

Rozwój robotyki w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej

Wojciech Szynkiewicz, Cezary Zieliński, Włodzimierz Kasprzak

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Instytut Automatyki i Informatyki Stosowanej, ul. Nowowiejska 15/19, 00-665 Warszawa

1. Wstęp

Badania w zakresie robotyki są prowadzone w Instytucie Automatyki i Informatyki Stosowanej (IAiIS) Politechniki Warszawskiej od ponad 25-ciu lat. Pierwsze prace nad mikroprocesorowym układem sterowania robotem PRO-30 rozpoczęto w 1979 roku. W początkowym okresie prace badawcze koncentrowały się na zagadnieniach modelowania kinematyki i dynamiki oraz sterowania manipulatorami robotów przemysłowych. W pierwszej połowie lat 80-tych ubiegłego wieku podjęto prace nad modelem dynamiki robotów przemysłowych RIMP-901 i RIMP-1000. Badania te, w szerszym zakresie i we współpracy z innymi ośrodkami badawczymi w kraju, kontynuowano w centralnych i resortowych programach badań. Zespół uczestniczył w pracach w ramach Problemu Węzłowego 06.6 *Roboty i manipulatory przemysłowe*. W drugiej połowie lat 80-tych prof. A. Gosiewski zostaje koordynatorem celu realizacyjnego 203 *Teoria dynamiki, sterowania i programowania robotów przemysłowych* w CPBR 7.1 *Roboty przemysłowe*. Uczestniczyło w tym projekcie wiele zespołów zarówno z ośrodków akademickich, jak też przemysłowych instytutów naukowo-badawczych. Efektem badań było m.in. opracowanie i identyfikacja modelu dynamiki oraz algorytmów sterowania pozycyjnego dla robotów przemysłowych IRp-6. Opracowano także specjalizowane języki programowania robotów ROOPL, ROPAS, Torbol [Zie91, Zie92]. W kolejnych latach zespół realizował szereg projektów badawczych finansowanych przez Komitet Badań Naukowych. Badanie dotyczyły algorytmów sterowania manipulatorami przemysłowymi z wykorzystaniem informacji z czujników sił i wizyjnych, eksperymentalnej weryfikacji cyfrowych algorytmów sterowania napędami robotów, sterowników programowych robotów, oraz układu sterowania dla szybkiego robota bez ograniczeń konfiguracji. Zespół brał także aktywny udział w pracach Programu Automatyki, Technik Informacyjnych i Automatyzacji (PATIA). Prace te były kontynuowane w Centrum Automatyki i Technik Informacyjno-Decyzyjnych (CATID). W wyniku tych prac powstały sterowniki systemów wielorobotowych (MRROC i MRROC++) [ZR97, ZRS98, ZSMN01].

Zespół od wielu lat współpracuje z innymi instytutami Politechniki Warszawskiej, m.in. Instytutem Techniki Lotniczej i Mechaniki Stosowanej w zakresie konstrukcji i kalibracji manipulatorów, a w szczególności chwytaków wyposażonych w różnorodne czujniki,

Instytutem Automatyki i Robotyki w zakresie budowy robotów mobilnych, Instytutem Sterowania i Elektroniki Przemysłowej w zakresie rozpoznawania obrazów (lokalizacja obiektów), Instytutem Telekomunikacji w zakresie syntezy mowy polskiej.

Ponadto zespół bierze aktywny udział w sieci doskonałości EURON (European Robotics Research Network of Excellence) w ramach 6. Programu Ramowego Unii Europejskiej – kierownik zespołu jest członkiem zarządu sieci i reprezentantem Polski.

2. Istotniejsze wykonane prace badawcze.

2.1. Programowa struktura ramowa MRROC++

Wielość typów oprogramowania sterującego robotami wynika z różnorodności rodzajów robotów, typów czujników będących na ich wyposażeniu, charakterów środowisk, w których działają roboty oraz klas zadań przez nie wykonywanych. Ta różnorodność w istotny sposób utrudnia tworzenie uniwersalnego oprogramowania sterującego prototypami nowych urządzeń, a w konsekwencji wydłuża czas projektowania i jego koszt. Dlatego istotnym jest opracowanie takiej struktury oprogramowania, by w zasadniczy sposób ograniczyć konieczne modyfikacje przy podejmowaniu nowych projektów. Jednym z możliwych rozwiązań problemu powtórnego wykorzystania oprogramowania są struktury ramowe [Zie01]. Większość oprogramowania, a więc również i struktury ramowe, tworzona jest bez zastosowania narzędzi formalnych. Formalizacja opisu oprogramowania ma jednak tę istotną zaletę, że przy projektowaniu wprowadza rygorystyczny sposób postępowania, który w fazie implementacji owocuje przejrzystością struktury, a co za tym idzie prostotą i niezawodnością produktu. W zespole opracowano opis formalny struktur oprogramowania robotów wykorzystujący funkcje przejścia stanu oraz pokazano sposób jego użycia do specyfikacji sterowników systemów wielorobotowych, ze szczególnym uwzględnieniem różnorodnych struktur sterowania pojedynczym robotem [Zie05a, Zie05b].

Jednym z istotniejszych efektów prac zespołu w zakresie programowania robotów było opracowanie programowej struktury ramowej MRROC++ *Multi-Robot Research-Oriented Controller*. Programowa struktura ramowa to [Zie05b]:

- biblioteka modułów, z których będą budowane programy sterujące robotami,
- szablony programów, w które wstawione zostaną wywołania modułów bibliotecznych,
- narzędzia do tworzenia nowych modułów – wzorce modułów i metody ich tworzenia.

Szablon programów stanowi rodzaj uniwersalnej powłoki zawierającej gniazda, do których wstawia się wywołania gotowych modułów bibliotecznych lub nowych modułów skonstruowanych specjalnie na potrzeby powstającego programu. Głównym narzędziem do tworzenia nowych modułów są wzorce modułów oraz metody (przepisy) ich tworzenia. Jest to zatem uniwersalne oprogramowanie do tworzenia sterowników robotów z możliwością powtórnego wykorzystania już istniejących, gotowych elementów oprogramowania dla innych typów robotów. System MRROC++ jest jednocześnie biblioteką modułów służących do konstruowania sterowników programowych dedykowanych zadaniom, które ma wykonać robot lub system wielorobotowy wyposażony w różnorodne czujniki, oraz językiem programowania, w którym to zadanie jest opisywane (programowane). Struktura ramowa MRROC++ charakteryzuje się następującymi cechami:

- Umożliwia tworzenie systemów sterujących dowolną liczbą robotów (systemy wielorobotowe).
- Umożliwia dołączenie dowolnej liczby różnych czujników (kamery, czujniki sił i momentów sił, czujniki dotykowe, czujniki zbliżeniowe: ultradźwiękowe i na podczerwień).
- Tworzone sterowniki programowe są dedykowane specyficznej konfiguracji sprzętowej oraz określonej klasie zadań.
- Bazowym językiem struktury jest uniwersalny język programowania C++.
- Procesy sterowników są aplikacjami systemu operacyjnego czasu rzeczywistego QNX Neutrino.
- Oprogramowanie pracuje w sieci komputerów klasy PC połączonych ze sprzętem wykonanym przez zespół.

