \left(\bigcup_{i \in [1, n_b]} \{\mathbf{q} \in \mathbb{C} \mid (\mathcal{M}(\mathbf{q}) \cap \mathcal{B}_i = \emptyset)\} \right) \cup \left(\bigcup_{i, j \in CP} \{\mathbf{q} \in \mathbb{C} \mid (L_i(\mathbf{q}) \cap L_j(\mathbf{q}) = \emptyset)\} \right), \quad (5)$$

gdzie CP oznacza potencjalne pary kolizyjne $[i, j]$ ($i \neq j$) ogniw $L_i(\mathbf{q})$ i $L_j(\mathbf{q})$ w danej konfiguracji \mathbf{q} takie, że $i \in [1, n_l - 1]$ oraz $j \in [i + 1, n_l]$. W przestrzeni \mathbb{C}_{free} są konfiguracje, dla których poszczególne elementy $\mathcal{M}(\mathbf{q})$, w szczególności obiekt, mogą stykać się ze sobą i przeszkodami, ale nie mogą wnikać w głąb.

Z geometrycznego punktu widzenia na ruch systemu robotycznego nałożone są ograniczenia, wyrażone ogólnie w postaci więzów dwustronnych i więzów jednostronnych. Zbiór konfiguracji spełniających wszystkie ograniczenia geometryczne można przedstawić jako

$$\mathbb{C}_{val} = \mathbb{C}_{clo} \cap \mathbb{C}_{free} \quad (6)$$

W ogólnym przypadku, zbiór \mathbb{C}_{val} , może składać się z rozłącznych podzbiorów (składowych) o skomplikowanej strukturze. Przykładem mechanizmu \mathcal{M} z pętlą kinematyczną jest system składający się z dwóch manipulatorów z uchwyconym sztywno obiektem będącym ciałem sztywnym (rys. 3). Obiekt wraz ramionami robotów oraz podłożem tworzą zamknięty łańcuch kinematyczny. Więzy wynikające z warunku domknięcia pętli określają osiągalne pozycje obiektu.



Rys. 3: Struktura geometryczna dwóch ściśle współpracujących robotów

Mechanizm może być traktowany jako łańcuch kinematyczny zbudowany z dwóch podłańcuchów. Niech $\mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T]^T$ oznacza wektor współrzędnych konfiguracyjnych tego mechanizmu. Dla opisu systemu dwurobotowego (3) wprowadza się odpowiednie układy współrzędnych ortokartezjańskich:

- F_w — globalny układ odniesienia dla całego systemu,
- F_{b_i} — układ bazowy i -tego robota, $i = 1, 2$,
- F_o — układ związany z obiektem,
- F_{e_i} — układ związany z chwytakiem i -tego robota.

Pozycje układów bazowych robotów F_{b_i} względem globalnego układu odniesienia F_w są określone za pomocą macierzy ${}^w_b T$, $i = 1, 2$. Macierze ${}^{b_i}_{e_i} T(\mathbf{q}_i)$, $i = 1, 2$, reprezentują kinematykę prostą i -tego łańcucha i opisują pozycję układu F_{e_i} jako funkcję zmiennych konfiguracyjnych \mathbf{q}_i , zaś macierz ${}^{e_i}_o T$, $i = 1, 2$ określa pozycję układu obiektu F_o względem układu chwytaka F_{e_i} .

Warunki zamknięcia pętli kinematycznej można, w rozważanym przypadku, zapisać w postaci równania macierzowego

$${}^{b_1}_{e_1} T(\mathbf{q}_1) {}^{e_1}_o T - {}^{b_2}_{e_2} T(\mathbf{q}_2) {}^{e_2}_o T = \mathbf{0}, \quad (7)$$

gdzie ${}^{b_2}_{e_2} T$ jest macierzą przekształcenia jednorodnego opisującą pozycję układu bazowego F_{b_2} względem układu bazowego łańcucha F_{b_1} .

Ograniczenia wynikające z ograniczonych zakresów ruchu w przegubach oraz potencjalnych kolizji między poszczególnymi elementami łańcucha kinematycznego można ogólnie przedstawić w postaci ograniczeń nierównościowych

$$\mathbf{g}(\mathbf{q}) \leq \mathbf{0}. \quad (8)$$

Zadanie planowania ścieżki ruchu jako problem wariacyjny

Podstawowym zadaniem jest planowanie ścieżki w dopuszczalnej przestrzeni konfiguracyjnej \mathcal{C}_{val} , polegające na obliczeniu krzywej $\mathbf{q}(\cdot)$, która realizuje odpowiednio gładką ścieżkę (krzywą) w przestrzeni zadaniowej. Dla danej konfiguracji początkowej $\mathbf{q}_p \in \mathcal{C}_{val}$ i końcowej $\mathbf{q}_f \in \mathcal{C}_{val}$ należy znaleźć dostatecznie gładką krzywą

$$\mathbf{q}(\cdot) : s \in [0, 1] \mapsto \mathbf{q}(s) = \mathbf{q} \in \mathcal{C}_{val}$$

taką, że $\mathbf{q}(0) = \mathbf{q}_p$ oraz $\mathbf{q}(1) = \mathbf{q}_f$. Parametr s może być utożsamiony z znormalizowaną długością krzywej określanej przez wybrany punkt, np. początek układu współrzędnych związany z obiektem. Warunkiem koniecznym istnienia rozwiązania tego zadania, jest przynależność obu konfiguracji \mathbf{q}_p i \mathbf{q}_f do jednej składowej spójnej przestrzeni \mathcal{C}_{val} . Krzywa $\mathbf{q}(\cdot)$ jest pewnym opisem parametrycznym ścieżki ruchu. Dla tak zdefiniowanego zadania planowania ścieżki przestrzeń \mathcal{C}_{clo} jest rozmaitością algebraiczną o ko-wymiarze 3 (dla przestrzennych łańcuchów zamkniętych), która może być nieparametryzowalna [16]. Jednakże, wymiar tej rozmaitości jest znacząco mniejszy od wymiaru przestrzeni konfiguracyjnej, w której \mathcal{C}_{clo} jest zanurzona.

Wykorzystując odpowiednią parametryzację do opisu pozycji obiektu więzy (7) można przedstawić w postaci:

$$\mathbf{h}(s, \mathbf{q}) = \mathbf{p}_1(\mathbf{q}_1(s)) - \mathbf{p}_2(\mathbf{q}_2(s)) = \mathbf{0}, \quad \forall s \in [0, 1], \quad (9)$$

gdzie wektor $\mathbf{p}_i(\mathbf{q}_i(s)) \in \mathbb{R}^6$, $i = 1, 2$ opisuje pozycję układu F_o związanego z obiektem w układzie F_w jako funkcję współrzędnych \mathbf{q}_i .

Przyjmujemy $\mathbf{p}_d(\cdot)$ jako zadaną ścieżkę (określającą zarówno krzywą położenia, jak i orientacji) obiektu w przestrzeni zadaniowej łączącą pozycję początkową \mathbf{p}_p i zadaną pozycję końcową \mathbf{p}_f . Załóżmy, że ścieżka $\mathbf{p}_d(\cdot)$ została narzucona arbitralnie bez uwzględniania, czy jest ona dopuszczalna (poza warunkami brzegowymi, które musi spełniać). Dla systemu przedstawionego na rysunku 3, problem planowania można sprowadzić do zadania minimalizacji funkcjonału $I[\mathbf{q}(\cdot)]$ postaci:

$$I[\mathbf{q}(\cdot)] = \int_0^1 (\mathbf{p}_i(\mathbf{q}_i(s)) - \mathbf{p}_d(s))^T W (\mathbf{p}_i(\mathbf{q}_i(s)) - \mathbf{p}_d(s)) ds, \quad (10)$$

przy ograniczeniach

$$\mathbf{q} \in \mathcal{Y} \triangleq \left\{ \mathbf{q}(\cdot) \in C^2([0, 1]) : \mathbf{h}(s, \mathbf{q}) = \mathbf{0} \wedge \mathbf{g}(s, \mathbf{q}) \leq \mathbf{0} \wedge \mathbf{q}(0) = \mathbf{q}_p, \mathbf{q}(1) = \mathbf{q}_f \right\}, \quad (11)$$

gdzie $\mathbf{p}_i(\cdot)$, $i = 1 \vee i = 2$ opisuje aktualną ścieżkę dla układu związanego z obiektem jako funkcję współrzędnych konfiguracyjnych jednego z manipulatorów (odpowiednio manipulatora 1 lub 2). Wektor \mathbf{p}_i opisuje pozycję obiektu przy odpowiedniej parametryzacji współrzędnych położenia i orientacji. Macierz wagowa $W = W^T \geq 0$ o wymiarach (6×6) jest macierzą dodatnio pół-określoną dobieraną stosowanie do postawionego zadania, określającą, które współrzędne obiektu będą lepiej „nadażały” za zadaną ścieżką. Ograniczenia równościowe postaci (7) przy przyjętej parametryzacji można zapisać jako

$$\forall s \in [0, 1] h_k(s, \mathbf{q}) = p_{1k}(\mathbf{q}_1(s)) - p_{2k}(\mathbf{q}_2(s)) = 0, \quad k = 1, \dots, 6 \quad (12)$$

Rozwiązaniem zadania (10)–(11) (o ile takie rozwiązanie istnieje) są funkcje reprezentujące dopuszczalne ścieżki we współrzędnych konfiguracyjnych obu robotów.

Do najbardziej ogólnych, a jednocześnie efektywnych obliczeniowo, metod przybliżonego rozwiązywania zadań wariacyjnych z ograniczeniami zalicza się tzw. metody bezpośrednie. Podstawową ideą metod bezpośrednich jest to, że zadanie wariacyjne rozpatruje się jako graniczny przypadek zadania na ekstremum funkcji o skończonej liczbie zmiennych. W niniejszej pracy wykorzystano jedną z takich metod, zwaną metodą Ritz'a [4, 9].

Aproksymacja skończenie-wymiarowa zadania wariacyjnego

Idea proponowanego podejścia polega na poszukiwaniu rozwiązań zadania wariacyjnego nie na wszystkich dopuszczalnych krzywych \mathbf{q} , lecz tylko wśród kombinacji liniowych $N+1$ pierwszych funkcji z ciągu $\{\varphi_j\}_{j=0}^{\infty}$. Dopuszczalność oznacza tutaj tylko, że funkcje φ_j należą do dziedziny funkcjonału, ale nie muszą spełniać ograniczeń.

$$\mathbf{q}(\cdot) = \sum_{j=0}^N \mathbf{c}_j \varphi_j(\cdot), \quad \dim \mathbf{q} = \dim \mathbf{c}_j, \quad (13)$$

gdzie \mathbf{c}_j jest wektorem stałych współczynników, zaś funkcje $\mathbf{q}(\cdot)$ leżą w zbiorze określoności danego funkcjonału. Ciąg funkcji bazowych φ_j nie może być wybrany w sposób dowolny. Istota metod bezpośrednich polega na takim wyborze skończonej wymiarowej bazy funkcyjnej dla dyskretyzacji problemu, aby baza ta była możliwie najlepiej dopasowana do ograniczeń zadania. Wybór konkretnej bazy jest uwarunkowany wymaganym charakterem rozwiązania. Dokładność aproksymacji zależy od wymiaru N podprzestrzeni $\mathcal{Y}_N \subset \mathcal{Y}$ i od postaci funkcji bazowych φ_j . Zakładamy, że funkcje φ_j są klasy C^2 i spełniają warunki brzegowe, a ciąg jest zupełny w sensie normy przestrzeni \mathcal{Y} (tzn. liniowe kombinacje (13) tworzą zbiór wszędzie gęsty w przestrzeni \mathcal{Y}).

Niech Δ_N oznacza podział przedziału $[0, 1] \subset \mathbb{R}^1$ na N równych podprzedziałów $\Delta s = s_{j+1} - s_j = \frac{1}{N}$ dla s_j , $j = 0, 1, \dots, N-1$, gdzie s_j są takimi wartościami, że $0 = s_0 < s_1 < \dots < s_N = 1$. Jednymi z możliwych funkcji bazowych są znormalizowane funkcje B-sklejane, które definiuje się w sposób rekurencyjny. Stała B-funkcja sklejana w j -tym podprzedziale jest postaci

$$b_{j,0}(s) = \begin{cases} 1 & s \in [s_j, s_{j+1}], \\ 0 & \text{dla pozostałych } s. \end{cases} \quad (14)$$

B-funkcja sklejana stopnia m w przedziale $[s_j, s_{j+m+1}]$ jest zdefiniowana jako

$$b_{j,m}(s) = \frac{s - s_j}{s_{j+m} - s_j} b_{j,m-1}(s) + \frac{s_{j+m+1} - s}{s_{j+m+1} - s_{j+1}} b_{j+1,m-1}(s). \quad (15)$$

W rozważanym przypadku jako funkcje bazowe φ_j wybrano sześciennne B-funkcje sklejane:

$$b_{j,3}(s) = \frac{s - s_j}{s_{j+3} - s_j} b_{j,2}(s) + \frac{s_{j+4} - s}{s_{j+4} - s_{j+1}} b_{j+1,2}(s). \quad (16)$$

Istotną cechą B-funkcji sklejanych jest możliwość *lokalnej modyfikacji* krzywej. Oznacza to, iż w przypadku konieczności lokalnej zmiany przebiegu ścieżki ruchu konieczne jest obliczenie nowej wartości tylko kilku współczynników krzywej, pozostała część pozostaje bez zmian.

Problem poszukiwania dopuszczalnego toru ruchu dla zespołu robotów może być sprowadzony do zadania znalezienia minimum funkcji $(N+3) \times n$ zmiennych. Bez utraty ogólności można

przyjąć macierz wagową w postaci $W = \text{diag}(w_1, \dots, w_6)$. Funkcjonał (10) można wyrazić jako funkcję $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T]^T$ ($\mathbf{c}_i \in \mathbb{R}^{n_i \times (N+3)}$, $i = 1, 2$), współczynników rozwinięcia $\mathbf{q}(\cdot)$ w bazie funkcyjnej $\{\varphi_j\}_{j=-1}^{N+1}$

$$\min_{\mathbf{c}} \sum_{j=0}^{N-1} \int_{s_j}^{s_{j+1}} \sum_{k=1}^6 w_k \left(p_{1k} \left(\sum_{i=j-1}^{j+2} \mathbf{c}_{1i} \varphi_i(s) \right) - p_k^d(s) \right)^2 ds \quad (17)$$

Poszukujemy zatem minimum funkcji (17) względem \mathbf{c} , przy ograniczeniach

$$\mathbf{c} \in \mathcal{Y}_N = \left\{ \mathbf{c} \in \mathbb{R}^{n \times (N+3)} : \mathbf{h}(s_j, \mathbf{c}) = 0 \wedge \mathbf{g}(s_j, \mathbf{c}) \leq 0, j = 0, \dots, N \right\} \quad (18)$$

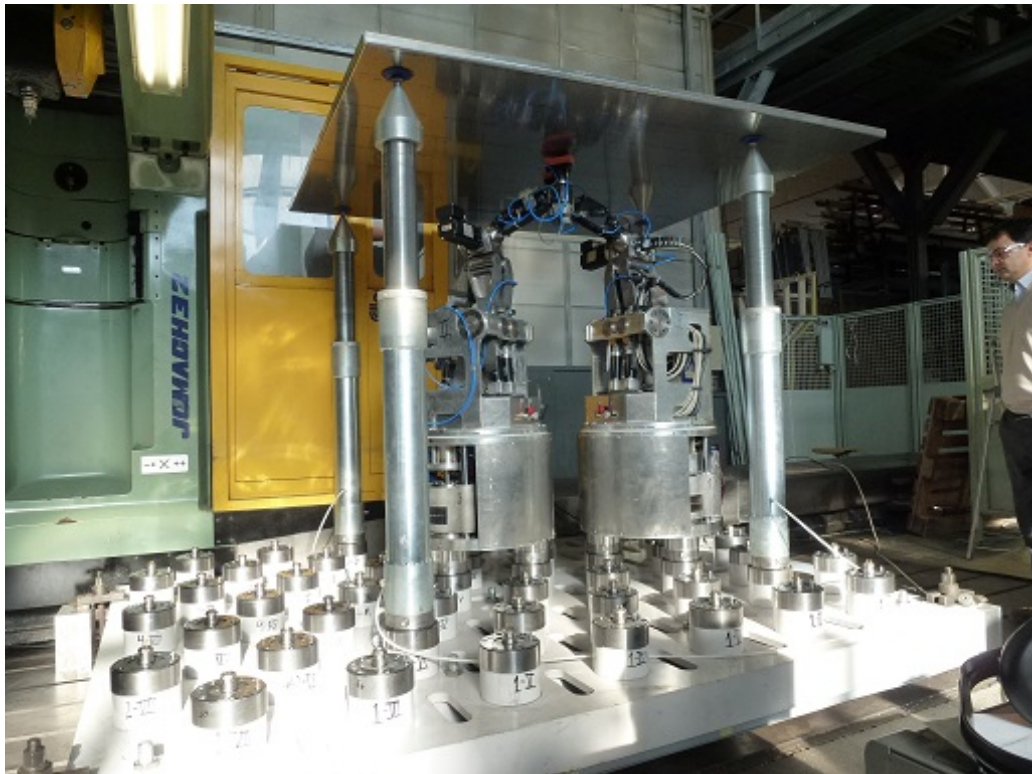
Tak otrzymane zadanie programowania nieliniowego może być rozwiązywane numerycznie za pomocą odpowiednich algorytmów optymalizacji nieliniowej z ograniczeniami, w szczególności za pomocą *solwerów* bazujących na metodzie punktu wewnętrznego dla zadań nieliniowych.

