

# Wielorobotowa rekonfigurowalna forma mocująca obrabiane detale – układ sterowania

Cezary Zieliński\*, Piotr Trojanek\*, Tomasz Kornuta\*, Tomasz Winiarski\*, Michał Wałęcki\*,  
Włodzimierz Kasprzak\*, Wojciech Szykiewicz\*, Teresa Zielińska\*\*

\*Instytut Automatyki i Informatyki Stosowanej, Politechnika Warszawska,

\*\*Instytut Techniki Lotniczej i Mechaniki Stosowanej, Politechnika Warszawska

**Streszczenie:** Formy mocujące muszą być idealnie dopasowane do detali, które mają podierać. Nawet mała modyfikacja kształtu w projekcie detalu powoduje, że kosztowna forma staje się bezużyteczna. Stąd duże zainteresowanie przemysłu formami rekonfigurowalnymi. Zastąpienie tradycyjnych form przez wiele robotów stanowiących ruchome podpory wymaga zaprojektowania specjalnego układu sterowania oraz dedykowanej metody programowania umożliwiającej szybką rekonfigurację takiego systemu. Układ sterowania systemu wielorobotowego został zaprojektowany na podstawie formalnej specyfikacji definiującej strukturę tego układu za pomocą agentów, których zachowanie określane jest za pomocą funkcji przejścia. W ten sposób stworzono system modułowy, umożliwiający parametryzację oprogramowania ułatwiającą wprowadzanie zmian przy sprawdzaniu różnych rozwiązań technicznych, co jest nieodzowne przy konstrukcji systemu prototypowego. Eksperymenty przeprowadzone w fabryce wykazały, że zaprojektowany system usztywnia detal na tyle, aby wynik obróbki mechanicznej był zadowalający. Jeżeli liczba różnych detali podlegających obróbce jest znaczna, to zaprojektowany system stanowi względnie tanią alternatywę dla wytworzenia i późniejszego magazynowania wielu form. Ponadto zaprojektowany układ sterowania może sterować zespołami robotów o różnej liczności oraz dopuszcza zmiany konstrukcyjne poszczególnych części robotów. Pierwsza część artykułu przedstawia problemy związane z konstrukcją form podpierających oraz prezentuje strukturę układu sterowania systemu wielorobotowego, natomiast część druga koncentruje się na programie planującym działania robotów.

**Słowa kluczowe:** rekonfigurowalne formy mocujące, systemy wielorobotowe, systemy wieloagentowe, układy sterowania

Mimo tego, że obecne badania w dziedzinie robotyki koncentrują się na robotach usługowych i terenowych, w przemyśle nadal istnieją nierozwiązane istotne problemy wymagające zastosowania robotów. Przykładowo, w przemyśle lotniczym stosowane są kosztowne formy mocujące detale (rodzaj kopyt kowalskich), na których wykonywane są operacje obróbcze. Z przyczyn technologicznych cienkie płyty blachy o znacznych rozmiarach i przestrzennych kształtach przed obróbką mechaniczną (np. wierceniem lub frezowaniem) muszą zostać przytwierdzone do usztywniającej je formy. Obecnie do tego celu najczęściej stosuje się osobne formy dla każdego z obrabianych detali (rys. 1). 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 wyko-

nywane jest na stanowisku obróbczym, czyni je przez ten czas nieproduktywnym.