W przypadku systemów złożonych, a takim jest system robotyczny, układ sterowania ma często wielowarstwową strukturę hierarchiczną wymiany usług typu *master/slave*. Warstwa nadrzędna – koordynator (*master*) – realizuje cel globalny systemu jako całości, warstwa niższa (*slave*) realizuje sterowanie poszczególnymi efektorami. Mamy więc wyraźny podział funkcji, przez co struktura sterownika jest przejrzysta. To zaś z kolei ułatwia wykrycie i usunięcie potencjalnych błędów i znacznie upraszcza modyfikację oraz rozbudowę sterownika. Cecha ta jest szczególnie istotna przy koncepcji tworzenia sterowników, kiedy każda zmiana zadania powoduje rekonstrukcję sterownika.

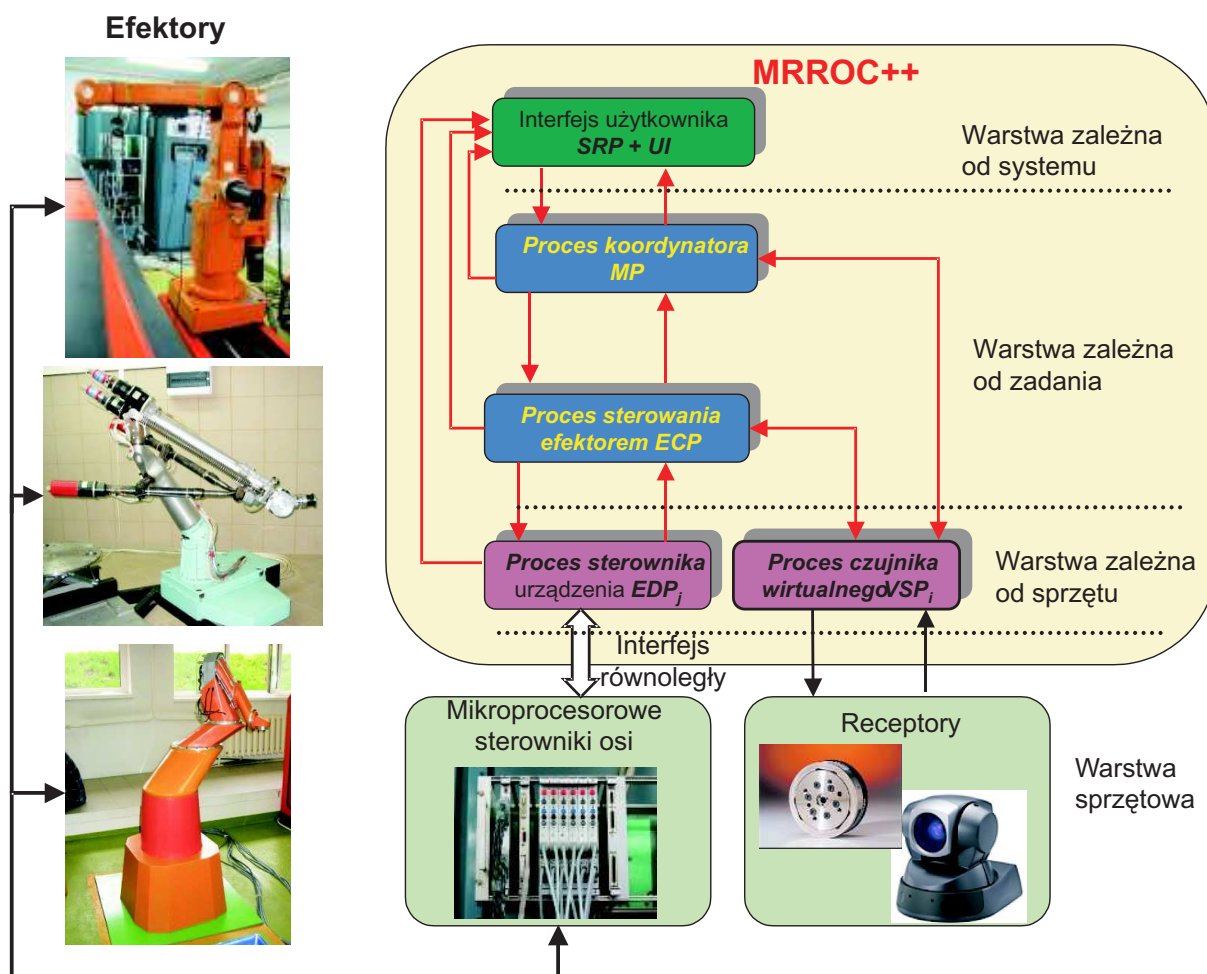
Całość oprogramowania stanowiąca sterownik jest tworzona zgodnie ze schematem wymiany usług (serwer-klient). Konkretnie sterowniki MRROC++ mają hierarchiczną strukturę funkcjonalną i składają się z trzech warstw: zależnej od sprzętu, zależnej od zadania oraz interfejsu operatora (Rys. 1)

MRROC++ wykorzystano z powodzeniem zarówno do tworzenia sterowników prototypowych robotów przemysłowych jak i do badań nad sterowaniem robotami usługowymi.

2.2. Sterowniki robotów stworzone w środowisku MRROC++

System MRROC++ został wykorzystany do konstrukcji układów sterujących dla szeregu robotów prototypowych oraz realizacji różnorodnych zadań. Zastosowano go w prototypowych robotach: RNT o strukturze szeregowo-równoległej [NMAR95, MNW⁺00, ZMNS03], Polycrank z napędami bezpośrednimi bez ograniczeń ruchu w przegubach (Rys. 2) [NM95], oraz robocie przemysłowym IRp-6 (Rys. 3), a także w systemie dwuramiennym składającym się z dwóch robotów IRp-6, w tym jednego posadowionego na torze jezdny [SG95, ZS96].