W pracach [H7] oraz [Hu5] do rozwiązania zadania programowania nieliniowego (17) – (18) wykorzystano algorytm IPOPT – prymalno-dualny algorytm punktu wewnętrznego z metodą minimalizacji w kierunku opartą na technice filtra [21]. IPOPT korzysta z drugich pochodnych funkcji celu i ograniczeń, co znacząco wpływa na szybkość zbieżności do rozwiązania lokalnego. Jako punkt startowy dla *solwera* IPOPT użyto wartości współczynników \mathbf{c} otrzymanych w wyniku rozwiązania pomocniczego układu równań liniowych dla uprzednio wyznaczonej dopuszczalnej ścieżki początkowej. Wyznaczenie ścieżki początkowej polega na iteracyjnym poszukiwaniu kolejnych jej punktów rozwiązując zadanie optymalizacji dla pojedynczego punktu, a nie dla całej ścieżki. Oczywiście nie ma gwarancji, że otrzymany w ten sposób ciąg punktów da ciągłą ścieżkę początkową, ale jest to zbiór punktów dopuszczalnych. W celu uniknięcia wyznaczania i wprowadzania skomplikowanych wzorów analitycznych na pierwsze i drugie pochodne funkcji, skorzystano z biblioteki automatycznego różniczkowania CppAD [1].

4.2 Planowanie rozmieszczenia robotów-podpór rekonfigurowalnego systemu mocowań [H1–H5, H8]

Zagadnienia badawcze przedstawione w artykułach [H1–H5, H8] dotyczą metody planowania ruchu oraz sterowania zespołem mobilnych robotów-podpór wchodzących w skład elastycznego systemu mocowań dla cienkościennych detali o dużych rozmiarach. System mocowań jest urządzeniem do pozycjonowania, unieruchamiania oraz podpierania przedmiotu w trakcie procesu obróbczego.

Głównym obszarem zastosowań systemów mocowań dla dużych, cienkościennych arkuszy metalowych lub kompozytowych płyt jest przemysł lotniczy, gdzie ze względu na wysokie wymagania dotyczące dokładności obróbki elementów poszycia kadłubów samolotów, niezbędne jest zapewnienie sztywnego podparcia i unieruchomienia elementu. Zadaniem mobilnych podpór jest sztywne i stabilne podparcie w celu redukcji statycznych i dynamicznych odkształceń materiału w otoczeniu miejsc aktualnie poddawanych obróbce maszynowej. Jest to zatem przykład manipulacji polegającej na unieruchamianiu obiektu. Prototyp wieloagentowego systemu mocującego powstał w wyniku prac w projekcie SwarmItFIX w ramach 7 Programu Ramowego Unii Europejskiej [5, H5] (rys. 4). Działanie systemu zostało eksperymentalnie zweryfikowane



Rys. 4: Stanowisko obróbcze z rekonfigurowalnymi podporami-robotami

w warunkach przemysłowych, w zakładach lotniczych Piaggio Aero Industries, w Finale Ligure, Włochy. Zakłada się, że poddawany obróbce cienkościenny element o dużych rozmiarach jest wstępnie mocowany za pomocą odpowiedniej liczby stałych uchwytów. Rozważane są dwie operacje obróbki maszynowej: frezowanie i wiercenie otworów. Sekwencje obróbcze są opisane w postaci danych CAD/CAM, które są danymi wejściowymi dla obrabiarek sterowanych numerycznie. Dysponując danymi CAD/CAM – geometrycznymi i czasowymi o obrabianym detalu, operacjach maszynowych i narzędziu obróbczym – zadaniem planera jest wygenerowanie sekwencji pozycji agentów-podpór zapewniających odpowiednią sztywność podparcia w otoczeniu poruszającego się narzędzia.

Do planowania ruchu i rozmieszczania mobilnych podpór rozważano dwa podejścia. Pierwsze podejście polega na zdefiniowaniu tego problemu jako zadania optymalizacji nieliniowej z ograniczeniami i rozwiązaniu zadania z użyciem wydajnych algorytmów programowania nieliniowego. Przedstawione w artykułach [H8, H9] rozwiązanie ogranicza się do wyznaczenia sekwencji dopuszczalnych pozycji głowic podpierających przy spełnieniu zadanych ograniczeń geometrycznych. Celem tych badań było sprawdzenie, czy istnieją dopuszczalne plany rozmieszczenia głowic podpierających dla wybranych detali i procesów obróbczych. W drugim podejściu sformułowano zadanie planowania jako problem spełnienia ograniczeń. W pracach [H1–H4] przedstawiono propozycję kompletnego systemu planowania dla rekonfigurowalnych agentów-podpór. Zadanie wyznaczenia sekwencji akcji (pozycji podparcia i ścieżek przejścia) poszczególnych elementów składowych (głowic, baz i PKM) agentów jest formułowane jako problem spełnienia ograniczeń (*Constraint Satisfaction Problem* – CSP). Proces planowania jest dekomponowany na cztery etapy planowania: dopuszczalnych pozycji głowic, sekwencji odpowiednich pozycji baz, ścieżek ruchu dla głowic i manipulatorów równoległych, oraz trajektorii ruchu baz i mani-

pulatorów.

Udział habilitanta w opracowaniu systemu planowania:

1. Analiza i specyfikacja wymagań wobec elementów składowych systemu mocowań: przestrzenie robocze, dopuszczalne prędkości i przyspieszenia głowic, manipulatorów równoległych i baz jezdnych.
2. Sformułowanie problemu rozmieszczenia głowic jako zadania programowania nieliniowego, a następnie rozwiązanie tego zadania z wykorzystaniem efektywnych obliczeniowo algorytmów numerycznych.
3. Udział w sformułowaniu zadania planowania jako hierarchicznego problemu CSP i opracowaniu nowej metody jego rozwiązania.
4. Udział w implementacji i testowaniu kompletnego systemu planowania (planera). Wyznaczenie planów dla rekonfigurowalnego systemu mocowań dla wybranych detali i procesów obróbczych.

4.2.1 Budowa rekonfigurowalnych mocowań [H2, H5]

Tradycyjne podpory stosowane przy obróbce dużych elementów wykonanych z metalowych cienkich blach są dużymi formami o specyficznym kształcie dedykowanym danemu detalowi i nie są rekonfigurowalne (rys. 5). Modułowe, elastyczne systemy mocowań mogą być częściowe rekon-



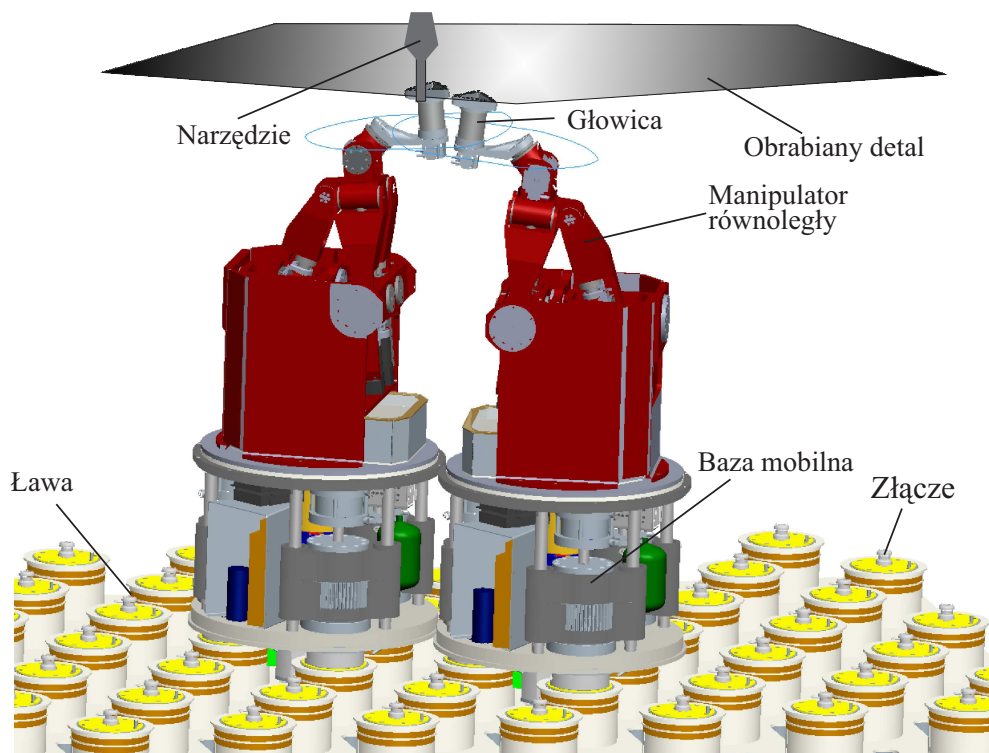
Rys. 5: Tradycyjna forma podpierająca stosowana w przemyśle lotniczym

figurowalne z ręcznym przestawianiem podpór lub samo-rekonfigurowalne z matrycą podpór,

z których każda ma wbudowany siłownik. Oprócz tradycyjnych systemów mocowań są stosowane systemy zrobotyzowane, gdzie rolę mocowań i uchwytów pełnią manipulatory wyposażone w odpowiednie chwytaki. Jednakże, proponowane dotychczas zrobotyzowane systemy mocowań, ze względu na użycie niewielkiej liczby stacjonarnych robotów, mają zastosowanie jedynie do mocowania detali o niezbyt dużych rozmiarach i zazwyczaj sztywnych. Zastosowanie tego typu mocowań dla elementów o dużych rozmiarach poprzez zwiększenie liczby robotów znacznie podnosi koszt całego systemu mocowań. Opracowany system rekonfigurowalnych mocowań-podpór jest nowatorskim rozwiązaniem i łączy zalety modułowych systemów mocowań oraz systemów zrobotyzowanych z manipulatorami stacjonarnymi pełniącymi taką rolę. Umożliwia szybką modyfikację i dostosowanie systemu mocowań do częstych zmian produktów, co jest szczególnie istotne przy produkcji małoseryjnej lub jednostkowej. Tym samym pozwala na znaczne obniżenie kosztów i skrócenie czasu produkcji.

Opis szczegółowy

Zrobotyzowany system rekonfigurowalnych mocowań-podpór składa się z dwóch lub więcej manipulatorów mobilnych przemieszczających się po specjalnej ławie i podpierających od spodu obrabiany przedmiot (rys. 6). Każdy fizyczny agent składa się z (rys. 6): z bazy mobilnej, z posa-



Rys. 6: System rekonfigurowalnych agentów-podpór

dowionym na niej manipulatorem równoległym (*Parallel Kinematic Machine* – PKM), którego efektem jest głowica samo-adaptująca się do lokalnego kształtu podpieranego przedmiotu. Baza mobilna ma trzy nogi ustawione w wierzchołkach trójkąta równobocznego i porusza się po ławie wykonując ruchy wahadłowe polegające na obrocie wokół jednej z trzech nóg, podczas gdy dwie pozostałe są podniesione. W ławie oraz w nogach bazy mobilnej są zamontowane specjalne elementy złącza mocująco-bazujących, które umożliwiają mocne i precyzyjne

połączenie oraz szybkie rozłączanie. Złącza (trzcienie) w ławie są rozmieszczone w postaci regularnej siatki. Baza mobilna przemieszcza się wykonując obroty będące krotnością kąta 60° wokół nogi podpierającej. Manipulator równoległy (PKM) jest tripodem z końcówką sferyczną i ma 6 stopni swobody. Dodatkowy, siódmy stopień swobody realizuje obrót głowicy wokół osi przechodzącej przez jej środek. Ponadto podstawa manipulatora równoległego obraca się względem bazy mobilnej. Robot-podpora jest zatem manipulatorem redundantnym o hybrydowej strukturze szeregowo-równoległej. Głowica podpierająca ma kształt trójkąta równobocznego i zawiera płyn magnetoreologiczny zmieniający stan skupienia pod wpływem pola magnetycznego. Dzięki temu, gdy płyn jest w stanie ciekłym głowica może dopasować się do lokalnego kształtu podpieranego elementu, a po utwardzeniu zapewniać sztywne podparcie. Dodatkowo w głowicy jest wytwarzane podciśnienie, dzięki czemu głowica szczelnie przylega do powierzchni płyty. Zastosowanie podpór o takiej konstrukcji umożliwia łatwą i szybką rekonfigurację oraz dostosowanie się do różnych kształtów i rozmiarów obrabianych detali przy jednoczesnym zapewnieniu odpowiedniej sztywności podparcia. Istotną zaletą jest możliwość szybkiego przemieszczania się robotów, w porównaniu do dotychczas stosowanych w przemyśle elastycznych systemów mocowań, co z kolei umożliwia użycie mniejszej liczby podpór i ich koncentrację w obszarach aktualnie poddawanych obróbce.

4.2.2 Planowanie rozmieszczenia rekonfigurowalnych agentów-podpór jako zadanie optymalizacji z ograniczeniami [H8, H9]

W pracach [H8, H9] problem rozmieszczenia głowic podpierających dla procesu wiercenia i frezowania wzdłuż zadanego konturu sformułowano jako zadanie optymalizacji nieliniowej z ograniczeniami. Proces planowania jest wieloetapowy. W początkowej fazie planowania wykonuje się analizę konturu lub krzywej odpowiadającej procesowi frezowania względnie wiercenia szeregu otworów. Następnie wyznacza się miary odległości i funkcje kosztu oraz formułuje problem optymalizacyjny w terminach programowania nieliniowego. Na koniec dobiera się właściwy algorytm rozwiązania tego problemu.