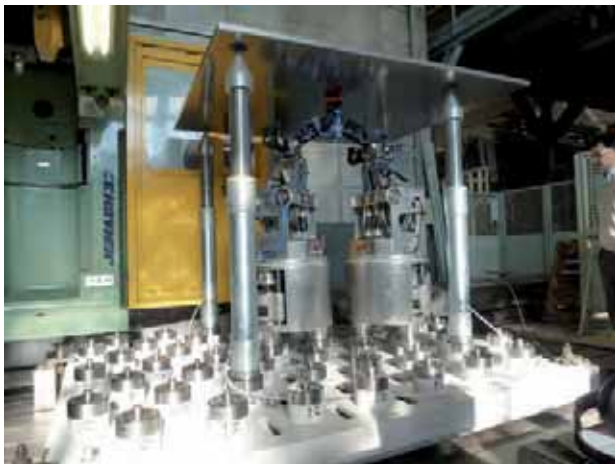


**Rys. 1.** Typowa forma mocująca stosowana w zakładach Piaggio Aero

**Fig. 1.** A standard fixture used by Piaggio Aero

Dotychczas wytwarzane były dwie postacie rekonfigurowalnych systemów mocujących dla cienkościennych detali: modułowe wymienne uchwyty (ang. MFFS – Modular Flexible Fixture Systems) oraz uchwyty o zmiennym kształcie i jednolitej strukturze (ang. SSFFS – Single Structure Flexible Fixture Systems). Te pierwsze składają się z zestawu modułów mocujących o różnych kształtach, mocowanych do podstawy, natomiast te drugie zazwyczaj składają się z gęstej sieci podpór o zmiennej wysokości lub tworzone są na bazie materiałów, które zmieniają swoją sztywność. Niestety obie wersje mają ograniczoną zdolność do rekonfiguracji [6, 7]. Głównymi wadami są wysokie koszty, ograniczona zdolność do zmiany kształtu oraz znaczny czas potrzebny do zmiany ustawienia takich uchwytów, co czyni je trudnymi lub nieefektywnymi w wykorzystaniu do detali o różnych rozmiarach i skomplikowanych kształtach. Istnieją jeszcze systemy wykorzystujące roboty przemysłowe (ang. Robotic Fixtureless Assemblies – RFA), które zastępują tradycyjne podpory zestawami manipulatorów z chwytakami wspólnie podpierającymi detal podlegający obróbce [2, 3]. Przy wykorzystaniu systemów typu RFA różne detale mogą być obrabiane w ramach jednego stanowiska, a przejścia do obróbki detali innego typu są stosunkowo szybkie – rekonfiguracja polega tu na wymianie programu, a nie na zmianie ustawienia mechanicznych części systemu. Jednak istniejące i proponowane systemy typu RFA nadają się tyl-

ko do części sztywnych i stosunkowo niewielkich detali, ponieważ wykorzystują ograniczoną liczbę tradycyjnych, stacjonarnych manipulatorów robotów, które nie zmieniają swojego miejsca zamocowania, a co więcej, wzajemnie sobie przeszkadzają, ponieważ duża część ich przestrzeni roboczej jest wspólna. Proponowany w tym artykule system podpierania łączy zalety systemów RFA i MFPS, a mianowicie zdolność do przemieszczania podpór w czasie i przestrzeni ze zdolnością dostosowywania się głowicy podpierającej do kształtu podpieranej powierzchni, przy zachowaniu wysokiej sztywności podparcia.



**Rys. 2.** Wielorobotowa rekonfigurowalna forma mocująca  
**Fig. 2.** The multi-robot fixture

Zauważono, że w praktyce detal podlegający obróbce musi być sztywny jedynie w pobliżu miejsca, gdzie ta obróbka się odbywa. Ta obserwacja doprowadziła do pomysłu wytworzenia samo-rekonfigurowalnego urządzenia składającego się z roju robotów zdolnych do przemieszczania się pod arkuszem blachy, podpierając ją tylko w aktualnym rejonie obróbki (rys. 2) [4]. Liczba robotów wspomagających produkcję może być różna dla arkuszy o różnej wielkości, jednak system sterujący powinien pozostać niezmienny. Artykuł ten pokazuje, jakie powinny być cechy sterownika takiego systemu. Liczby robotów podpierających nie można określić a priori, więc do projektowania struktury systemu sterowania zostało wybrane podejście wieloagentowe. Specyfikacja systemu opera się na wcześniejszych pracach [9, 11, 13] wykorzystujących podejście agentowe i funkcje przejścia opisujące zachowania każdego agenta. Inspiracją tej metody specyfikacji systemów sterowania robotów był sposób określania znaczenia rozkazów języków programowania przez definiowanie, jaki efekt będzie miało wykonanie ich instrukcji na działanie pewnej abstrakcyjnej maszyny obliczeniowej, a więc określenie ich semantyki operacyjnej [8].