Tworzone sterowniki programowe były aplikacjami systemu operacyjnego czasu rzeczywistego QNX Neutrino. Ponieważ ten system operacyjny umożliwia rozproszenie procesów w wielu węzłach sieci komputerowej, więc moc obliczeniowa powstałego sterownika może być zwiększana poprzez dołączanie kolejnych komputerów. Umożliwiło to wykonanie zadań wymagających dużych nakładów obliczeniowych i zachowania rygorystycznych wymagań czasowych. W ramach prowadzonych prac zrealizowano zarówno klasyczne zadania przemysłowe (frezowanie w materiałach miękkich (Rys. 6), polerowanie (Rys. 7) [MNW⁺00, ZMNS03]) jak i badawcze [ZSW05].



Rys. 1: Struktura sterownika stworzonego w systemie MRROC++

2.3. Współpraca robotów

Koordinacja działania i sterowanie współpracującymi robotami jest problemem o znacznie większej złożoności w stosunku do planowania ruchu i sterowania pojedynczym robotem [Szy00, Szy03]. Wynika to z konieczności uwzględnienia interakcji, nie tylko między robotem i środowiskiem jak w przypadku pojedynczego robota, lecz także wzajemnych oddziaływań między robotami. Typowym, a jednocześnie jednym z trudniejszych do realizacji, przykładem ścisłej współpracy jest wspólne przeniesienie przez dwa roboty jednego sztywnego obiektu Rys. 8. Problem ten pojawia się wówczas, gdy ciężar obiektu przekracza możliwości udźwigu pojedynczego robota lub rozmiary (kształt) obiektu uniemożliwiają manipulowanie przez pojedynczego robota. W trakcie realizacji tego zadania wymagana jest kinematyczna i dynamiczna koordynacja ruchu robotów, tak aby nie upuściły obiektu i jednocześnie nie uszkodziły go np. przez zginanie lub skręcanie. Po uchwyceniu obiektu przez palce/chwytyki tworzy się mechanizm wieloprzegubowy zawierający zamknięty łańcuch kinematyczny Rys. 9. Więzy wynikające z warunku zamknięcia pętli kinematycznej istotnie ograniczają możliwości ruchowe całego układu [SG95].



Rys. 2: Robot Polycrank



Rys. 3: Robot IRp-6 na torze jezdny



Rys. 4: Frezowanie w drewnie



Rys. 5: Polerowanie powierzchni metalowych

2.4. Rozpoznawanie i śledzenie obiektów w obrazach cyfrowych.

Jednym z obszarów intensywnych badań prowadzonych przez członków zespołu jest rozpoznawanie obiektów w obrazach cyfrowych [Kas00a, Kas00b, Kas03, KCO05, KM05]. Omówione poniżej wyniki prac pokazują możliwości praktycznego zastosowania wyników tych badań.

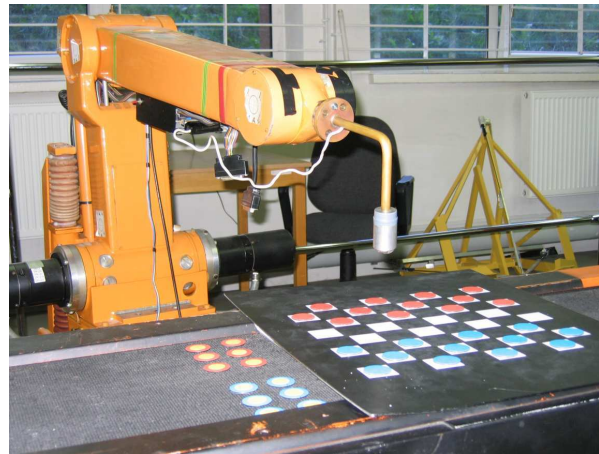
2.4.1. Rozpoznawanie tablic rejestracyjnych i napisów

Stworzono program do analizy obrazów zawierających czołowe widoki samochodów. Celem analizy obrazu jest identyfikacja pojazdu na podstawie automatycznego rozpoznania jego tablicy rejestracyjnej. Możemy wyodrębnić trzy podstawowe etapy analizy [KM05]:

1. Wykrycie prostokątnego obszaru zainteresowania (tablicy) w obrazie (Rys. 10).



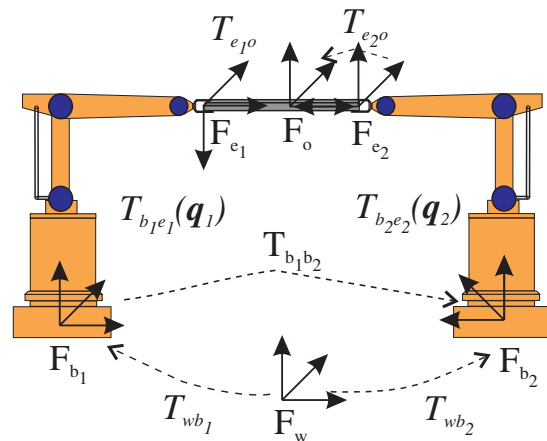
Rys. 6: Sterowanie siłowe – kręcenie korbą



Rys. 7: Gra w warcaby z robotem



Rys. 8: Dwa roboty IRp-6 przenoszące wspólnie sztywną belkę

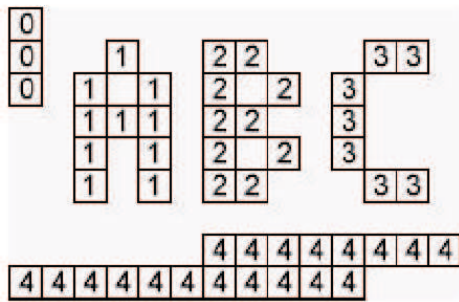


Rys. 9: Zamknięty łańcuch kinematyczny

2. Wyznaczenie ramek obejmujących poszczególne znaki wewnątrz obszaru tablicy (Rys. 11–13).
3. Klasyfikacja znaków (może tu znaleźć zastosowanie komercyjny pakiet OCR lub własna procedura dopasowania pomiaru ze wzorcową bitmapą).