Opis szczegółowy

Kontur jest opisany jako łamana zamknięta P (wielobok) w przestrzeni 2D i zdefiniowany przez uporządkowany zbiór jej wierzchołków:

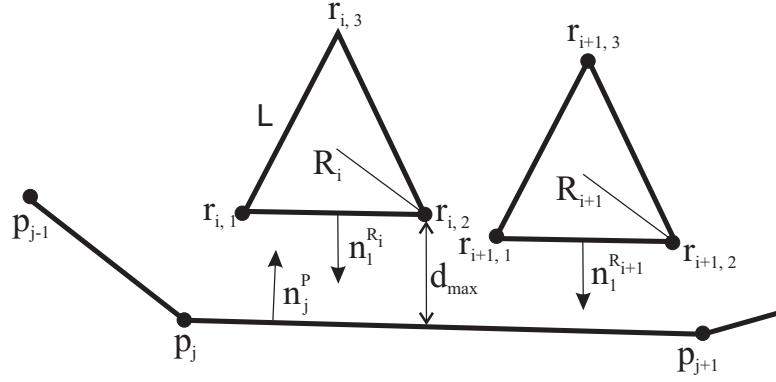
$$P = \{p_1, \dots, p_{M+1}\} = \{(x_1, y_1), \dots, (x_{M+1}, y_{M+1})\}, \quad (19)$$

gdzie $p_{M+1} = p_1$, czyli łamana składa się z M odcinków. Wierzchołki są klasyfikowane w zależności od miary kąta między dwoma segmentami liniowymi tworzącymi kąt wierzchołkowy α . Z kolei od rodzaju segmentu i jego ograniczających wierzchołków zależy strategia rozmieszczania głowic. Głowice podpierające mają kształt trójkąta równobocznego

$$\mathcal{R}_i = \{r_{i,1}, \dots, r_{i,4}\}, \quad \text{gdzie } r_{i,4} = r_{i,1}, \quad (20)$$

o długości boku równym L .

Pozycja i -ta głowicy jest dana jako $R_i(x_i, y_i, \theta_i)$, ($i = 1, \dots, N$). Dla każdego boku k głowicy i określamy wektor normalny $n_k^{R_i}$ oraz wektor normalny n_j^P do każdego odcinka P_j konturu P .



Rys. 7: Ograniczenia geometryczne dla rozmieszczenia głowic

Na rysunku 7 zilustrowano ograniczenia geometryczne dla rozmieszczenia głowic: Ograniczenia na dopuszczalne pozycje głowic R_i określają:

- Maksymalną dopuszczalną odległość d_{max} głowicy od konturu zapewniającą odpowiednią sztywność podparcia, aby nie pojawiły się zbyt duże drgania i odkształcenia materiału w trakcie obróbki:

$$d_i(P, R_i) \leq d_{max}, \quad i = 1, \dots, N - 1 \quad (21)$$

- Minimalną odległość głowicy od konturu d_{min} określającą margines bezpieczeństwa aby narzędzie nie weszło w kontakt z głowicą:

$$d_i(P, R_i) \geq d_{min}, \quad i = 1, \dots, N \quad (22)$$

- Niezachodzenie głowic na siebie:

$$\text{int}(R_i) \cap \text{int}(R_{i+1}) = \emptyset, \quad i = 1, \dots, N - 1, \quad (23)$$

gdzie $\text{int}(R_i)$ określa wnętrze trójkąta R_i . Możliwy jest jednak, kontakt głowic krawędziami lub wierzchołkami. Głowice stykają się jeśli spełnione są następujące warunki:

$$(r_{i,j-1}(\theta_i) - r_{i,j}(\theta_i)) \cdot n_k^{R_{i+1}}(\theta_{i+1}) \geq 0 \wedge (r_{i,j+1}(\theta_i) - r_{i,j}(\theta_i)) \cdot n_k^{R_{i+1}}(\theta_{i+1}) \geq 0, \quad (24)$$

$$i = 1, \dots, N - 1, \quad j, k = 1, 2, 3.$$

Jeśli powyższy warunek jest spełniony wówczas krawędź $E_k^{R_{i+1}}$, głowicy R_{i+1} i wierzchołka $r_{i,j}$ głowicy R_i stykają się. W jednym skrajnym przypadku pokrywają się wierzchołki $r_{i,j}$ i $r_{i+1,k}$, a w drugim skrajnym przypadku wierzchołki $r_{i,j}$ i $r_{i+1,k+1}$. Analogicznie, gdy spełniony jest warunek

$$(r_{i+1,j-1}(\theta_{i+1}) - r_{i+1,j}(\theta_{i+1})) \cdot n_k^{R_i}(\theta_i) \geq 0 \wedge (r_{i+1,j+1}(\theta_{i+1}) - r_{i+1,j}(\theta_{i+1})) \cdot n_k^{R_i}(\theta_i) \geq 0, \quad (25)$$

$$i = 1, \dots, N - 1, \quad j, k = 1, 2, 3,$$

krawędź $E_k^{R_i}$ głowicy R_i i wierzchołka $r_{i+1,j}$ głowicy R_{i+1} stykają się. Podobnie jak wyżej, w skrajnych przypadkach, pokrywają się odpowiednie wierzchołki sąsiednich głowic.

- Maksymalną odległość D_{max} między kolejnymi głowicami zapewniającą odpowiednie podparcie, gdy narzędzie jest między strefami podparcia zapewnianymi oddzielnie przez każdą z głowic:

$$d_i(R_i, R_{i+1}) \leq D_{max}, \quad i, = 1, \dots, N - 1 \quad (26)$$

Odległość między dwoma wielobokami P i Q w przestrzeni 2D można zapisać wzorem

$$d_m(P, Q) = \min_{p \in P, q \in Q} \|p - q\| = \min_{z \in P \ominus Q} \|z\|, \quad (27)$$

gdzie

$$P \ominus Q = \{z | z = p - q, p \in P, q \in Q\} = Z,$$

przy czym $Z = P \ominus Q$ jest różnicą Minkowskiego dwóch zbiorów. W rozważanym przypadku do obliczenia odległości $d_m(P, Q)$ wykorzystano efektywny algorytm GJK (Gilbert-Johnson-Keerthi) [11].

Obliczenie dopuszczalnej sekwencji pozycji głowic wymaga rozwiązania następującego zadania optymalizacji nieliniowej

$$\min \max f_i(\mathbf{x}) = d_i^2(P, R_i), \quad i = 1, \dots, N, \quad (28)$$

gdzie $\mathbf{x} = [x_1, y_1, \theta_1, \dots, x_N, y_N, \theta_N]^T$, zaś $d_i^2(P, R_i)$ jest kwadratem odległości głowicy R_i od konturu P . W celu efektywnego rozwiązania zadania typu minimaxowego wprowadzamy dodatkową zmienną z oraz N dodatkowych ograniczeń nierównościowych postaci

$$f_i(\mathbf{x}) - z \leq 0, \quad i = 1, \dots, N \quad (29)$$

Co pozwala zastąpić zadanie (28) z ograniczeniami (21)–(26) równoważnym zadaniem

$$\min z, \quad (30)$$

z ograniczeniami jak w pierwotnym zadaniu oraz dodatkowymi ograniczeniami postaci (29). W rozważanym przypadku zadanie programowania nieliniowego rozwiązano w środowisku Matlab z wykorzystaniem funkcji z pakietu Optimization Toolbox [17]. Na zbieżność algorytmu optymalizacji i uzyskane rozwiązanie (sekwencję pozycji głowic) decydujący wpływ ma dobór punktu startowego oraz wartości parametrów d_{min} , d_{max} , D_{max} . W przykładzie numerycznym wyznaczono dopuszczalną sekwencję pozycji głowic podpierających dla procesu frezowania konturu detalu (statecznika samolotu) o wymiarach $2800 \text{ mm} \times 1100 \text{ mm}$. Kontur charakteryzował się stosunkowo dużą nieregularnością i był aproksymowany wielobokiem o 26 bokach. Liczba zmiennych w przykładowym zadaniu optymalizacji wynosiła $N = 268$, co odpowiada 89 pozycjom głowic. Liczba nieliniowych ograniczeń nierównościowych wynosiła 882 a liniowych ograniczeń nierównościowych 26.

4.2.3 Planowanie rozmieszczania ruchomych podpór jako problem spełnienia ograniczeń [H1–H4]

W pracach [H1–H4], oraz [Hu2] jest przedstawiona metoda planowania ruchu (sekwencji pozycji podparcia oraz ścieżek przejścia między nimi) dla rekonfigurowalnych robotów-podpór. Proponowane podejście jest nowatorskie w dwóch aspektach.

1. Po pierwsze, zadanie planowania sformułowaliśmy jako problem spełnienia ograniczeń (CSP), w odróżnieniu od powszechnie stosowanego podejścia polegającego sformułowaniu problemu planowania jako zadania optymalizacji z ograniczeniami.
2. Po drugie, do rozwiązania problemu CSP zaproponowaliśmy nowy algorytm polegający na przeszukiwaniu kolejno trzech przestrzeni stanu tworzących strukturę hierarchiczną. Ponadto, dopuszczamy rozwiązanie częściowe (pamiętane jest dotychczasowe najlepsze rozwiązanie częściowe), gdyż może nie być rozwiązania zupełnego. Rozwiązanie tego problemu polega na przeszukiwaniu dyskretnej przestrzeni stanu przy spełnianiu zadanych ograniczeń. Proponowana metoda nie wymaga dużych mocy obliczeniowych i umożliwia spełnienie wszystkich istotnych ograniczeń wynikających z geometrii i właściwości materiału obrabianego elementu, procesu obróbczego oraz własności robotów-podpór.

W pracy [H2] przedstawiono szczegółowe sformułowanie problemu planowania rozmieszczania i relokacji mobilnych robotów-podpór i jego dekompozycji. Zadanie planowania podzielono na etapy: analizy konturu obrabianego detalu, planowania dopuszczalnej sekwencji pozycji głowic, planowania ścieżek baz oraz manipulatorów mobilnych i etap końcowy parametryzacji czasowej obliczonych wcześniej ścieżek ruchu. Na podstawie analizy danych CAD/CAM określono ograniczenia geometryczne i czasowe wynikające z kształtu detalu i procesu obróbczego.

W artykule [H1] przedstawiono szczegółowy algorytm „hierarchicznego CSP” i podano analizę jego złożoności obliczeniowej. W odróżnieniu od podziału dziedziny zmiennych na poddziedziny o różnych rozdzielczościach, zwykle rozpatrywanych w problemie hierarchicznego przeszukiwania dla CSP w ramach metod sztucznej inteligencji, zastosowaliśmy hierarchię zmiennych, naturalnie odpowiadających różnym efektorom robotów. Pozwoliło to na oczywiste zmniejszenie złożoności procesu przeszukiwania dziedziny problemu, ale przede wszystkim na wyznaczenie odpowiedniej hierarchii ograniczeń. Dzięki takiej dekompozycji problemu możliwe stało się także dobranie efektywnych strategii dyskretyzacji dziedziny zmiennych i ustalaniu preferencji przypisania zmiennym kolejnych wartości. W ten sposób powstał nowy efektywny algorytm o ogólnym charakterze rozwiązujący problem CSP dla współpracujących ze sobą agentów (robotów) o wzajemnie powiązanych efektorach.

W [H3] przedstawiono strategię wyboru (preferowania) pozycji głowic (w CSP: wartościowaniu zmiennych reprezentujących głowice – pierwszy podzbiór zmiennych w hierarchicznym CSP) dla operacji wiercenia otworów po okręgu i po prostej.

Pozycja [H4] ma charakter całościowej prezentacji opracowanego planera. Podano dekompozycję problemu planowania jako problemu CSP, zdefiniowano zmienne dla efektorów robota i ich dziedziny oraz wyznaczono potrzebne ograniczenia. Określono strategię wyznaczania kolejnych wartości dla zmiennych trzech rodzajów – reprezentujących głowice, bazy mobilne i manipulatory.

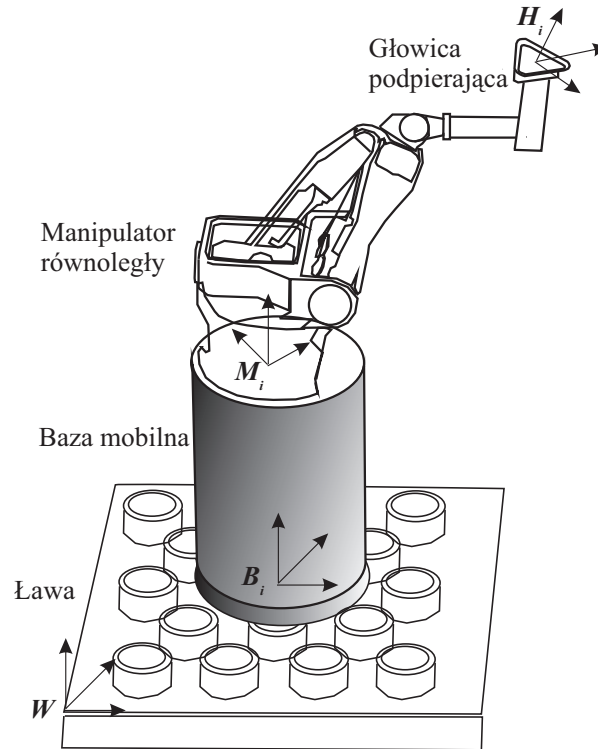
Opis szczegółowy

Dekompozycja problemu planowania

Do opisu pozycji podpory (manipulatora mobilnego) wprowadzono następujące układy współrzędnych kartezjańskich:

- globalny układ odniesienia $W = (X_W, Y_W, Z_W)$ związany z ławą,

- układ $B_i = (X_B, Y_B, Z_B)$ związany z bazą mobilną i ,
- układ $M_i = (X_M, Y_M, Z_M)$ związany z PKM i ,
- układ $H_i = (X_H, Y_H, Z_H)$ związany z głowicą i .



Rys. 8: Struktura agenta-podpory

${}^W H_i = ({}^W p_{H_i}, {}^W o_{H_i})$ określa pozycję (położenie ${}^W p_{H_i}$ i orientację ${}^W o_{H_i}$) głowicy względem globalnego układu W , ${}^B H_i$ określa pozycję głowicy względem układu B_i bazy mobilnej, a ${}^W B_i$ oznacza pozycję bazy mobilnej względem układu W , zaś M_i oznacza konfigurację PKM. Ponieważ głowica jest efektem końcowym manipulatora dlatego mamy ${}^B M_i = {}^B H_i$. W dalszych rozważaniach, dla rozróżnienia agentów, przyjmuje się, że nieparzyste indeksy i odpowiadają agentowi nr 1, a parzyste agentowi nr 2.

Problem planowania rozmieszczenia każdego agenta-podpory (rys. 8) jest dekomponowany na trzy hierarchiczne podproblemy planowania sekwencji pozycji głowicy H_i , sekwencji pozycji baz mobilnych B_i oraz manipulatorów równoległych M_i . Pierwszy problem polega na poszukiwaniu odpowiedniej sekwencji ${}^W H_i$, rozwiązanie drugiego problemu wiąże się z obliczeniem sekwencji ${}^W B_i$, a trzeci problem polega na wyznaczeniu odpowiedniej sekwencji pozycji ${}^B H_i$ oraz obliczeniu ciągu konfiguracji M_i przeprowadzających do tych pozycji. Ostatni etap tworzenia planu polega na obliczeniu trajektorii przejścia dla bazy oraz PKM między kolejnymi pozycjami podparcia.

Analiza geometrii detalu i segmentacja ścieżki narzędzia

Pierwszym etapem tworzenia planu jest analiza danego detalu i segmentacja ścieżki narzędzia odtwarzanej w trakcie jego obróbki. Korzystając z danych CAD/CAM wyznacza się kontur dla

frezowania lub krzywą przemieszczania narzędzia przy wierceniu otworów. Podczas frezowania narzędzie przemieszcza się wzdłuż zadanej krzywej w przestrzeni roboczej. Podobnie możemy wyznaczyć krzywą łączącą środki kolejno wierconych otworów. Strategia rozmieszczania głowic w obu przypadkach zależy od krzywej łamanej opisującej ścieżkę narzędzia. Inne będą jedynie parametry dla geometrycznych i czasowych ograniczeń w obu zadaniach, gdyż w przypadku frezowania ruch narzędzia ma charakter ciągły, a w przypadku wiercenia – zachodzi na przemian wiercenie i przemieszczanie narzędzia. Następnie kontur lub krzywa są dzielone na odcinki i wierzchołki określonych typów. Biorąc pod uwagę kształt głowicy (trójkąt równoboczny) wyróżnia się cztery typy wierzchołków konturu, wzdłuż którego będą rozmieszczane głowice. Typy wierzchołków są określane na podstawie wartości kąta wierzchołkowego α .