Rys. 10: Idea etapu detekcji obszaru tablicy – analiza rozkładów jasności wzdłuż linii obrazu



Rys. 11: Wyznaczenie i etykietowanie łańcuchów pikseli w obrazie binarnym



Rys. 12: Analiza 2 histogramów - wzdłuż osi X i Y - w celu detekcji pojedynczych znaków



Rys. 13: Przykład znalezionych prostokątów dla pojedynczych znaków

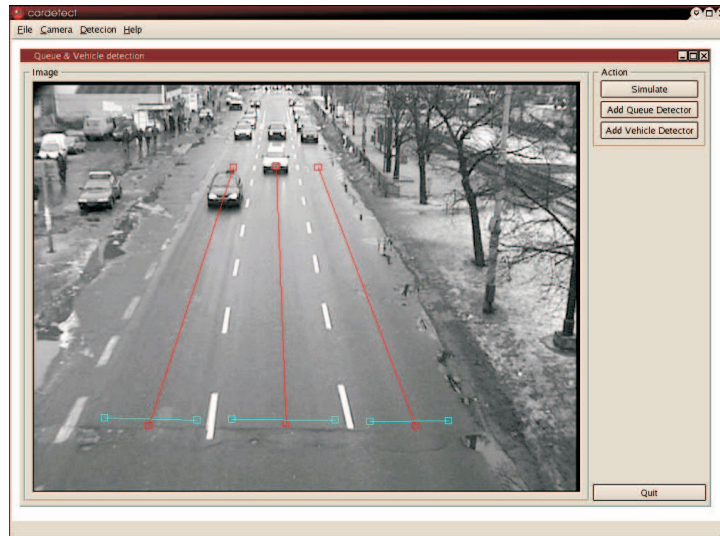
2.4.2. Implementacja czujnika wizyjnego w systemie sterowania ruchem drogowym

W ramach projektu badawczego 5PR Unii Europejskiej p.t. OMNI ("Open Model For Network-wide Heterogeneous Intersection-based Transport Management") zaprojektowano system analizy sekwencji obrazów cyfrowych przeznaczony do nadążnego zliczania pojazdów poruszających się na skrzyżowaniu i do wyznaczania długości kolejki pojazdów oczekujących na sygnalizację świetlną [Kas00b].

2.5. Adaptacyjne metody separacji i rozplatanie sygnałów z ich mieszanin.

W robotyce usługowej jednym z istotniejszych problemów wymagających rozwiązania jest komunikacja człowieka z robotem. Potencjalnie możliwych jest kilka sposobów porozumiewania się człowieka z maszyną m.in. z wykorzystaniem mowy, rozpoznawania gestów, itp. Dla zwykłego użytkownika naturalnym sposobem komunikowania z robotem jest wydawanie poleceń głosem. Problem polega na rozpoznaniu zarówno mówcy jak i wydawanych przez niego poleceń. Zakładamy działanie robota w częściowo znanym środowisku wewnętrznym, w którym rozmieszczono wiele mikrofonów - na platformach mobilnych i w otoczeniu robotów. Zagadnienie rekonstrukcji właściwego, niezakłóconego sygnału mowy z pewnej ilości bodźców (zakłóconych sygnałów) modelowane jest jako problem „ślepej separacji” (*BSS – Blind Source Separation*) względnie „ślepego rozplatania źródeł” (*BSD – Blind Source Deconvolution*) [Kas03, KCO05, KOS05].

Rozwiązaniem problemu jest podanie algorytmu uczenia bez nauczyciela macierzowej funkcji (np. wag sieci neuronowej) przetwarzającej sygnały wektorowe. Wykorzystuje

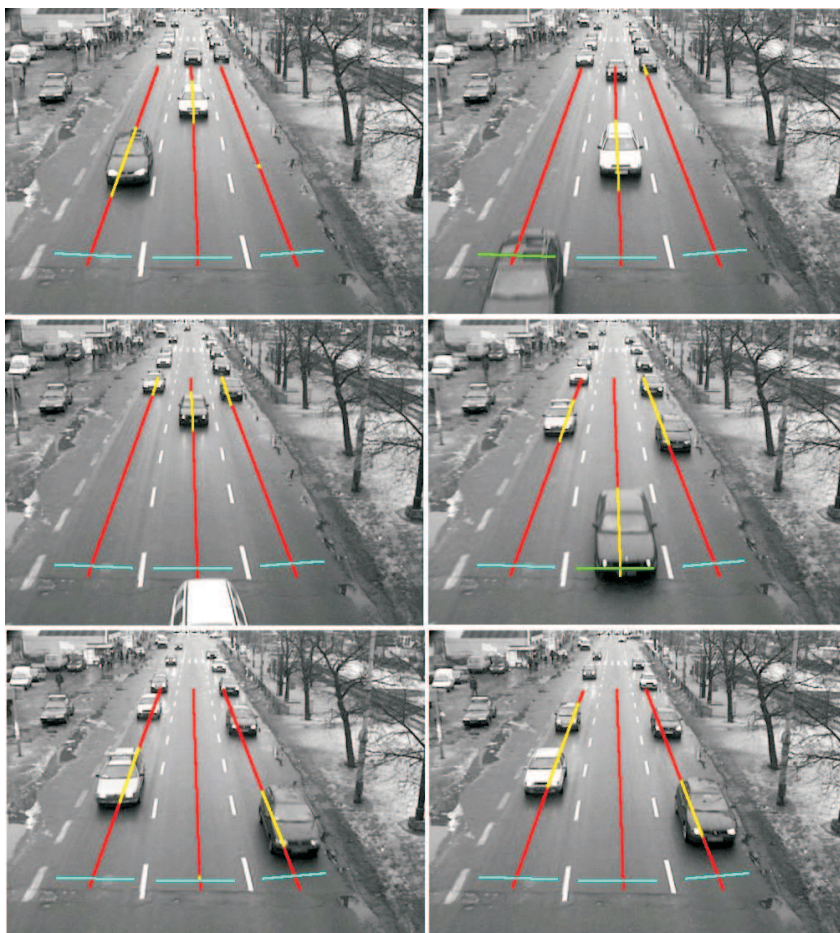


Rys. 14: Linie poziome wyznaczają bramki, w których zliczana jest liczba pojazdów. Linie pionowe reprezentują pasy ruchu, wzdłuż których mierzona jest długość kolejki

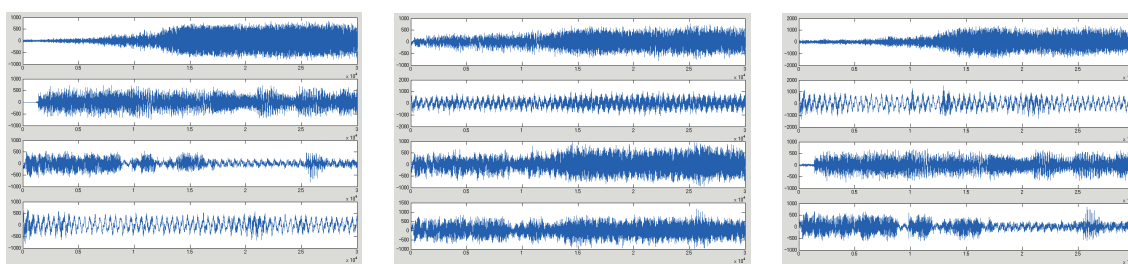
się przekształcenie ICA (*Independent Component Analysis* – analiza niezależnych składowych) w zastosowaniu do wektorowych sygnałów (będących różnymi mieszaninami wielu nieznanych sygnałów źródłowych) [KO04].