Typ 0 : $\alpha < 60^\circ$ – taki wierzchołek musi być wierzchołkiem początkowym segmentu, gdyż głowica nie może być w niego wpasowana.

Typ 1 : $60^\circ \leq \alpha < 90^\circ$ – taki wierzchołek jest pokryty jedną głowicą, której dwa boki są aktywne (podpierają dwie krawędzie konturu przecinające się w danym wierzchołku).

Typ 2 : $90^\circ \leq \alpha < 120^\circ$ – taki wierzchołek jest pokryty jedną głowicą z jednym aktywnym bokiem, alternatywne pozycje głowicy zawierają się między dwoma skrajnymi, w których aktywny bok głowicy jest równoległy do jednej z krawędzi konturu.

Typ 3 : $120^\circ \leq \alpha < 165^\circ$ – taki wierzchołek jest pokryty dwiema głowicami. Kąt wierzchołkowy jest podzielony na dwa równe kąty, w których alternatywne rozmieszczenia głowic mogą być takie jak dla wierzchołka typu 1.

Wynikiem etapu inicjacji jest lista zaklasyfikowanych segmentów konturu i ich wierzchołków, zdefiniowanych w globalnym układzie odniesienia, związanym z ławą.

Ograniczenia

Bardzo istotnym elementem przy formułowaniu zadania planowania rozmieszczania podpór jest właściwe zdefiniowanie ograniczeń. W rozważanym zadaniu należy uwzględnić następujące ograniczenia:

- Ograniczenia na dopuszczalne pozycje ${}^W H_i$ głowicy wynikające z wymaganych sił podparcia, które z kolei wynikają z kształtu, grubości i rodzaju materiału obrabianego przedmiotu. Na podstawie analizy metodą elementów skończonych dopuszczalnych ugięć wynikających z działania sił wywieranych przez narzędzie, ograniczenia te sprowadza się do ograniczeń geometrycznych postaci (21)–(26).
- Ograniczenia na dozwolone pozycje baz mobilnych ${}^W B_i$ wynikające z dyskretnych położень złączy dokujących w ławie oraz unikania kolizji między agentami.
- Ograniczenia na osiągalne pozycje ${}^B H_i$ wynikające z ograniczonej przestrzeni roboczej manipulatora równoległego.
- Ograniczenia czasowe ruchu wynikające z dopuszczalnych zakresów prędkości oraz przyspieszeń bazy mobilnej i manipulatora równoległego.

Problem spełnienia ograniczeń

Biorąc pod uwagę dużą liczbę ograniczeń, jak również dużą liczbę możliwych pozycji baz i głowic, które muszą być wybrane dla konturów o zróżnicowanych kształtach, zdecydowaliśmy, iż najbardziej adekwatną metodą rozwiązania zadania planowania będzie sformułowanie tego zadania jako problemu spełnienia ograniczeń i wykorzystanie odpowiednich technik sztucznej inteligencji do jego rozwiązania.

Formalnie dyskretny problem spełnienia ograniczeń (CSP) jest trójką $P = (\mathbf{X}, \mathbf{D}, \mathbf{C})$ gdzie:

- $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$ jest skończonym zbiorem zmiennych decyzyjnych;
- $\mathbf{D} = \{\mathbf{D}_1, \dots, \mathbf{D}_N\}$ jest zbiorem dziedzin zmiennych decyzyjnych. Każda dziedzina jest skończonym zbiorem możliwych wartości danej zmiennej decyzyjnej;
- $\mathbf{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_N\}$ jest zbiorem ograniczeń. Każde ograniczenie \mathbf{C}_i nakłada więzy na możliwe wartości z dziedziny \mathbf{D}_i dla pewnego podzbioru zmiennych.

Rozwiązaniem problemu spełnienia ograniczeń jest przydział każdej zmiennej decyzyjnej pewnej wartości spełniającej wszystkie ograniczenia: $\{\mathbf{X}_i = \mathbf{d}_i : 1, \dots, N\}$, gdzie $\mathbf{d}_i \in \mathbf{D}_i$. Końcowym stanem zupełnym $\mathbf{s} \in S$ jest zatem taki stan, dla którego zachodzi $\mathbf{s} = \{\mathbf{X}_1 = \mathbf{d}_1, \dots, \mathbf{X}_N = \mathbf{d}_N\}$ i spełnione są wszystkie ograniczenia.

Typowo, rozwiązywanie CSP polega na przeszukiwaniu w głąb z nawrotami drzewa decyzyjnego, w celu przypisania alternatywnych wartości poszczególnym zmiennym decyzyjnym. Strategia ta ma następujące własności: (a) *stan początkowy* jest stanem pustym a zmiennym decyzyjnym nie są przypisane żadne wartości; (b) *rozszerzenie stanu* – polega na generowaniu możliwych następników aktualnego stanu \mathbf{s} przez przypisanie wartości zmiennym, którym dotychczas ich nie przydzielono; (c) *test spójności* – polega na sprawdzeniu, czy naruszono zbiór ograniczeń dla istniejących przydziałów wartości poszczególnym zmiennym; (d) *nawrót* – odwrócenie ostatniej akcji przypisania wartości zmiennej i wykonanie przydziału innej wartości; (e) *test celu* – sprawdzenie, czy stan aktualny jest stanem końcowym. Rozwiązaniem jest każde kompletne przypisanie (ścieżka od węzła początkowego drzewa decyzyjnego do węzła końcowego na głębokości N) spełniające wszystkie ograniczenia.

Ze względu na przyjętą dekompozycję zadania planowania rozmieszczania podpór należy rozwiązać trzy hierarchiczne podproblemy CSP, odpowiednio dla głowic (*Head-CSP*), baz mobilnych (*Base-CSP*) oraz manipulatorów mobilnych (*PKM-CSP*). Do rozwiązania każdego problemu stosuje się ten sam algorytm numeryczny – sterowany ograniczeniami algorytm przeszukiwania w głąb z nawrotami, inne są natomiast zmienne, zakresy wartości oraz ograniczenia. W proponowanym podejściu, standardowy algorytm przeszukiwania w głąb z nawrotami, został zmodyfikowany w dwóch istotnych elementach:

- (1) dopuszcza się rozwiązanie częściowe – należy pamiętać dotychczasowe najlepsze rozwiązanie częściowe, gdyż może nie być rozwiązaniem zupełnego;
- (2) tworzy się hierarchię trzech procesów CSP – w warunkach stopu procesu nadrzędnego jest wywoływany podrzędny proces CSP.

Wartości określające pozycje głowic, baz i PKM mogą przyjmować wartości rzeczywiste z określonych przedziałów. Zastosowanie dyskretnego CSP wymaga zatem odpowiedniej dyskretyzacji

wartości zmiennych decyzyjnych. Ścieżka będąca rozwiązaniem zadania planowania składa się z list stanów baz, PKM i głowic $\{\mathbf{h}_1, \mathbf{b}_1, \mathbf{m}_1, \dots, \mathbf{h}_N, \mathbf{b}_N, \mathbf{m}_N\}$ dla każdego agenta.

Strategia przydzielania pozycji głowic

Danymi wejściowymi dla problemu *Head-CSP* są listy segmentów i zaklasyfikowanych wierzchołków opisujących kontur, wzdłuż którego mają być rozmieszczone głowice. Liczba zmiennych decyzyjnych jest równa liczbie segmentów konturu. Każda zmienna decyzyjna odpowiada sekwencji pozycji głowic dla pojedynczego segmentu $\mathbf{X}_i = \{\mathbf{h}_{i1}, \dots, \mathbf{h}_{in}\}$. W jednym kroku przydziału wartości danej zmiennej decyzyjnej jest przydzielana cała sekwencja pozycji głowic. Jeśli kontur zwiera wierzchołek typu 0, to planowanie rozpoczyna się od segmentu, dla którego jest on początkiem. Jeśli poszukiwanie sekwencji pozycji głowic spełniających wszystkie ograniczenia kończy się niepowodzeniem, to zmienia się wierzchołek startowy. W przypadku, gdy nie jest możliwe wyznaczenie kompletnego planu dla całego konturu, wówczas kontur jest dzielony na fragmenty, dla których są tworzone odrębne plany częściowe. Wymaga to jednakże przerwania obróbki po zakończeniu danego fragmentu, a następnie jej wznowienie od początku kolejnego fragmentu. Po połączeniu planów częściowych uzyskuje się kompletny plan dla całego konturu. Podczas planowania jest wybierana najmniejsza możliwa liczba pozycji podpór, przy której są spełnione ograniczenia geometryczne.

Głowice są rozmieszczane w taki sposób, aby możliwe największa część boku głowicy dawała podparcie wzdłuż linii konturu. W przypadku, gdy n głowic jest zbyt dużo, a $n - 1$ zbyt mało, do podparcia danego segmentu konturu, wybierana jest prosta strategia przesuwania głowic. Polega ona na odsuwaniu co drugiej głowicy od konturu na odległość d_{max} i przybliżaniu pozostałych na odległość d_{min} , dzięki czemu możliwe jest rozmieszczenie n głowic wzdłuż danego segmentu. Zazwyczaj dla każdego segmentu jest dużo alternatywnych planów, dla których odległość boku głowicy od konturu mieści się w przedziale $[d_{min}, d_{max}]$ oraz odległości między kolejnymi głowicami mieszczą się w zakresie $[0, D_{max}]$. W przypadku, gdy do znalezienia dopuszczalnego planu konieczne jest obracanie głowic, rozpoczynamy od małego kąta $\Delta\varphi = 1^\circ$ zwiększając go o 1° w kolejnych krokach aż do osiągnięcia 30° . Po przydzieleniu głowic dla jednego segmentu konturu sprawdzamy, czy można znaleźć dopuszczalne pozycje baz i PKM, które zapewniają osiągnięcie obliczonych pozycji głowic. Hipotetyczny plan dla głowic wymaga weryfikacji w postaci istnienia odpowiedniego planu dla mobilnych baz i PKM. Weryfikacja jest wykonywana kolejno dla każdego segmentu, czyli po przydzieleniu każdej zmiennej decyzyjnej $\mathbf{X}_i = \{\mathbf{h}_{i1}, \dots, \mathbf{h}_{in}\}$ odpowiednich wartości.

Strategia przydzielania pozycji i akcji przemieszczenia baz

Celem tego etapu jest znalezienie sekwencji pozycji baz ${}^W B_i$ dla obliczonych w pierwszym etapie pozycji głowic ${}^W H_i$. Plan jest poszukiwany z wykorzystaniem algorytmu *Base-CSP*, dla którego danymi wejściowymi jest sekwencja pozycji głowic. Przestrzeń osiągalnych pozycji bazy jest skończona ze względu na ograniczoną liczbę złączy dokujących w ławie. Ta sama pozycja głowicy ${}^W H_i$ może być osiągnięta przy różnych pozycjach bazy i konfiguracjach PKM. Z listy dopuszczalnych pozycji baz jest wybierana ta, która wymaga minimalnego przemieszczenia z bieżącej pozycji do następnej. Jeśli zadana pozycja głowicy może być osiągnięta bez konieczności przedstawiania bazy, to takie rozwiązanie jest wybierane. Taka sytuacja ma miejsce przy obróbce

wybranych elementów kadłuba samolotu, gdy 2-3 kolejne pozycje głowice są osiągalne z jednej pozycji bazy. Jest to możliwe m.in. dzięki nadmiarowi stopni swobody całego agenta. W trakcie ruchu bazy nie mogą wystąpić kolizje z drugim agentem, który jest wówczas nieruchomy. Wykrywanie kolizji będące zazwyczaj najbardziej czasochłonną czynnością algorytmów planowania ruchu, w rozważanym przypadku jest istotnie uproszczone. Ponieważ bazy mobilne mają przekrój kołowy i poruszają się na płaszczyźnie unikanie kolizji sprowadza się do spełnienia warunku nieprzecinania się dwóch okręgów:

$$\forall_{[{}^0t_{i,B}^F, {}^1t_{i,B}^S]} C({}^W B_i(t), r) \cap C({}^W B_j, r) = \emptyset,$$

gdzie $C({}^W B_i(t), r)$ oznacza okrąg o promieniu r i środku w początku poruszającego się układu ${}^W B_i(t)$, zaś $C({}^W B_j, r)$ jest okręgiem opisującym drugą bazę pozostającą w tym czasie w spoczynku, a ${}^0t_{i,B}^F$ jest chwilą rozpoczęcia ruchu od poprzedniej pozycji, a ${}^0t_{i,B}^S$ chwilą rozpoczęcia podparcia w następnej pozycji. Wynikiem planowania jest ciąg pozycji baz ${}^W B_i$ spełniający wszystkie ograniczenia.

Wyznaczanie ścieżek ruchu dla manipulatora równoległego

Mając wyznaczone częściowe plany dla baz i głowic pozostaje obliczenie konfiguracji manipulatora równoległego (PKM) oraz ścieżek przejścia między nimi. Para pozycji głowica-baza wyznacza lokalizację końcówki PKM i jego podstawy. Rozwiązując odwrotne zadanie kinematyki dla PKM oblicza się jego konfigurację i sprawdza się, czy dla danej pozycji bazy i związanej z nią podstawy PKM zadana pozycja głowicy będzie osiągalna. Zadana pozycja głowicy musi zatem należeć do przestrzeni roboczej PKM. Ścieżka przejścia dla PKM między dwiema kolejnymi pozycjami podparcia jest określona przez cztery punkty (konfiguracje) pośrednie i składa się z następujących ruchów:

- opuszczenie głowicy w dół na odległość równą wysokości głowicy,
- wycofanie głowicy do pozycji bezpiecznej (tzn. rzut głowicy na płaszczyznę bazy musi zawierać się w okręgu $C({}^W B_i(t), r)$),
- obrót podstawy PKM kompensujący obrót bazy,
- przemieszczenie głowicy do lokalizacji dokładnie pod pozycją docelową
- podniesienie głowicy do pozycji docelowej ${}^W H_i$.

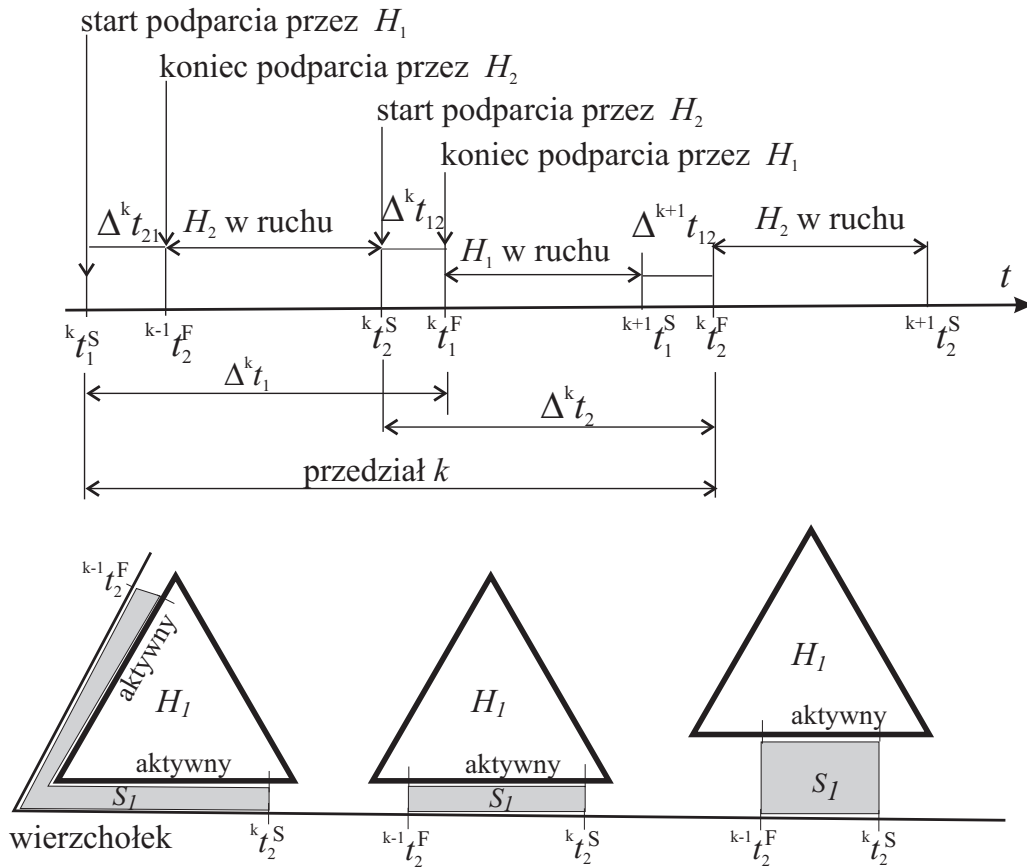
Po zakończeniu sukcesem tego etapu otrzymujemy częściowy plan w postaci ścieżek ruchu dla trzech składowych agenta, czyli plan dla pojedynczego segmentu. Jeśli uda się wyznaczyć ścieżki dla wszystkich pozostałych segmentów otrzymujemy kompletną sekwencję ścieżek.