3. Aktualna tematyka badań

Obecny stan rozwoju robotyki umożliwia realizację przez roboty zarówno zadań przemysłowych jak i usługowych. Należy zwrócić uwagę na fakt, że w przypadku zadań przemysłowych można było rozdzielić sferę działalności robota i człowieka. W przypadku zadań usługowych robot musi wchodzić w kontakt z człowiekiem, a więc sfery te nie mogą być rozgraniczone. Roboty usługowe działają w niestrukturalnym środowisku dynamicznym, gdzie oprócz robotów mogą być inne ruchome obiekty np. ludzie, których trajektorie ruchu są zazwyczaj nieznane i nieprzewidywalne. W wielu zadaniach usługowych konieczny jest bezpośredni kontakt robotów z człowiekiem. Pojawiają się zatem nowe wymagania na bezpieczeństwo funkcjonowania tego typu urządzeń w naturalnym otoczeniu człowieka. Aktualne prace badawcze zespołu koncentrują się na tej drugiej klasie zadań i zasadniczym celem jest opracowanie i zbadanie metod sterowania wielorobotowymi systemami wykonującymi zadania usługowe. Praktycznym celem jest skonstruowanie dwuramiennego systemu robotycznego wyposażonego we wzrok, słuch i czucie siły/dotyku, a przeznaczonego do wykonywania zadań wymagających zarówno zręczności manualnej jak i inteligencji. Konstrukcja takiego systemu wymaga rozwiązania szeregu problemów badawczych, m.in opracowania ogólnej architektury układów sterujących robotami usługowymi [Zie05a], opracowania algorytmów sterowania dla serwomechanizmu wizyjnego, algorytmów sterowania siłowego i pozycyjno-siłowego [WZ05b, WZ05a], opracowania metod fuzji danych pochodzących z różnych czujników, rozwiązania zadania planowania działań i transformacji tych ostatnich w ruchy manipulatorów [Szy05], konstrukcji chwytaków do zręcznej manipulacji, wykorzystania zarówno klasycznych metod sztucznej inteligencji (przeszukiwanie przestrzeni rozwiązań problemu) jak i metod behawioralnych



Rys. 15: Przykłady zliczania pojazdów (detekcja pojazdu sygnalizowana jest zmianą koloru linii z niebieskiego na zielony) i detekcji kolejki (zajętość jezdni przez pojazd sygnalizowana jest zmianą czerwonej linii na zieloną)

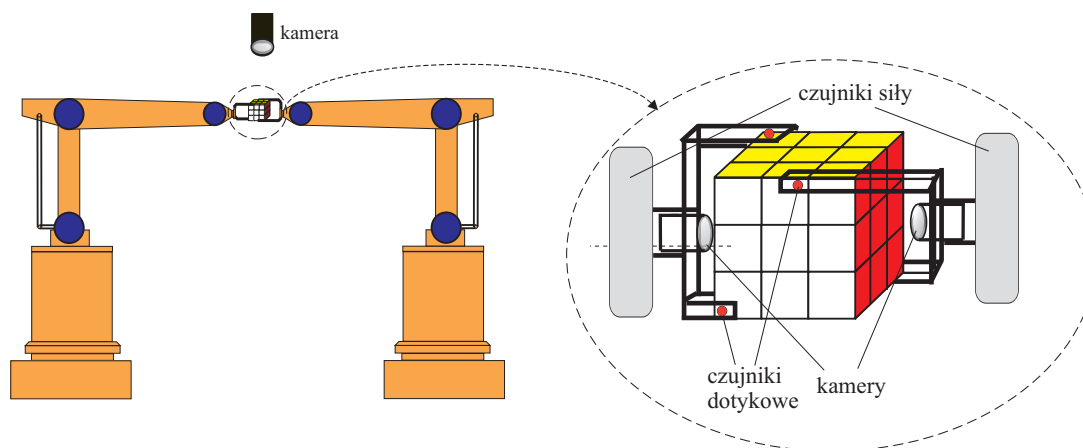


Rys. 16: Nieznane 4 sygnały dźwiękowe (źródła) Rys. 17: Znane są 4 sygnały – sploty źródeł Rys. 18: Wynik rozplatania źródeł z ich splotów

w sterowaniu systemem [Zie03,Zie04], detekcji sygnału mowy oraz rozpoznawania komend wydawanych głosem w języku polskim [OK05].

3.1. Dwuramienny (dwuręczny) robot usługowy

Człowiek wykonując wiele codziennych czynności wymagających manipulowania obiektami, posługuje się dwiema rękami. Odpowiednikiem tego jest manipulacja wykonywana przez robota dwuramiennego, którego ramiona zakończone są wielopalczastymi chwytakami. Może to być dwuręczny robot humanoidalny lub dwa manipulatory stacjonarne/mobilne współpracujące ze sobą. Robot ma wykonywać typowe zadania manipulacji w naturalnym środowisku człowieka. Stworzenie systemu sterowania dla takiego robota wymaga rozwiązania szeregu problemów, takich jak, manipulacja dwuręczna [Szy05], sterowanie pozycyjno-siłowe [WZ05a], serwomechanizmy wizyjne [CS04], rozpoznawanie słów mówionych – komend głosowych [KOS05, OK05]. Prowadzone przez zespół badania mają na celu opracowanie efektywnych algorytmów rozwiązywania poszczególnych zadań i ich integrację w jeden spójny system sterowania robota usługowego. Testowym zadaniem mającym pokazać możliwości dwuramiennego robota jest układanie kostki Rubika (Rys. 19).