Planowanie trajektorii ruchu

Ostatnim etapem planowania rozmieszczania i relokacji podpór jest parametryzacja czasowa obliczonych ścieżek ruchu. Przy obliczeniu trajektorii uwzględnia się czasy rozpoczęcia i zakończenia odpowiednich faz podparcia dla poszczególnych części składowych agenta:

- ${}^k t_{i,j}^S$ – chwila czasowa, w której powinna rozpocząć się akcja podpierania dla j -tej składowej agenta (gdzie $j \in \{H, B, M\}$) robota i , $i = 1, 2$, w kroku $k = 1, \dots, K$,
- ${}^k t_{i,j}^F$ – chwila czasowa, w której powinna zakończyć się akcja podpierania dla j -tej składowej robota i w kroku k .

Czasy te są różne dla poszczególnych składowych agenta, przy czym spełnione są następujące zależności ${}^k t_{i,B}^S < {}^k t_{i,M}^S < {}^k t_{i,H}^S$ oraz ${}^k t_{i,H}^F < {}^k t_{i,M}^F < {}^k t_{i,B}^F$. Taka sekwencja chwil wynika z przyjętego ze względów bezpieczeństwa założenia, że baza i PKM nie mogą poruszać się jednocześnie. W trakcie ruchu manipulatora przemieszczającego głowicę do pozycji docelowej baza pozostaje w spoczynku, głowica utwardza się dopiero wówczas, gdy manipulator (zatem i głowica) osiągnie pozycję docelową. Ponieważ znana jest pełna trajektoria narzędzia oraz rozmieszczenie głowic podpierających można określić strefy S_1 podparcia dla każdej głowicy. Odpowiednie przedziały czasowe i strefy podparcia dla dwóch agentów pokazano na rys. 9. Przyjmuje się,



Rys. 9: Przedziały czasowe i strefy podparcia

że każdy przedział czasowy rozpoczyna się z chwilą podparcia przez głowicę H_1 i kończy się w chwili zakończenia podparcia przez głowicę H_2 , zatem przedział k -ty rozpoczyna się w chwili ${}^k t_1^S$ i kończy w ${}^k t_2^F$. Można wyróżnić fazy, gdy podpira tylko jedna głowica oraz przedziały, w których dwie głowice jednocześnie podpierają detal. Dla każdego przedziału wyróżnia się trzy okresy $\Delta^k t_{21}$, $\Delta^k t_{12}$ oraz $\Delta^{k+1} t_{12}$ podparcia przez dwie głowice.

$$\Delta^k t_{21} = {}^{k-1} t_2^F - {}^k t_1^S, \quad \Delta^k t_{12} = {}^k t_1^F - {}^k t_2^S, \quad \Delta^{k+1} t_{12} = {}^k t_2^F - {}^{k+1} t_1^S$$

Głowica i -ta pozostaje nieruchoma w przedziałach czasowych: $\Delta^k t_i = {}^k t_i^F - {}^k t_i^S$, $i = 1, 2$. Poszczególne czasy muszą być odpowiednio dobrane, aby plan mógł być wykonany. Na podstawie analizy sił podpierających można wyróżnić obszary podparcia \mathcal{S}_1 przez pojedynczą głowicę (rys. 9). Obszary te są zdefiniowane przez długość boku obszaru mierzoną wzdłuż konturu podparcia i wyznaczone po obliczeniu pozycji głowic. W obszarach wierzchołków głowica może podbierać dwa boki konturu.

Każda pozycja podparcia ${}^W H_i$ jest określona przez trzy zmienne wektorowe \mathbf{h}_i , \mathbf{b}_i , i \mathbf{p}_i , gdzie:

- parametry \mathbf{h}_i i -tej głowicy są zdefiniowane jako:

$$\mathbf{h}_i = [{}^k t_{i,H}^S, {}^k t_{i,H}^F, {}^W H_i] \quad (31)$$

- parametry \mathbf{b}_i i -tej bazy mobilnej jako:

$$\mathbf{b}_i = [{}^k t_{i,B}^S, {}^k t_{i,B}^F, {}^W B_i] \quad (32)$$

- parametry \mathbf{p}_i i -tego manipulatora równoległego (PKM) jako:

$$\mathbf{p}_i = [{}^k t_{i,M}^S, {}^k t_{i,M}^F, {}^B M_i] \quad (33)$$

Po obliczeniu pozycji podparcia oraz trajektorii przejścia dla składowych obu agentów otrzymujemy kompletny plan rozmieszczenia podpór dla danego detalu i procesu obróbczego.

4.3 Implementacje algorytmów planowania i sterowania z użyciem programowej struktury ramowej MRROC++ [H10, H11]

W przedstawionych powyżej różnych zadaniach manipulacji oprogramowanie układów sterowania robotów stworzono w środowisku MRROC++.

4.3.1 Struktura ramowa dla tworzenia oprogramowania sterującego robotami

Programowa struktura ramowa MRROC++ (Multi-Robot Research Oriented Controller utilizing C++) składa się z biblioteki modułów (klas, procesów, wątków, funkcji i procedur) oraz wzorców ich wykorzystania [H10, H11], [P47]. Jest to zatem uniwersalne narzędzie do tworzenia programów sterujących zarówno dla pojedynczego robota, jak również systemów wielorobotowych.

Struktura MRROC++ oraz przykłady sterowników programowych dla różnych typów robotów stworzonych z użyciem MRROC++ są omówione w punkcie 5.1.2.

4.3.2 Implementacja *planera* i układu sterowania dla układania kostki Rubika [H10, H11]

Do wykonania zadania układania kostki Rubika wykorzystano system składający się z dwóch zmodyfikowanych robotów IRp-6 o 6 stopniach swobody każdy, wyposażonych w dwupalczaste chwytaki oraz zestawu czujników w postaci kamer wizyjnych, czujników sił i momentów sił zamocowanych w kiściach manipulatorów i czujników dotykowych w szczękach chwytaków. Umiejętności manipulacyjne zaprogramowano jako moduły sterowników programowych stworzonych

za pomocą programowej struktury ramowej MRROC++ [24, H6]. Implementacja powyższych umiejętności w strukturze MRROC++ nie nastęca istotnych trudności, gdyż stanowi naturalne rozszerzenie dotychczasowych jej możliwości, zaś do tworzenia umiejętności wykorzystano już istniejące elementy struktury MRROC++. Oprogramowanie stanowiące sterownik programowy systemu dwurobotowego składa się z wielu współbieżnie wykonujących się wielowątkowych procesów, które działają pod kontrolą systemu operacyjnego czasu rzeczywistego QNX Neutrino [H10]. W rozważanym zadaniu, ze względu na duże wymagania obliczeniowe, poszczególne procesy sterownika wykonują się równoległe na kilku komputerach klasy PC połączonych siecią Ethernet. Jest to warstwowa struktura hierarchiczna z podziałem funkcji pomiędzy poszczególne warstwy. Warstwa dolna realizuje obsługę sprzętu (efektorów i czujników) oraz wykonuje zlecenia warstw wyższych. Procesy Effector Driver Process (EDP) są właściwymi sterownikami robotów i realizują zlecenia przesyłane z procesów Effector Control Process (ECP). Procesy czujników wirtualnych Virtual Sensor Process (VSP) realizują obsługę czujników eksteroreceptywnych.

Warstwa środkowa wykonuje właściwe zadanie, czyli wszystkie algorytmy składające się na realizację zadania układania kostki. W implementacji sterownika dedykowanego realizacji zadania układania kostki Rubika procesy ECP są przezroczyste i jedynie przesyłają zlecenia z procesu koordynatora Master Process (MP). Warstwa górna (proces UI) jest interfejsem użytkownika.

4.3.3 Planowanie ścieżki dla systemu z zamkniętym łańcuchem kinematycznym [H7]

W celu weryfikacji przydatności proponowanego algorytmu planowania dopuszczalnych ścieżek rozpatrzono zadanie polegające na przeniesieniu sztywnej belki wspólnie przez dwa manipulatory robotów IRp-6, każdy o pięciu stopniach swobody [Hu5]. Utworzony układ z zamkniętym łańcuchem kinematycznym ma tylko cztery stopnie swobody, co istotnie ogranicza zbiór dopuszczalnych pozycji belki. Dzięki aproksymacji funkcjami sklejanymi z wielomianów trzeciego stopnia obliczone ścieżki dla ogniw są dostatecznie gładkie. Ścieżki we współrzędnych konfiguracyjnych obu robotów są ze sobą ściśle zsynchronizowane.

Dokładność spełnienia więzów ruchu, wynikająca ze spełnienia ograniczeń równościowych w zadaniu optymalizacji nieliniowej, dla rozpatrywanego przypadku jest rzędu 10^{-8} , co w zestawieniu z powtarzalnością pozycjonowania robota IRp-6 wynoszącą ± 0.2 mm, czyni obliczone ścieżki ruchu praktycznie stosowanymi w układzie sterowania systemem dwurobotowym. Czas obliczeń zależy od kroku dyskretyzacji i na komputerze klasy PC (z procesorem Intel Q6600 2.4 GHz i pamięcią operacyjną 1GB) dla liczby przedziałów dyskretyzacji 25 jest poniżej jednej sekundy.

4.3.4 Układ sterowania rekonfigurowalnych robotów mocujących [H5]

Do realizacji planów wykorzystano układ sterowania mobilnymi robotami-podporami. Oprogramowanie sterujące wykonano wykorzystując programową strukturę ramową MRROC++ [H5], [Hu11]. Układ sterowania dla systemu rekonfigurowalnych podpór bazuje na podejściu agentowym i ma strukturę hierarchiczną. Każdy robot-podpora jest sterowany przez trzy agenty, każdy z nich jest dedykowany poszczególnym modułom robota: bazie mobilnej, manipulatorowi równoległemu i głowicy. W układzie sterowania jest również odrębny agent sterujący aktywnymi

elementami ławy roboczej. Działanie wszystkich agentów sterujących nadzoruje wyróżniony agent-koordynator. Dla oprogramowania układu sterowania została opracowana formalna specyfikacja opisująca zachowania agentów w postaci odpowiednich funkcji przejścia [27].

W celu praktycznej weryfikacji wykonalności i poprawności obliczonych planów przeprowadzono testy działania systemu mocowań z rekonfigurowalnymi podporami w zakładach lotniczych Piaggio Aero, Finale Ligure, Włochy. Wyznaczono plany rozmieszczenia podpór dla różnych elementów wchodzących w skład poszycia kadłuba samolotu. Wykonane eksperymenty polegające na wykonaniu obróbki maszynowej w postaci frezowania i wiercenia otworów w elementach kadłuba samolotu potwierdziły poprawność i praktyczną przydatność zarówno generowanych planów, jak też ich realizacji przez układ sterujący systemem rekonfigurowalnych robotów-podpór.

Publikacje powiązane z jednotematycznym cyklem:

- [Hu1] W. Kasprzak, W. Szynkiewicz, Ł. Czajka. Rubik's Cube Reconstruction from Single View for Service Robots. *Machine GRAPHICS & VISION*, 2006, 25(3-4):451–459.
- [Hu2] W. Kasprzak, W. Szynkiewicz, C. Zielińska, T. and Zieliński, P. Trojanek, T. Kornuta, T. Winiarski, M. Wałęcki. Wielorobotowa rekonfigurowalna forma mocująca obrabiane detale – program sterujący. *Pomiary Automatyka Robotyka (PAR)*, 2013, (3):96–102.
- [Hu3] W. Szynkiewicz. Planowanie manipulacji dwuręcznej. In: *Postępy robotyki. Sterowanie robotów z percepcją otoczenia* Red. K. Tchoń, pp. 187–196. Wydawnictwa Komunikacji i Łączności, Warszawa 2005.
- [Hu4] W. Szynkiewicz. Manipulacja dwuręczna w zadaniu układania kostki Rubika. In: *Postępy Robotyki: Sterowanie, percepcja i komunikacja* Red. K. Tchoń, wol. 1, pp. 247–256. Wydawnictwa Komunikacji i Łączności, Warszawa 2006.
- [Hu5] W. Szynkiewicz, J. Błaszczak, K. Malinowski. Optymalizacja w zastosowaniu do planowania ruchu ściśle współpracujących robotów. *Pomiary Automatyka Robotyka (PAR)*, 2009, (2):71–89.
- [Hu6] W. Szynkiewicz, W. Kasprzak, T. Zielińska. Planowanie rozmieszczenia ruchomych podpór jako dyskretny problem spełniania ograniczeń. *Pomiary Automatyka Robotyka (PAR)*, 2012, 16(2):344–350.
- [Hu7] W. Szynkiewicz, W. Kasprzak, T. Zielińska, D. Zlatanov. Planowanie rozmieszczeń ruchomych podpór przy obróbce przedmiotów o dużych rozmiarach. In: *Problemy Robotyki* Red. K. Tchoń, C. Zieliński, wol. 1 serii *Prace Naukowe Elektronika*, pp. 321–330. Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej 2010.
- [Hu8] T. Zielińska, W. Kasprzak, C. Zieliński, W. Szynkiewicz. Distributing the supporting heads for robotized machining. In: *Advances on Theory and Practice of Robots and Manipulators* Red. M. Ceccarelli, V.A. Glazunov, Vol. 22 series *Mechanisms and Machine Science*, pp. 509–517. Springer International Publishing 2014.

- [Hu9] C. Zieliński, W. Szynkiewicz, T. Winiarski, W. Czajewski, M. Staniak. Układanie kostki Rubika jako zadanie testujące zdolności robota usługowego. In: *Postępy Robotyki: Systemy i współdziałanie robotów. Proceedings* Red. K. Tchoń. Wydawnictwa Komunikacji i Łączności, 2006. vol. 2, pp. 237–246.
- [Hu10] C. Zieliński, W. Szynkiewicz, T. Winiarski, M. Staniak. Rubik’s cube puzzle as a benchmark for service robots. In: *12th IEEE International Conference on Methods and Models in Automation and Robotics, (MMAR). Proceedings*, August 28-31, 2006, pp. 579–584.
- [Hu11] C. Zieliński, P. Trojanek, T. Kornuta, T. Winiarski, M. Wałęcki, W. Kasprzak, W. Szynkiewicz, T. Zielińska. Wielorobotowa rekonfigurowalna forma mocująca obrabiane detale – układ sterowania. *Pomiary Automatyka Robotyka (PAR)*, 2013, (2):79–85.

5 Omówienie pozostałej działalności naukowo-badawczej

5.1 Autorstwo lub współautorstwo publikacji naukowych innych niż przedstawione w punkcie 2

Poniższy opis dotyczy tematyki oraz najważniejszych wyników moich prac naukowo-badawczych niepowiązanych bezpośrednio z problematyką poruszaną w pracach należących do jednotematycznego cyklu publikacji. Prace zostały podzielone na następujące grupy tematyczne:

- synteza chwytów i ich realizacja za pomocą chwytaków wielopalczastych;
- projektowanie i implementacja wysokopoziomowych sterowników o strukturze otwartej dla systemów jedno- i wielorobotowych oraz ich aplikacje;
- planowanie trajektorii ruchu dla współpracujących manipulatorów robotów;
- sterowanie i autonomiczna nawigacja kołowych robotów mobilnych.

5.1.1 Synteza chwytów i ich realizacja za pomocą chwytaków wielopalczastych

Zagadnienia syntezy i realizacji chwytu obiektów są przedmiotem moich zainteresowań badawczych ze względu na fakt, że chwyt jest nieodłącznym elementem każdego zadania manipulacji chwytniej. W praktyce pozycja chwytanego obiektu jest wyznaczana na podstawie danych pomiarowych (np. z systemu wizyjnego), i ze względu na różne źródła błędów, obliczona pozycja może istotnie różnić się od rzeczywistej. W pracach [P20,P21] zaproponowałem metodę syntezy chwytu przy niepewności pozycji obiektu i założeniu, że obiekt może przemieszczać się w trakcie chwytania. Istotą proponowanego podejścia jest wykorzystanie symulacji procesu chwytania z uwzględnieniem dynamiki obiektu do wyznaczenia domkniętych siłowo, stabilnych chwytów. Zastosowanie kwaternionów dualnych jako jednolitej reprezentacji położenia i orientacji obiektu umożliwiło wykorzystanie odpowiednich rozkładów prawdopodobieństwa do losowego generowania pozycji obiektu i symulacji wykonania chwytu. Jakość każdego chwytu jest oceniana na podstawie trzech kryteriów określających odporność chwytu na zakłócenia zewnętrzne, orientację chwytaka względem obiektu oraz przemieszczenie obiektu w trakcie chwytania względem pozycji początkowej.

W pracach [P10,P25,P34] przedstawiamy budowę mechaniczną, układ sterowania oraz syntezę chwytu dla chwytaka trójpalczastego. Chwytnak i jego układ sterowania zaprojektowano i zbudowano w Politechnice Warszawskiej. Konstrukcja chwytaka z wielosegmentowymi palcami ze sprężyniętymi sąsiednimi palczkami umożliwia realizację obejmujących chwytów kształtowych z wieloma punktami kontaktu. Nowatorskim rozwiązaniem w konstrukcji chwytaka jest zastosowanie w każdym przegubie dwóch czujników kąta obrotu bezwzględne i względne oraz wprowadzenie podatności. Zastosowanie czujników umożliwiło obliczenie wartości momentu obciążenia zewnętrznego działającego na szczękę chwytaka. Oprogramowanie układu sterowania chwytakiem stworzono za pomocą programowej struktury ramowej MRROC++. W algorytmie syntezy chwytów dla wybranych obiektów zakłada się znajomość modeli geometrycznych obiektów, współczynnika tarcia, środka masy oraz materiału, z którego wykonano obiekt. Algorytm bazuje na wykorzystaniu określonych postur chwytaka przy chwytaniu znanych obiektów,

których model 3D jest przybliżany jednym z kształtów podstawowych (brył prostych) [P25]. Wprowadzenie podatności w przegubach palców i pomiar momentów sił pozwalają na uzyskanie stabilnych chwytów odpornych na zakłócenia zewnętrzne. W artykule [P48] przedstawiamy strukturę układu sterowania manipulatorem z chwytakiem trójpalczastym stworzoną z użyciem struktury ramowej MRROC++. Manipulator robota i jego chwytak są sterowane przez dwa dedykowane agenty upostaciowione współpracujące ze sobą za pośrednictwem agenta koordynatora.

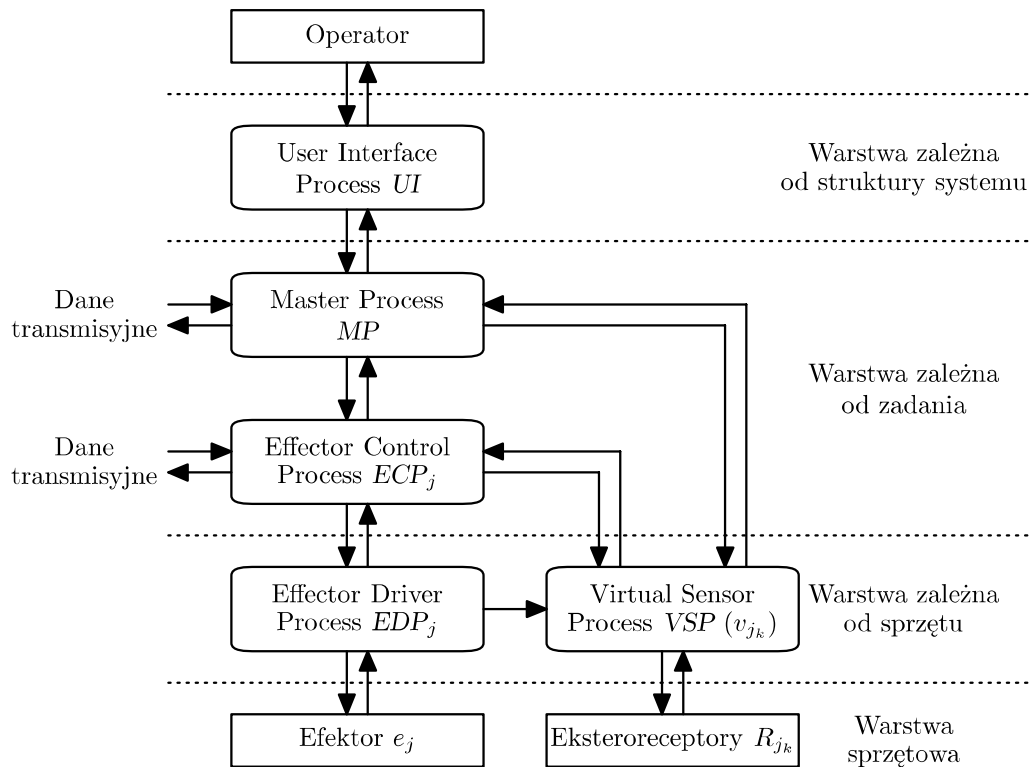
5.1.2 Projektowanie i implementacja wysokopoziomowych sterowników o strukturze otwartej dla systemów jedno- i wielorobotowych

Programowe struktury ramowe są obecnie jednymi z częściej stosowanych narzędzi do tworzenia oprogramowania sterowników robotów. Prace badawcze nad rozwojem tego typu środowisk są prowadzone od wielu lat w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej [P31, P32, P37, P38, P40, P41, P43, P44, P47]. Motywacją dla tych badań była potrzeba szybkiego tworzenia oprogramowania dla różnego typu robotów wykonujących różnorodne zadania. Istotnym powodem popularności struktur ramowych jest możliwość powtórzonego użycia sprawdzonego kodu oraz istnienie gotowej powłoki programu (części niezależnej od zadania i sprzętu), która zapewnia m.in. komunikację pomiędzy poszczególnymi modułami sterownika i obsługę błędów. Ponadto cechą charakterystyczną struktur ramowych jest wykorzystanie wzorców, w szczególności wzorców architektury i wzorców projektowych.

W pierwotnej wersji, opracowanej przez nas struktury ramowej zwanej MRROC (*Multi-Robot Research Oriented Controller*), wykorzystywaliśmy podejście proceduralne do programowania systemów robotycznych [P31, P32]. Językiem bazowym struktury MRROC był język C. W kolejnej wersji struktury ramowej nazwanej MRROC++ wykorzystaliśmy podejście obiektowe przy tworzeniu oprogramowania, a językiem bazowym był język C++. Sterownik programowy tworzony w środowisku MRROC++ składa się z wielu współbieżnie działających procesów wykonujących się pod kontrolą systemu operacyjnego czasu rzeczywistego QNX. Każdy proces może składać się z wielu wątków. Wszystkie procesy tworzone z użyciem MRROC++ mają jednakową strukturę ogólną i składają się z powłoki oraz jądra. W powłoce jest realizowana komunikacja międzyprocesowa wewnątrz agenta oraz komunikacja między agentami, a ponadto powłoka jest odpowiedzialna za obsługę wykrytych błędów. Natomiast jądro zawiera kod wykonywanego zadania. Powłoka stanowi niezmienną część procesu, natomiast jądro jest częścią wymienną zależną od zadania bądź wykorzystywanego sprzętu, a więc dostarczana jest przez twórcę systemu (użytkownika MRROC++). Rozszerzenie struktury MRROC++ o nowe funkcje wymaga napisania nowych modułów w języku C++.

Specyfikacja struktury MRROC++ powstała w wyniku prac profesora Cezarego Zielińskiego poświęconych formalnemu opisowi systemów wieloagentowych [26–28].

Oprogramowanie sterujące systemami jedno- i wielorobotowymi (wieloefaktorowymi – efektozem może być również inne urządzenie) tworzone z wykorzystaniem struktury MRROC++ mają strukturę wielowarstwową. Przykładową strukturę oprogramowania sterownika przedstawia rysunek 10. Jest to wieloagentowa, warstwowa struktura z podziałem funkcji pomiędzy poszczególne warstwy. Każdej z warstw są przyporządkowane ściśle określone funkcje – wyróżniają się procesy zależne od sprzętu oraz zależne od zadania. Warstwa dolna realizuje obsługę sprzę-



Rys. 10: Struktura sterownika stworzonego w środowisku MRROC++

tu (efektorów i czujników) oraz wykonuje zlecenia warstw wyższych. Procesy Effector Driver Process (EDP) są właściwymi sterownikami robotów i realizują zlecenia przesyłane z procesów Effector Control Process (ECP). MP – Master Process koordynuje pracę całego systemu w tym wszystkich efektorów systemu. Proces ten może być uśpiony – wówczas procesy ECP nie są koordynowane. Proces UI realizuje warstwę komunikacji z użytkownikiem. Procesy VSP są odpowiedzialne za obsługę eksteroreceptorów.

Struktura ramowa MRROC++ była i jest z powodzeniem stosowana do tworzenia sterowników robotów przemysłowych i usługowych, działających pojedynczo, a także w zespołach wykonujących bardzo różne zadania. Wykorzystaliśmy MRROC++ do budowy sterownika dla prototypowego robota szeregowo-równoległego o podwyższonej sztywności wykonującego operacje frezowania i szlifowania [P11, P36, P38, P43]. W pracy [P42] przedstawiliśmy budowę sterownika robota Polycrank z napędami bezpośrednimi bez ograniczeń zakresu ruchu w przegubach. Wykorzystując strukturę MRROC++ opracowaliśmy sterownik robota z funkcjami pomiarowymi umożliwiającymi identyfikację tarcia [P35] oraz sterownik robota usługowego wykorzystujący dane pomiarowe z różnych czujników (kamer wizyjnych, czujników sił, czujników dotykowych) [P46]. Sterowniki stworzone w MRROC++ współpracują ze sterownikami niskopoziomymi napędów robota [P3].

5.1.3 Planowanie trajektorii ruchu dla współpracujących manipulatorów robotów

Planowanie ruchu robotów było przedmiotem moich prac badawczych zarówno przed doktoratem [P12–P14, P22, P26, P27], jak również po uzyskaniu stopnia doktora [P15–P17, P23], [Hu5]. W pracy [P23] przedstawiłem algorytm planowania trajektorii ruchu dla dwóch ściśle współpra-

cujących robotów. Algorytm jest dwufazowy, w pierwszej fazie obliczane są dopuszczalne ścieżki ruchu we współrzędnych wewnętrznych dla obu robotów, w drugiej fazie są obliczane trajektorie ruchu poprzez skalowanie czasowe wcześniej obliczonych ścieżek z uwzględnieniem ograniczeń na dopuszczalne prędkości i przyspieszenia w poszczególnych przegubach robotów. Obliczone trajektorie ruchu posłużyły jako przebiegi zadane dla układu sterowania dwóch robotów ściśle współpracujących robotów. Błędy odtwarzania zadanych trajektorii ruchu były poniżej 0.05 rad, co umożliwiło wykonanie zadanych trajektorii bez naruszania ograniczeń. W artykule [P15] rozważałem problem planowania trajektorii ruchu dla dwóch robotów zginających wspólnie arkusz cienkiej elastycznej blachy. Dwuwymiarowy problem zginania arkusza blachy został sprowadzony do jednowymiarowego problemu zaginania belki. Do budowy modelu belki zastosowałem metodę elementów skończonych. Trajektorie ruchu manipulatorów są wyznaczane w taki sposób, aby minimalizować siły i momenty sił wywierane na belkę. W pracy [P16] przedstawiłem sformułowanie problemu planowania ruchu i sterowania systemami wielorobotowymi z wykorzystaniem teorii gier. Zaproponowałem opis zadań wymagających różnego stopnia współpracy jako różnych postaci gry dynamicznej poczynając od gry o sumie zerowej, gdy nie ma możliwości współpracy robotów, poprzez gry niekooperatywne i kooperatywne, kończąc na zadaniu, w którym roboty pracują jako zespół realizujący jeden wspólny cel. W artykułach [P17], [Hu5] przedstawiłem różne podejścia do zadania planowania ruchu mechanizmów zawierających zamknięte pętle kinematyczne. Rozważane były trzy główne podejścia: algorytmy zupełne, algorytmy losowe oraz algorytmy wariacyjne.

5.1.4 Sterowanie i autonomiczna nawigacja kołowych robotów mobilnych

Kolejnym obszarem moich zainteresowań badawczych są zagadnienia związane ze sterowaniem oraz autonomiczną nawigacją robotów mobilnych [P1, P4–P9, P24, P29, P33, P45]. W pracach [P1, P24] przedstawiliśmy strukturę sprzętową i układ sterowania robota mobilnego Elektron zbudowanego w Politechnice Warszawskiej (rys. 11). Dzięki modułowej konstrukcji umożliwia-



Rys. 11: Robot mobilny Elektron

jącej łatwą zmianę wyposażenia robota, platforma ta była i jest wykorzystywana w różnych projektach badawczych i pracach dyplomowych studentów.

W artykułach [P4–P6] zaproponowaliśmy algorytm samolokalizacji robota wykorzystujące kolorowe cechy naturalnych znaczników występujących w środowisku robota. Do lokalizacji wykorzystaliśmy różne wersje filtra cząsteczkowego będącego implementacją filtra Bayesa z rozkładami gęstości prawdopodobieństwa aproksymowanymi zbiorami dyskretnych próbek losowych. Zbadaliśmy jakość procesu lokalizacji przy użyciu metod analizy obrazów wykorzystujących trzy różne cechy globalne obrazów kolorowych. Rozważaliśmy wektory pomiarowe składające się odpowiednio: z wartości średnich i odchyłeń standardowych trzech składowych kolorowych obrazu, pierwszych sześciu składowych szybkiej transformaty Fouriera obrazu oraz wartości trzech dominujących składowych kolorowych obrazu i ich gęstości. Na podstawie badań eksperymentalnych pokazaliśmy, że dla naturalnych scen ze zmiennym oświetleniem i niewielkich błędów pomiarów odometrycznych, wykorzystanie nawet niewielkiego zbioru cech kolorowych, dostarczających jedynie globalnej informacji o danej scenie, umożliwia wiarygodną samo-lokalizację.