Rys. 19: Manipulacja dwuręczna z wykorzystaniem czujników

Układanie kostki Rubika jest zadaniem pokazującym zdolności do wnioskowania w połączeniu z umiejętnościami manualnymi. Zadanie to wymaga następujących wykonania czynności:

- Detekcji sygnału mowy i rozpoznania komendy głosowej zlecającej wykonanie zadania.
- Wykrycia i rozpoznania, w polu widzenia kamery, kostki trzymanej przez człowieka.
- Odebrania kostki z ręki człowieka – śledzenie na bieżąco zmiany jej położenia (kostka nie musi być trzymana nieruchomo).
- Identyfikacji stanu początkowego kostki – wymaga obracania kostki przed kamerą i przekładania kostki z ręki do ręki (chwytaka do chwytaka).
- Zastosowania metod sztucznej inteligencji w celu wyznaczenia planu ruchów ścianami kostki.
- Transformacji planu układania na lokalizacje chwytów oraz nominalne trajektorie ruchów dwuręcznego manipulatora.

- Realizacji ruchów z wykorzystaniem informacji z kamer wizyjnych, czujników sił i momentów, czujników dotykowych.
- Oddania ułożonej kostki człowiekowi.

3.2. Rozpoznawanie słów mówionych – komend głosowych

Problematyka analizy sygnałów mowy badana jest od wielu lat i znane są znaczne osiągnięcia praktyczne w tym zakresie. Nasze podejście zakłada możliwość sterowania maszyną za pomocą systemu rozpoznawania komend głosowych podawanych w języku polskim [KO04, OK05]. Wyróżniamy następujące zasadnicze etapy analizy: analiza wstępna – próbkowanie, analiza spektralna, detekcja cech dla okien sygnału – tzw. liftrowane cechy mel-cepstralne, wyznaczenie klas fonetycznych (trzyfony), wyznaczenie modelu fonetycznego (transkrypcji fonetycznej w terminach fonemów a następnie trzyfonów), rozpoznawanie komend – optymalne dopasowanie modeli komend z sekwencją cech okien dla aktualnego sygnału, identyfikacja cech osobniczych mówcy – detekcja częstotliwości bazowej mówcy, detekcja położenia formantów dla wybranych samogłosek. Większość powyższych zagadnień ma mniej lub bardziej standardowy charakter, cechą wyróżniającą nasze rozwiązanie jest sposób fonetycznego modelowania mowy polskiej [KO04].

3.3. Rozpoznawanie dłoni w obrazie cyfrowym - komendy dawane dłonią

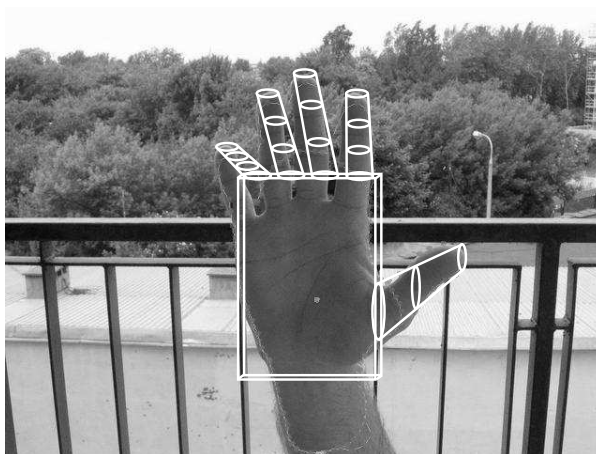
Analiza obrazu zawierającego dłoń ma zastosowanie w zakresie komunikacji człowieka z maszyną i identyfikacji osób (w biometrii). Celem jest rozpoznawanie układu dłoni (ułożenia dłoni frontem lub bokiem) i liczby widocznych palców, celem interpretacji takiego układu jako polecenia danego maszynie przez operatora. Wyróżniamy cztery etapy analizy obrazu dla rozpoznawania dłoni:

1. analiza kolorowa w celu detekcji obszaru skóry człowieka,
2. detekcja dwóch „aktywnych konturów” – jednego obejmującego całą dłoń z palcami, i drugiego obejmującego jedynie podstawę dłoni,
3. na podstawie modelu 3-D dłoni i cech obu konturów – trójwymiarowa interpretacja ułożenia dłoni,
4. śledzenie konturu w sekwencji obrazów i interpretacja ruchu dłoni w terminach gestu.

Detekcja pikseli obrazu odpowiadających swoim kolorem dłoni jest wykonywana w przestrzeni kolorów YCbCr. Aktywny kontur zdefiniowany jest w postaci listy punktów kontrolnych w obrazie (Rys. 20). Podczas iteracyjnej minimalizacji energii związanej z aktywnym konturem, kontur może przemieszczać się w obrazie i zmieniać swoją długość oraz kształt. Zewnętrzna energia konturu zależy od aktualnego obrazu. Siła zewnętrzna odpowiada ujemnemu gradientowi energii potencjalnej. Jej zadaniem jest przemieszczanie konturu w kierunku krawędzi w obrazie. Energia wewnętrzna zależy od samego konturu i jest liczona jako suma dwóch składowych: energii elastyczności i energii sprężystości. Generujemy 3-wymiarową interpretację ułożenia dłoni. W celu wizualizacji dysponujemy stosunkowo prostym modelem dłoni złożonym z 1 prostopadłościanu i 14 ściętych stożków (Rys. 21). Analiza rozmiarów obu konturów i znalezionych końców palców przy wykorzystaniu modelu dłoni pozwala na 3-wymiarową interpretację gestu-komendy.