W [P7] zadanie jednoczesnej samolokalizacji i budowy mapy (*SLAM*) sformułowaliśmy jako problem spełnienia ograniczeń (*CSP*). Zastosowaliśmy podejście kinematyczne, w którym dane pomiarowe z czujników w postaci pozycji robota i położenia znaczników są wykorzystywane do tworzenia grafu ograniczeń. Nowym elementem naszego podejścia jest brak założenia, że możemy na podstawie pomiarów jednoznacznie zidentyfikować wszystkie znaczniki, czyli że znana jest dokładna zależność (korespondencja) między znacznikami i ich obserwacjami. Do rozwiązania problemu *CSP* zastosowaliśmy techniki obliczeń przedziałowych.

Praca [P29] dotyczy problemu budowy i upraszczania map 3D pomieszczeń zamkniętych z użyciem pomiarów ze skanera laserowego. Opracowaliśmy algorytm upraszczania i poprawy jakości mapy 3D wykorzystujący metodę rozszerzania obszarów. Pierwotna mapa w postaci siatki trójkątów podlega uproszczeniu przez wyszukanie w scenie powierzchni płaskich i usuwanie nadmiarowych trójkątów wchodzących w skład tych powierzchni. Dzięki temu zmniejsza się szczegółowość mapy, i przez to rozmiar pamięci potrzebnej do przechowywania mapy. Nasze podejście zostało zaimplementowane i eksperymentalnie zweryfikowane dla środowisk wewnątrz budynków.

W artykule [P8] przedstawiliśmy algorytm unikania kolizji z przeszkodami przez robota mobilnego. Proponowane podejście polega na przeszukiwaniu przestrzeni prędkości w celu znalezienia sterowań (prędkości) dopuszczalnych przy uwzględnieniu ograniczeń wynikających z kinematyki robota i uproszczonego modelu dynamiki. Sformułowaliśmy problem omijania przeszkód jako zadanie optymalizacji z ograniczeniami w przestrzeni prędkości robota. Poprawność działania algorytmu zweryfikowaliśmy w badaniach symulacyjnych na grupie robotów poruszającym się w całkowicie nieznanym środowisku.

Współpracę kilku robotów mobilnych rozważaliśmy w zadaniach gry w piłkę nożną [P9] oraz wspólnym pchaniu pudła przez roboty [P33, P45]. W artykułach [P33, P45] przedstawiliśmy formalny opis zadania transportu pudła z podziałem ról dla poszczególnych robotów z wykorzystaniem funkcji przejścia. Opracowaliśmy metodę transportu obiektu przez roboty nieholonomiczne z wykorzystaniem warstwowego układu regulacji – zadaniem każdego z robotów jest utrzymywanie odpowiedniej orientacji względem obiektu, natomiast pożądanym ruchem obiektu jest uzyskiwany przez odpowiedni dobór i koordynację prędkości robotów. Proponowane podejście zweryfikowaliśmy zarówno w badaniach symulacyjnych, jak i na rzeczywistych robotach.

5.2 Czynny udział w konferencjach i seminariach naukowych

Brałem czynny udział w następujących konferencjach międzynarodowych i krajowych:

- IEEE Mediterranean Electrotechnical Conference MELECON, Antalya, Turcja (1994) [P13]
- IFAC Workshop on New Trends in Design of Control Systems, Smolenice, Słowacja (1994) [P26]
- IEEE Symposium on Industrial Electronics, Warszawa (1996) [P39]
- CISM IFToMM Symposium on Theory and Practice of Robots and Manipulators Ro.Man.Sy, Warszawa (2006) [H11]
- International Workshop on Robot Motion and Control (RoMoCo), Bukowy Dworek, Polska (2002) [P4]
- IEEE International Conference on Methods and Models in Automation and Robotics MMAR, Międzyzdroje, Polska (2003,2000,1998,1997,1996,1995) [P5, P15–P17, P22, P23, P37]
- Konferencja Naukowo-Techniczna Automation, Warszawa (2012,2009,1997) [Hu6, Hu5, P31]
- Krajowa Konferencja Robotyki (2014,2010,2008,2006,2004,1996,1993,1990,1988) [P20,P29], [Hu7, Hu4, Hu3], [P2, P9, P12, P24, P28, P30, P32]
- Krajowa Konferencja Algorytmy Ewolucyjne i Optymalizacja Globalna (2006,2005) [P18, P19]

Pozostałe publikacje habilitanta:

- [P1] R. Chojecki, M. Olszewski, W. Szynkiewicz, Trojanek. Laboratoryjny robot mobilny Elektron i jego aplikacje. In: Pomiary Automatyka Robotyka (PAR). Automation 2007 XI Konferencja Naukowo-Techniczna. *Proceedings*, 2007, nr 2, pp. 1–10.
- [P2] A. Gosiewski, W. Szynkiewicz. Zastosowanie algorytmu ruchu ślizgowego do układu sterowania manipulatora robota. 1988. wol. 1, pp. 141–150.
- [P3] A. Grodecki, W. Szynkiewicz. Projektowanie nastaw cyfrowego regulatora położenia ramion robota. In: V Krajowa Konferencja Robotyki, Świeradów Zdrój, Prace Naukowe ICT Politechniki Wrocławskiej. *Proceedings*. Politechnika Wroclawska, 1996, pp. 269–278.
- [P4] W. Kasprzak, W. Szynkiewicz. A method for discrete self-localization using image analysis. In: Robot Motion and Control, RoMoCo'02. Proceedings of the Third International Workshop on. *Proceedings*, Poznań, Poland, IEEE, 9-11 Nov., 2002, pp. 369–374.

- [P5] W. Kasprzak, W. Szynkiewicz. Using color image features in discrete self-localization of a mobile robot. In: Proceedings of the 9th IEEE International Conference on Methods and Models in Automation and Robotics MMAR'2003. *Proceedings*, Międzyzdroje, Poland, Aug. 25–28, 2003, pp. 1101–1106.
- [P6] W. Kasprzak, W. Szynkiewicz, M. Karolczak. Global colour image features for discrete self-localization of an indoor vehicle. In: *Computer Analysis of Images and Patterns* Red. A. Gagalowicz, W. Philips, Vol. 3691 series *Lecture Notes in Computer Science*, pp. 620–627. Berlin, Heidelberg, Springer 2005. IF = 0.302.
- [P7] B. Kubica, W. Szynkiewicz. CuikSLAM with Unknown Correspondence – Preliminary Results. In: *Evolutionary Computation and Global Optimization 2007* Red. J. Arabas, Vol. 1 series *Prace Naukowe Elektronika*, pp. 143–151. Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej 2007.
- [P8] M. Majchrowski, W. Szynkiewicz. Algorytm unikania kolizji przez roboty mobilne bazujące na przeszukiwaniu przestrzeni prędkości. In: *Pomiary Automatyka Robotyka (PAR). Automation 2007 XI Konferencja Naukowo-Techniczna. Proceedings*, 2007, nr 2, pp. 1–11.
- [P9] M. Majchrowski, P. Trojanek, W. Szynkiewicz. Sterowanie robotem mobilnym w zastosowaniu do ligi RoboCup. In: *Postępy Robotyki. Przemysłowe i medyczne systemy robotyczne* Red. K. Tchoń, wol. 2, pp. 233–242. Wydawnictwa Komunikacji i Łączności, Warszawa 2005.
- [P10] K. Mianowski, K. Banachowicz, T. Winiarski, C. Zieliński, W. Szynkiewicz, K. Czajkowski. Chwytnik wielopalczysty dla robota usługowego – konstrukcja. *Pomiary Automatyka Robotyka (PAR)*, 2011, (5):46–52.
- [P11] K. Mianowski, K. Nazarczuk, M. Wojtyra, W. Szynkiewicz, Zieliński C., A. Woźniak. Application of the RNT robot to milling and polishing. In: *Proceedings of the CISM IFToMM Symposium on Theory and Practice of Robots and Manipulators Ro.Man.Sy 13*, pp. 421–429. Zakopane, Poland, Springer 3–6 July, 2000.
- [P12] W. Szynkiewicz. Zastosowanie funkcji sklepanych do aproksymacji ścieżek ruchu dla dwóch współpracujących robotów. In: IV Krajowa konferencja robotyki, Prace Naukowe ICT Politechniki Wrocławskiej. *Proceedings*, Wrocław, 1993. wol. 1, pp. 317–324.
- [P13] W. Szynkiewicz. Admissible Path Planning for Two Cooperated Robot Arms. In: Proceedings of the IEEE Mediterranean Electrotechnical Conference MELECON'94. *Proceedings*, Antalya, Turkey, 1994. Vol. 94, pp. 699–702.
- [P14] W. Szynkiewicz. Planowanie skoordynowanych trajektorii ruchu dla dwóch współpracujących robotów. In: *Biuletyn Przemysłowego Instytutu Automatyki i Pomiarów PIAP: Projekty badawcze granty w dziedzinie robotyki*, pp. 15–18. PIAP 1994.
- [P15] W. Szynkiewicz. Trajectory planning for two robot arms manipulating flexible objects. In: Proceedings of the 4th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). *Proceedings*, Międzyzdroje, Poland, 1997, pp. 1185–1188.

- [P16] W. Szynkiewicz. Game-theoretic approach to multi-robot motion planning and control. In: Proceedings of the 6th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). *Proceedings*, Międzyzdroje, Poland, 2000, pp. 555–560.
- [P17] W. Szynkiewicz. Motion planning for multi-robot systems with closed kinematic chains. In: Proceedings of the 9th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). *Proceedings*, Międzyzdroje, Poland, August 25-28, 2003, pp. 779–786.
- [P18] W. Szynkiewicz. Optimization-based approach to dual-arm manipulation planning. In: *Materiały VIII Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna*, pp. 235–242. Oficyna Wydawnicza Politechniki Warszawskiej 30 May – 1 June, 2005.
- [P19] W. Szynkiewicz. A game-theoretic model of multi-robot interactions. In: *Evolutionary Computation and Global Optimization. Prace Naukowe Elektronika z. 156* Red. J. Arabas, Vol. 1, pp. 365–374. Oficyna Wydawnicza Politechniki Warszawskiej 2006.
- [P20] W. Szynkiewicz. Synteza chwytu przy niepewności pozycji obiektu. In: *Postępy robotyki*, vol. 1, pp. 25–34. Oficyna Wydawnicza Politechniki Warszawskiej 2014.
- [P21] W. Szynkiewicz. Robot grasp synthesis under object pose uncertainty. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 2015, 9(1):53–61.
- [P22] W. Szynkiewicz, Gosiewski A. Motion space analysis and trajectory planning for dual-arm system. In: Proceedings of the Second IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). *Proceedings*, Międzyzdroje, Poland, 1995, pp. 503–510.
- [P23] W. Szynkiewicz, Gosiewski A. Experimental verification of trajectory planning algorithm for two cooperating robot arms. In: Proceedings of the Third IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). *Proceedings*, Międzyzdroje, Poland, 1996, pp. 1011–1016.
- [P24] W. Szynkiewicz, R. Chojecki, A. Rydzewski, M. Majchrowski, P. Trojanek. Modułowy robot mobilny Elektron. In: *Postępy Robotyki: Sterowanie, percepcja i komunikacja* Red. K. Tchoń, vol. 1, pp. 265–274. Wydawnictwa Komunikacji i Łączności, Warszawa 2006.
- [P25] W. Szynkiewicz, K. Czajkowski, C. Zieliński, T. Winiarski, K. Mianowski, K. Banachowicz. Chwytnik wielopalczysty dla robota usługowego – planowanie chwytów. *Pomiary Automatyka Robotyka (PAR)*, 2011, (7–8):75–81.
- [P26] W. Szynkiewicz, A. Gosiewski. Coordinated Trajectories Planning for Two Cooperating Robots. In: 1st IFAC Workshop on New Trend in Design of Control Systems. *Proceedings*, Smolenice, Slovak Republik, 1994, pp. 133–139.
- [P27] W. Szynkiewicz, A. Gosiewski. Coordinated trajectories planning for two cooperating robots. In: *A Postprint Volume from the IFAC Workshop on New Trends in Design of Control Systems* Red. M. Huba, J. Mikles, pp. 123–128. Pergamon Press 1995.

- [P28] W. Szynekiewicz, A. Gosiewski, D. Janecki. Weryfikacja doświadczalna modelu dynamiki manipulatora IRb-6. In: III Krajowa Konferencja Robotyki, Prace Naukowe ICT Politechniki Wrocławskiej. *Proceedings*. Politechnika Wrocławska, 1990, pp. 255–260.
- [P29] W. Szynekiewicz, K. Przedniczek. Upraszczanie map trójwymiarowych z użyciem metody rozszerzania obszaru. In: *Problemy Robotyki* Red. K. Tchoń, C. Zieliński, wol. 1 serii *Prace Naukowe Elektronika*, pp. 111–120. Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej 2008.
- [P30] W. Szynekiewicz, C. Zieliński. Sterownik dwóch współpracujących robotów o 5 stopniach swobody. In: V Krajowa Konferencja Robotyki, Świeradów Zdrój, Prace Naukowe ICT Politechniki Wrocławskiej. *Proceedings*. Politechnika Wrocławska, 1996, pp. 140–147.
- [P31] W. Szynekiewicz, C. Zieliński, K. Kierzenkowski, T. Zielińska. Środowisko programowe do tworzenia sterowników wielorobotowych dla złożonych zastosowań. In: *Automation'97. Proceedings*, marzec, 1997. wol. 1, pp. 127–134.
- [P32] W. Szynekiewicz, C. Zieliński, K. Kierzenkowski, T. Zielińska, A. Grodecki. Rozproszony sterownik wielorobotowy MRROC. In: V Krajowa Konferencja Robotyki, Świeradów Zdrój, Prace Naukowe ICT Politechniki Wrocławskiej. *Proceedings*. Politechnika Wrocławska, 1996, pp. 287–295.
- [P33] P. Trojanek, C. Zieliński, W. Szynekiewicz. Definition and composition of individual robot behaviours in cooperative box pushing. In: *Proceedings of the 13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics. Proceedings*. Technical University of Szczecin, 29–30 August, 2007, pp. 1137–1141.
- [P34] T. Winiarski, K. Banachowicz, C. Zieliński, W. Szynekiewicz, K. Mianowski, K. Czajkowski. Chwytnik wielopalczysty dla robota usługowego – sterowanie. *Pomiary Automatyka Robotyka (PAR)*, 2011, (6):52–57.
- [P35] A. Woźniak, W. Szynekiewicz, C. Zieliński. Robot controller with a self-measurement capability enabling the identification of friction. *Archives of Control Sciences*, 2003, 13(4):391–414.
- [P36] C. Zieliński, K. Mianowski, K. Nazarczuk, W. Szynekiewicz. A Prototype Robot for Polishing and Milling Large Objects. *Industrial Robot*, January, 2003, 30(1):67–76.
- [P37] C. Zieliński, A. Rydzewski, W. Szynekiewicz. Multi-robot system controllers. In: *Proc. of the 5th International Symposium on Methods and Models in Automation and Robotics MMAR'98, Międzyzdroje, Poland. Proceedings*, August 25–29, 1998. Vol. 3, pp. 795–800.
- [P38] C. Zieliński, A. Rydzewski, W. Szynekiewicz, A. Woźniak. *Układ sterowania robotem o strukturze szeregowo-równoległej*. Oficyna Wydawnicza Politechniki Warszawskiej 1997, pp. 55–76.
- [P39] C. Zieliński, W. Szynekiewicz. Control of two 5 d.o.f. robots manipulating a rigid object. June 17-20, 1996. Vol. 2, pp. 979–984.

- [P40] C. Zieliński, W. Szynkiewicz. *Systemy MRROC i MRROC++; Część I: Struktura*. Oficyna Wydawnicza Politechniki Warszawskiej 1997, pp. 25–40.
- [P41] C. Zieliński, W. Szynkiewicz. *Systemy MRROC i MRROC++; Część II: Realizacja*. Oficyna Wydawnicza Politechniki Warszawskiej 1997, pp. 41–54.
- [P42] C. Zieliński, W. Szynkiewicz. Sterownik robota POLYCRANK. *Pomiary – Automatyka – Kontrola PAK*, sierpień, 1999, 45(8):10–16.
- [P43] C. Zieliński, W. Szynkiewicz, K. Mianowski, K. Nazarczuk. Mechatronic design of open-structure multi-robot controllers. *Mechatronics*, November, 2001, 11(8):987–1000.
- [P44] C. Zieliński, W. Szynkiewicz, A. Rydzewski. Metoda konstruowania sterowników systemów wielorobotowych. *Pomiary Automatyka Robotyka (PAR)*, 1998, (12):8–14.
- [P45] C. Zieliński, W. Szynkiewicz, P. Trojanek, M. Majchrowski. Sterowanie zespołu heterogenicznych robotów na przykładzie pchania pudła. In: *Postępy robotyki: Systemy i współdziałanie robotów*. *Proceedings* Red. K. Tchoń. Wydawnictwa Komunikacji i Łączności, 2006. wol. 2, pp. 299–308.
- [P46] C. Zieliński, T. Winiarski, K. Mianowski, A. Rydzewski, W. Szynkiewicz. End-effector sensors role in service robots. In: *Robot Motion and Control 2007 (LNCiS) Lecture Notes in Control & Information Sciences* Red. K. Kozłowski, pp. 401–413. Springer Verlag London Limited June, 11–13, 2007.
- [P47] C. Zieliński, T. Winiarski, W. Szynkiewicz, T. Kornuta, P. Trojanek. MRROC++ – programowa struktura ramowa do tworzenia sterowników systemów wielorobotowych. In: *Inteligencja wokół nas. Współdziałanie agentów softwareowych, robotów, inteligentnych urządzeń* Red. S. Ambroszkiewicz, A. Borkowski, K. Cetnarowicz, C. Zieliński, wol. 15, pp. 317–384. Seria Monografie Komitetu Automatyki i Robotyki PAN, EXIT 2010.
- [P48] C. Zieliński, T. Winiarski, W. Szynkiewicz, K. Mianowski, K. Banachowicz, K. Czajkowski. Sterownik manipulatora z chwytakiem wielopalczastym. In: *KAiR – Postępy Automatyki i Robotyki* Red. K. Malinowski, R. Dindorf, wol. 16, pp. 577–592. Politechnika Świętokrzyska Czerwiec, 2011.

5.3 Kierowanie projektami badawczymi lub udział w takich projektach

5.3.1 Udział w projektach

Projekty międzynarodowe:

1. 7 Program Ramowy UE: *RAPP – Robotic Applications Store for Delivering Smart User Empowering Applications*, FP7-610947, ICT-2013.5.3 ICT for smart and personalised inclusion. Okres realizacji: 01.12.2013-30.11.2016, wykonawca.

2. 7 Program Ramowy UE: *Self reconfigurable Intelligent Swarm Fixtures SwarmItFIX* FP7-214678, NMP-2007-3.2-1 Rapidly configurable machines and production systems. Okres realizacji: 1.10.2008-31.01.2012, wykonawca.
3. 6 Program Ramowy UE: Sieć Doskonałości UE FP6 IST FET 507728 *European Robotics Research Network of Excellence* (EURON II). Okres realizacji: 01.05.2004-31.05.2008, wykonawca.
4. 5 Program Ramowy UE: Sieć Doskonałości EU FP5 IST-200-26048 *European Robotics Research Network of Excellence* (EURON). Okres realizacji: 8.04.2001-30.04.2004, wykonawca.

Projekty krajowe:

1. Grant NCBiR – Program Badań Stosowanych I (ID 179497): *RobREx - Autonomia dla robotów ratowniczo-eksploracyjnych*, PBS1/A3/8/12. Koordynator: Przemysłowy Instytut Automatyki i Pomiarów (PIAP). Okres realizacji: 1.12.2012-30.11.2015, wykonawca.
2. Projekt badawczy MNiSW – N514 1287 33: *Problemy aktywnego czucia, interpretacji informacji sensorycznej i manipulacji w robotach usługowych*. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Okres realizacji: 31.10.2007-30.10.2010, wykonawca.
3. Projekt badawczy MNiSW – 3 T11C 038 29: *Współdziałanie inteligentnych autonomicznych urządzeń*. Koordynator: Akademia Podlaska, Siedlce. Okres realizacji: 5.12.2005-4.12.2008, wykonawca.
4. Projekt badawczy MNiSW – 3 T11A 009 29: *Metodyka postępowania przy wspomaganii człowieka w życiu codziennym oraz w sytuacjach nadzwyczajnych przez roboty mobilne*. Koordynator: Wydział Mechatroniki, Politechnika Warszawska. Okres realizacji: 25.10.2005-24.04.2008, wykonawca.
5. Projekt badawczy MNI – 4 T11A 003 25: *Sterowanie w systemach wielorobotowych wykonujących zadania usługowe*. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych. Okres realizacji: 15.11.2003-14.11.2006, wykonawca.
6. CATID (Centrum Automatyki i Technik Informacyjno-Decyzyjnych) grant: *Projektowanie złożonych systemów robotycznych*. Okres realizacji: 01.10.2001-31.05.2002, wykonawca.
7. CATID grant: *Projektowanie heterogenicznych systemów wielorobotowych*. Okres realizacji: 20.03.2003-31.08.2003, wykonawca.
8. PATIA grant: *Sterowanie w złożonych systemach robotycznych i układach adaptacyjnych*. Okres realizacji: 01.08.1998-31.05.1999, wykonawca.
9. Projekt badawczy KBN – 8 T11A 028 09: *Szybki robot bez ograniczeń konfiguracji z ramieniem odsprężonym dynamicznie*. Politechnika Warszawska. Okres realizacji: 01.09.1995-28.02.1999, wykonawca.

10. Projekt badawczy KBN – 3 P 403 016 05: *Sterowanie inteligentnych robotów przemysłowych wykorzystujące informacje o środowisku przez sprzężenie zwrotne od czujników siły i wizji*. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych. Okres realizacji: 1.08.1993-31.07.1996, wykonawca.
11. Projekt badawczy KBN – 713/S5/93/04: *Eksperymentalna weryfikacja cyfrowych algorytmów sterowania napędami robota przemysłowego*. Koordynator: Przemysłowy Instytut Automatyki i Pomiarów (PIAP), Warszawa. Okres realizacji 01.04.93-31.03.95, wykonawca.
12. Projekt badawczy KBN – 3 0220 91 01: *Nowe i ulepszone algorytmy sterowania robotów przemysłowych*. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych. Okres realizacji: 1.09.1991-31.08.1993, wykonawca.
13. Centralny Program Badań Rozwojowych CPBR 7.1 *Roboty przemysłowe cel 203: Teoria dynamiki, sterowania i programowania robotów i manipulatorów*. MERA-PIAP. Okres realizacji: 1985-1990, wykonawca.
14. Problem węzłowy 06.6: *Roboty i manipulatory przemysłowe – Zbadanie własności dynamicznych oraz projekt regulacji nadążnej dla robotów IRb-6 oraz IRb-60*. Okres realizacji: 1985-1988, wykonawca.

5.3.2 Kierowanie projektami

1. Grant Rektora Politechniki Warszawskiej Nr 503/W/0050 *Drużyna robotów piłkarskich do zawodów RoboCup*. Okres realizacji: 16.04.2009-31.12.2009, kierownik.
2. Grant Rektora Politechniki Warszawskiej Nr 503W/0049/008 *Mobilny robot patrolujący sterowany drogą internetową*. Okres realizacji: 18.04.2008-31.12.2008, kierownik.
3. Grant Rektora Politechniki Warszawskiej Nr 503W/0043/007 *Rozwój projektu czworonożnego robota kroczącego*. Okres realizacji: 16.04.2007-1.12.2007, kierownik
4. Grant Rektora Politechniki Warszawskiej Nr 503R/0040/006 *Czworonożny robot kroczący*. Okres realizacji: 22.05.2006-31.12.2006, kierownik.
5. Grant Rektora Politechniki Warszawskiej Nr 503G/0029/005 *Sterowanie zespołem miniaturowych robotów mobilnych*. Okres realizacji: 17.03.2005-31.12.2005, kierownik.
6. Grant Rektora Politechniki Warszawskiej Nr 503G/0006/003 *Robot mobilny Nomad-IA*. Okres realizacji: 16.06.2003-31.12.2003, kierownik.
7. Grant Dziekana Wydziału Elektroniki Nr 503/011/8: *Sterownik MRROC++ robota IRp-6 posadowionego na torze jezdny*. Okres realizacji: 01.07.1998-31.05.1999, kierownik.
8. Grant WZ179/030/98: *Cooperation with Institute for Robotics and Computer Control of Technical University of Braunschweig*. Okres realizacji: 01.07.1999-31.05.2000, kierownik.

9. Projekt programu PATIA: *Rozproszony sterownik wielorobotowy przeznaczony do celów badawczych (MRROC)*. Okres realizacji: 1994-1995, kierownik.
10. Projekt Centrum Automatyki i Technik Informacyjno-Decyzyjnych (CATID): *Percepcja i sterowanie w systemach robotycznych i układach adaptacyjnych*. Okres realizacji: 1.10.1999-31.05.2000, kierownik.
11. Projekt Centrum Automatyki i Technik Informacyjno-Decyzyjnych (CATID): *Konstrukcja i sterowanie robotów mobilnych i manipulatorów*. Okres realizacji: 01.10.2000-31.05.2001, kierownik.
12. Grant Rektora Politechniki Warszawskiej Nr 503G/0020/200 *Planowanie i koordynacja ruchu dla układów wielorobotowych* 01.09.2000-31.05.2001, kierownik.
13. Grant Rektora Politechniki Warszawskiej Nr 503G/0040/006 *Czworonożny robot kroczący*. Okres realizacji: 01.09.2000-31.05.2001, kierownik.

5.4 Nagrody za osiągnięcia naukowe

Nagrody JM Rektora Politechniki Warszawskiej:

1991 – nagroda zespołowa II stopnia za osiągnięcia naukowe

1995 – nagroda indywidualna II stopnia za osiągnięcia naukowe

1996 – nagroda zespołowa II stopnia za osiągnięcia naukowe

1996 – wyróżnienie rozprawy doktorskiej

1997 – nagroda indywidualna II stopnia za osiągnięcia naukowe

2015 – nagroda zespołowa za osiągnięcia naukowe

5.5 Zbiorcze wskaźniki bibliometryczne

Wskaźniki bibliometryczne na podstawie bazy Web of Science (WoS) Core Collections – Thomson Reuters:

- Sumaryczny Impact Factor (IF) jednotematycznego cyklu 11 publikacji przedstawionego w punkcie 2 – **7.749**.
- Liczba publikacji w indeksowanych w bazie WoS – **17**
- Liczba cytowań publikacji indeksowanych w bazie WoS (**bez autocytowań**) – 69 (**58**)
- Sumaryczna liczba cytowań publikacji habilitanta w bazie WoS (**bez autocytowań**) – 109 (**84**)
- Indeks Hirscha (h-indeks) opublikowanych prac według WoS – **6**

Sumaryczna liczba punktów MNiSW dla jednotematycznego cyklu 11 publikacji – **192** pkt.

Wskaźniki wpływu (Impact Factor) oraz liczba punktów MNiSW są przyjęte zgodnie z rokiem publikacji poszczególnych artykułów.

Literatura

- [1] B. M. Bell, J. V. Burke. Algorithmic differentiation of implicit functions and optimal values. In: *Advances in Automatic Differentiation* Red. Ch. H. Bischof, H. M. Bücker, P. D. Hovland, U. Naumann, J. Utke, pp. 67–77. Springer, Berlin Heidelberg 2008.
- [2] H. Bruyninckx, J. De Schutter. Specification of force-controlled actions in the “task frame formalism”: A synthesis. *IEEE Transactions on Robotics and Automation*, Aug, 1996, 12(4):581–589.
- [3] S. Cambon, R. Alami, F. Gravot. A hybrid approach to intricate motion, manipulation and task planning. *International Journal of Robotics Research*, 2009, 28:104–126.
- [4] J.W. Daniel. *Approximate minimisation of functionals*. Englewood Cliffs, N.J., Prentice Hall 1971.
- [5] L. de Leonardo, M. Zoppi, X. Li, D. Zlatanov, R. Molfino. Swarmitfix: A multi-robot-based reconfigurable fixture. *Industrial Robot*, 2013, 40(4):320–328.
- [6] A. Fiser, R.K. Do, A. Sali. Modeling of loops in protein structure. *Protein Science*, 2000, 9(9):1753–1773.
- [7] J. R. Flanagan, M. C. Bowman, R. S. Johansson. Control strategies in object manipulation tasks. *Current Opinion in Neurobiology*, December, 2006, 16(6):650–659.
- [8] T. Flash, B. Hochner. Motor primitives in vertebrates and invertebrates. *Current Opinion in Neurobiology*, 2005, 15(6):660–666.
- [9] I.M. Gelfand, S.W. Fomin. *Rachunek wariacyjny*. Warszawa, PWN 1979.
- [10] J. J. Gibson. *The Theory of Affordances*. Lawrence Erlbaum 1977, chapter 8, pp. 127–143.
- [11] E. Gilbert, D. Johnson, S. Keerthi. A fast procedure for computing the distance between objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, 1988, 4:193–203.
- [12] R. S. Johansson, J. R. Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 2009, 10(5):345–359.
- [13] M. Kallmann, A. Aubel, T. Abaci, D. Thalmann. Planning collision-free reaching motions for interactive object manipulation and grasping. *Eurographics*, 2003, 22(3):313–322.
- [14] R. Korf. Finding optimal solutions to Rubik’s cube using pattern databases. In: National Conf. on Artificial Intelligence (AAAI-97). *Proceedings*, July, 1997, pp. 700–705.
- [15] J.-C. Latombe. *Robot motion planning*. Boston, MA, Kluwer 1991.
- [16] S.M. LaValle. *Planning Algorithms*. Cambridge, U.K., Cambridge University Press 2006.
- [17] Matlab. *Optimization Toolbox User’s Guide*. The MathWorks, Inc. 2009.

- [18] J.P. Merlet. *Parallel Robots*. Dordrecht, Kluwer 2000.
- [19] R. M. Murray, Z. Li, S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press 1994.
- [20] K. A. Thoroughman, R. Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 2000, 407(6805):742–747.
- [21] A. Wächter, L. T. Biegler. On the implementation of a primal-dual interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 2006, 106(1):25–57.
- [22] W. Wedemeyer, H. Scheraga. Exact analytical loop closure in proteins using polynomial equations. *Journal of Computational Chemistry*, 1999, 20(8):819–844.
- [23] J. Winkler, G. Bartels, L. Mösenlechner, M. Beetz. Knowledge enabled high-level task abstraction and execution. Vol. 2, pp. 131–148. 2012.
- [24] C. Zieliński, T. Winiarski. Motion Generation in the MRROC++ Robot Programming Framework. *The International Journal of Robotics Research*, 2010, 29(4):386–413.
- [25] T. Zielińska. *Maszyny kroczące*. PWN 2003.
- [26] C. Zieliński. Formal approach to the design of robot programming frameworks: the behavioural control case. *Bulletin of the Polish Academy of Sciences – Technical Sciences*, March, 2005, 53(1):57–67.
- [27] C. Zieliński. Transition-function based approach to structuring robot control software. In: *Robot Motion and Control* Red. Krzysztof Kozłowski, Vol. 335 series *Lecture Notes in Control and Information Sciences*, pp. 265–286. Springer-Verlag 2006.
- [28] C. Zieliński. Formalne podejście do programowania robotów – struktura układu sterującego. In: *Inteligencja wokół nas. Współdziałanie agentów softwareowych, robotów, inteligentnych urządzeń* Red. S. Ambroszkiewicz, A. Borkowski, K. Cetnarowicz, C. Zieliński, wol. 15, pp. 267–300. EXIT 2010.



Wojciech Szynkiewicz