Rys. 20: Wynik detekcji dwóch aktywnych konturów



Rys. 21: Wizualizacja wynikowej interpretacji ułożenia dłoni w obrazie

3.4. Autonomiczna nawigacja robotów mobilnych

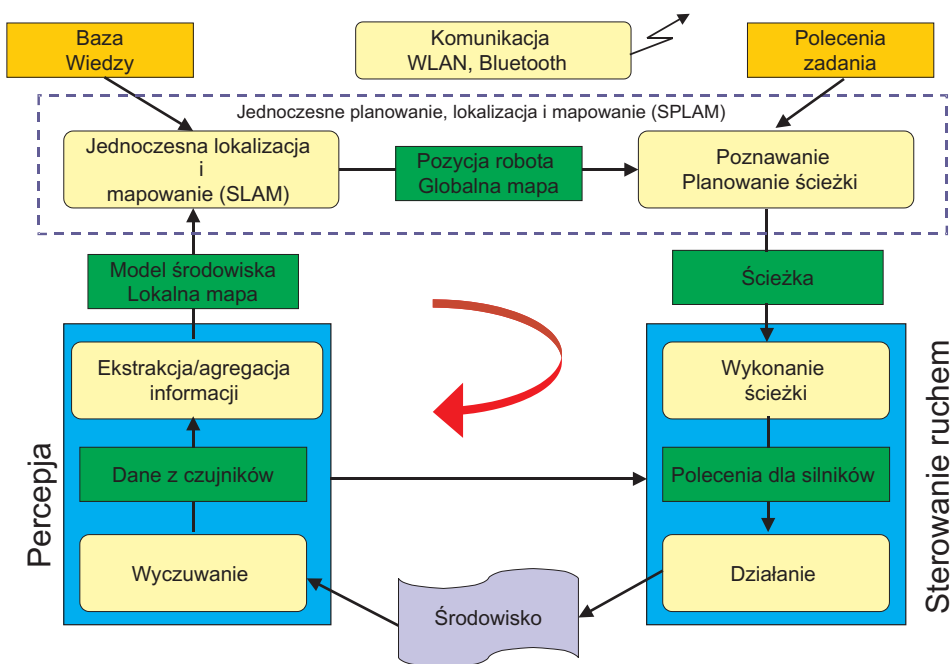
Nowym obszarem zainteresowań zespołu jest problematyka autonomicznej nawigacji robotów mobilnych (Rys. 22). Przy sterowaniu robotami mobilnymi kluczowym zagad-



Rys. 22: Roboty mobilne

nieniem staje się wykorzystanie pochodzącej z różnorodnych czujników informacji o środowisku, która zwykle jest cząstkowa, obarczona niepewnością, a czasami błędna. Zakres aktualnie prowadzonych badań obejmuje następujące zagadnienia: lokomocję, percepcję, wykrywanie i unikanie kolizji, samo-lokalizację robotów, planowanie i generację trajektorii ruchu, budowę modelu otoczenia robota (mapy) oraz współpracę zespołu robotów. Docelowa struktura układu sterowania dla robota autonomicznego jest przedstawiona na Rys. 23.

Autonomiczne roboty mobilne zazwyczaj działają w niestrukturalnym środowisku dynamicznym, gdzie oprócz robotów mogą być inne ruchome obiekty np. ludzie, których trajektorie ruchu są zazwyczaj nieznane i nieprzewidywalne. W wielu zadaniach usługowych konieczny jest bezpośredni kontakt robotów z człowiekiem. W takiej sytuacji planowanie



Rys. 23: Struktura systemu nawigacji autonomicznej robotów mobilnych

działań/ruchu robota czy też zespołu robotów jest bardzo złożonym problemem, którego rozwiązanie wymaga stosowania zaawansowanych metod i algorytmów.

Jeśli wiele robotów porusza się w tej samej przestrzeni roboczej ich ruchy muszą być koordynowane (harmonizowane, jeśli nie ma nadrzędnego koordynatora) w celu uniknięcia wzajemnych kolizji i zakleszczeń. Złożoność problemu sterowania zależy od bardzo wielu czynników m.in. liczby i typu robotów, stopnia ich współpracy, rodzaju środowiska (statyczne/dynamiczne, znane/nieznane) itd.

3.5. Analiza obrazu na potrzeby samo-lokalizacji

Jednym z podstawowych problemów jest określenie pozycji robota mobilnego. W proponowanym podejściu do tego celu wykorzystywane są pewne cechy obrazu cyfrowego i zastosowanie odpowiedniego filtru do estymacji pozycji. Wykorzystywana jest bayesowska filtracja rozkładu prawdopodobieństwa stanów przy spełnieniu założenia Markowa o skończonym horyzoncie czasowym zależności prawdopodobieństwa stanów [KS03, KSK05]. Jest to podejście zbliżone do filtracji cząsteczkowej.

Detekcja wektora pomiarowego dla cyfrowego obrazu obejmuje [KSK05]:

- W każdym stanie robota (3 stopnie swobody: lokalizacja (X, Y) + kierunek) zastosowanym do uczenia się otoczenia dokonujemy akwizycji obrazu.
- Obraz dzielony jest na 9 podobrazów i dla każdego z nich określamy wektor (np. zestaw *Hist6* – składowe HSV lub YCbCr dla 3 dominujących kolorów wraz z ich względną częstością w obrazie) cech globalnych.

4. Podsumowanie

Przedstawione w niniejszym artykule prace badawcze z zakresu robotyki i rozpoznawania i przetwarzania obrazów i mowy stanowią tylko wybrane zagadnienia, którymi zajmował się i aktualnie pracuje Zespół Programowania Robotów i Rozpoznawania Wzorców IAIIS. W całym okresie działalności zespołu ambicją jego członków było dążenie do praktycznego weryfikacji tworzonych koncepcji teoretycznych dotyczących programowania i sterowania robotów oraz przetwarzania obrazów i mowy. Dlatego też większość prac miała aspekt praktyczny w postaci działającego urządzenia i jego oprogramowania. Obecnie dalekosiężnym celem prowadzonych prac naukowych jest stworzenie robota usługowego, który miałby zdolność do autonomicznego działania w naturalnym środowisku człowieka oraz wspomagania lub nawet zastąpienia człowieka w jego codziennych czynnościach.

Literatura

- [CS04] W. Czajewski, M. Staniak. Visual identification of the rubik's cube in human environment. *Przegląd Elektrotechniczny*, (4):311–314, 2004.
- [Kas00a] W. Kasprzak. *Adaptive computation methods in digital image sequence analysis. Prace Naukowe - Elektronika*, Nr 127. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2000.
- [Kas00b] W. Kasprzak. Adaptive methods of moving car detection in monocular image sequences. *Machine Graphics & Vision, International Journal*, 9(1-2):167–186, 2000.
- [Kas03] W. Kasprzak. Blind deconvolution of timely correlated sources by gradient descent search. *Image Processing and Communications – An International Journal*, 9(1):33–52, 2003.
- [KCO05] W. Kasprzak, A. Cichocki, A. Okazaki. Blind extraction of sparse images from under-determined mixtures. *Computer Vision and Graphics. Proceedings Int. Conf. on Computer Vision and Graphics*, strony 826–831. 2005.
- [KM05] W. Kasprzak, M. Jankowski M. Implementacja czujnika wizyjnego w systemie sterowania ruchem drogowym. *Postępy robotyki. Sterowanie robotów z percepcją otoczenia*, strony 73–82. WKŁ, 2005.
- [KO04] W. Kasprzak, A. Okazaki. Applying independent component analysis for speech feature detection. *Proc. of 11th Int. Workshop on Systems, Signals and Image Processing IWSSIP04, Poznań, Poland*, strony 323–326. 13-15 September 2004.
- [KOS05] W. Kasprzak, A. Okazaki, R. Seta. A common feature representation scheme for speech and contour recognition. *Computer Vision and Graphics. Proceedings Int. Conf. on Computer Vision and Graphics*, strony 463–468. Kluwer Publisher, 2005.
- [KS03] W. Kasprzak, W. Szyrkiewicz. Using color image features in discrete self-localization of a mobile robot. *Proceedings of the 9th IEEE International Con-*

- ference on Methods and Models in Automation and Robotics MMAR'2003, Międzyzdroje, strony 1101–1106. 2003.
- [KSK05] W. Kasprzak, W. Szynkiewicz, M. Karolczak. Global colour image features for discrete self-localization of an indoor vehicle. *11th Int. Conf. on Computer Analysis of Images and Patterns CAIP2005. Lecture Notes in Computer Science 3691*, strony 620 – 627. Springer Verlag, 2005.
- [MNW⁺00] K. Mianowski, K. Nazarczuk, M. Wojtyra, W. Szynkiewicz, Zieliński, A. Woźniak. Application of the rnt robot to milling and polishing. *Proceedings of the CISM IFToMM Symposium on Theory and Practice of Robots and Manipulators Ro.Man.Sy 13, Zakopane, Poland, 3–6 July*, strony 421–429. 2000.
- [NM95] K. Nazarczuk, K. Mianowski. Polycrank – fast robot without joint limits. *Proceedings of the 12-th CISM-IFTToMM Symposium on Theory and Practice of Robots and Manipulators Ro.Man.Sy'12, Vienna, 6–9 June*, strony 317–324. Springer-Verlag, 1995.
- [NMAR95] K. Nazarczuk, K. Mianowski, A. Ołędzki A., C. Rzymkowski. Experimental investigation of the robot arm with serial-parallel structure. *Proceedings of the 9-th World Congress on the Theory of Machines and Mechanisms, Milan, Italy*, strony 2112–2116. 1995.
- [OK05] F. A. Okazaki, W. Kasprzak. A two-step approach to blind deconvolution of speech and sound sources in the time domain. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 53(1):49–55, 2005.
- [SG95] W. Szynkiewicz, A. Gosiewski. Motion space analysis and trajectory planning for dual-arm system. *Proceedings of the 2nd International Symposium on Methods and Models in Automation and Robotics, Międzyzdroje*, strony 503–510. 1995.
- [Szy00] W. Szynkiewicz. Game-theoretic approach to multi-robot motion planning and control. *Proceedings of the 6th International Conference on Methods and Models in Automation and Robotics*, strony 555–560. 2000.
- [Szy03] W. Szynkiewicz. Motion planning for multi-robot systems with closed kinematic chains. *Proceedings of the 9th IEEE International Conference on Methods and Models in Automation and Robotics MMAR'2003, Międzyzdroje*, strony 779–786. August 25–28 2003.
- [Szy05] W. Szynkiewicz. Planowanie manipulacji dwuręcznej. *Postępy robotyki. Sterowanie robotów z percepcją otoczenia*, strony 187–196. WKŁ, 2005.
- [WZ05a] T. Winiarski, C. Zieliński. Implementation of position–force control in MRROC++. *Proceedings of the 5th International Workshop on Robot Motion and Control, RoMoCo'05, Dymaczewo, Poland*, strony 259–264. June, 23–25 2005.
- [WZ05b] T. Winiarski, C. Zieliński. Stanowisko do badania algorytmów sterowania pozycyjno–siłowego robotów. *Postępy robotyki. Sterowanie robotów z percepcją otoczenia*, strony 85–94. WKŁ, 2005.
- [Zie91] C. Zieliński. TORBOL: An object level robot programming language. *Mechatronics*, 1(4):469–485, 1991.

- [Zie92] C. Zieliński. Object level robot programming languages. A. Morecki et.al., redaktor, *Robotics Research and Applications*, strony 221–235. Warsaw, 1992.
- [Zie01] C. Zieliński. By how much should a general purpose programming language be extended to become a multi-robot system programming language? *Advanced Robotics*, 15(1):71–95, 2001.
- [Zie03] C. Zieliński. A unified formal description of behavioural and deliberative robotic multi-agent systems. *Proc. 7th IFAC International Symposium on Robot Control SYROCO 2003, Wrocław, Poland*, wolumen 2, strony 479–486. September 1–3 2003.
- [Zie04] C. Zieliński. Specification of behavioural embodied agents. K. Kozłowski, redaktor, *Fourth International Workshop on Robot Motion and Control, RoMoCo'04, Puszczykowo, Poland*, strony 79–84. June 17–20 2004.
- [Zie05a] C. Zieliński. Formal approach to the design of robot programming frameworks: the behavioural control case. *Bulletin of the Polish Academy of Sciences – Technical Sciences*, 53(1):57–67, March 2005.
- [Zie05b] C. Zieliński. Formalizacja opisu struktur ramowych do programowania systemów wielorobotowych. *Postępy robotyki. Przemysłowe i medyczne systemy robotyczne*, strony 53–66. WKŁ, 2005.
- [ZMNS03] C. Zieliński, K. Mianowski, K. Nazarczuk, W. Szynkiewicz. A prototype robot for polishing and milling large objects. *Industrial Robot*, 30(1):67–76, January 2003.
- [ZR97] C. Zieliński, T. Zielińska (Red.). *Konstrukcja, sterowanie i programowanie złożonych systemów robotycznych*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1997.
- [ZRS98] C. Zieliński, A. Rydzewski, W. Szynkiewicz. Multi-robot system controllers. *Proc. of the 5th International Symposium on Methods and Models in Automation and Robotics MMAR'98, Międzyzdroje, Poland*, wolumen 3, strony 795–800. August 25–29 1998.
- [ZS96] C. Zieliński, W. Szynkiewicz. Control of two 5 d.o.f. robots manipulating a rigid object. *IEEE Int. Symp. on Industrial Electronics ISIE'96, Warsaw, Poland*, wolumen 2, strony 979–984. June 17–20 1996.
- [ZSMN01] C. Zieliński, W. Szynkiewicz, K. Mianowski, K. Nazarczuk. Mechatronic design of open-structure multi-robot controllers. *Mechatronics*, 11(8):987–1000, November 2001.
- [ZSW05] C. Zieliński, W. Szynkiewicz, T. Winiarski. Applications of MRROC++ robot programming framework. *Proceedings of the 5th International Workshop on Robot Motion and Control, RoMoCo'05, Dymaczewo, Poland*, strony 251–257. June, 23–25 2005.