## 1. Koncepcja urządzenia samorekonfigurowalnego

Skonstruowany system działa w następujący sposób. Panele (cienkie blachy ukształtowane przez prasę) wprowadzane są do urządzenia ręcznie. Statyczne zaciski pozycjonują panel w płaszczyźnie poziomej, natomiast siła grawitacji powoduje ugięcie blachy w pionie. Roboty podpierające

przemieszczają się w pobliże miejsca obróbki, aby usztywnić panel tam, gdzie będzie pracowało narzędzie maszyny CNC. Każdy z robotów składa się z trzech części: bazy mobilnej przemieszczającej robota pod panelem, manipulatora o strukturze równoległej (ang. Parallel Kinematic Machine – PKM) poruszającego trzecim elementem, czyli głowicą bezpośrednio podpierającą blachę. Roboty przemieszczają się na ławie, z której wystają elementy dokujące – trzpienie. Baza mobilna ma trzy nogi zakończone elementami dokującymi – dokładniej gniazdami. Owe gniazda nasadzone są na trzpieniu wystające z ławy. Tworzą one na ławie siatkę trójkątów równobocznych. Podstawa robota, podpierając obrabiany panel, jest przytwierdzona do trzech sąsiednich elementów dokujących, tworzących trójkąt. Przemieszczając się do nowego położenia, baza mobilna odłącza się od dwóch trzpieni, podnosi nogi, do których były one przytwierdzone, i obraca się wokół trzeciego elementu dokującego – tego, który pozostał złączony z ławą. Gdy robot w nowym położeniu bezpiecznie przytwierdzi przenoszone nogi do ławy, manipulator unosi głowicę do zdefiniowanej programowo pozycji, zapewniając tym samym podparcie dla obrabianych blach. Po zetknięciu głowicy z blachą wytwarzane jest podciśnienie, które zapobiega odskoczeniu blachy. Początkowo głowica jest podatna, ale po osiągnięciu właściwego położenia jest utwardzana, a podciśnienie utrzymuje blachę w ścisłym kontakcie z robotem.

Program działający off-line, na podstawie geometrycznych danych CAD opisujących panel i danych CAM opisujących przewidzianą procedurę obróbki, generuje plan ruchu robotów. Panele są poddawane wierceniu lub frezowaniu. Wiercenie wykonywane jest, gdy roboty podpierające są nieruchome. Frezowanie wymaga, aby roboty przemieszczały się podczas obróbki. Sterownik nadrzędny korzysta z planu wygenerowanego off-line do sterowania robotami podczas obróbki. Wpływa również na sterownik maszyny CNC, synchronizując pracę robotów z działaniami tej maszyny.

## 2. Urządzenia składowe

Prototypowy system został przedstawiony na rys. 2. Roboty podpierające panel przemieszczają się ponad ławą z trzpieniami, do których dokują się nogi robotów. Ława z jednej strony stanowi rodzaj planszy, po której przemieszczają się roboty, a z drugiej strony jest odpowiedzialna za dostarczenie do nich zasilania i sprężonego powietrza. Oba media są dostarczane za pośrednictwem trzpieni stanowiących elementy dokujące. W chwili dokowania wtyczka znajdująca się na szczycie trzpienia wchodzi w gniazdo umieszczone w nodze robota, natomiast samozamykające się połączenie przewodów pneumatycznych znajduje się w kołnierzu otaczającym podstawę trzpienia.

Baza mobilna każdego robota ma trzy nogi umieszczone w trzech wierzchołkach trójkąta równobocznego [4]. Każda noga może być opuszczona lub podniesiona. Ten ruch jest wykonywany przez dwupołożeniowy siłownik pneumatyczny sterowany elektrycznymi zaworami pneumatycznymi. Wszystkie nogi są w stanie obracać się wokół pionowej osi (osi trzpienia, do którego przytwierdzona jest noga). Jeden silnik elektryczny jest odpowiedzialny za jednoczesny obrót wszystkich nóg. Wszystkie nogi są sprzężone

z silnikiem za pomocą wspólnej przekładni, ich ruch jest zsynchronizowany, tzn. wszystkie nogi utrzymują niezmienną orientację. Nogi są mocowane do trzpieni znajdujących się w ławie przez pneumatyczne urządzenia dokujące (zamki) firmy Schunk. Robot podpierający blachę ma trzy nogi przytwierdzone do ławy. Podczas translacji bazy mobilnej dwa zamki zostają otwarte, odpowiadające im dwie nogi są podnoszone powyżej trzpieni, a robot obraca się wokół trzeciej nogi nadal przytwierdzonej do trzpienia. Manipulator umieszczony na wierzchu bazy mobilnej może być obracany niezależnie. Za ten ruch jest odpowiedzialny drugi, niezależny silnik elektryczny. Stąd baza mobilna wymaga sterowania: dwoma silnikami, trzema siłownikami pneumatycznymi i trzema pneumatycznymi zamkami dokującymi.

Manipulator (PKM) [5, 14] ma 6 stopni swobody, równo podzielonych między równoległą strukturę kinematyczną i kiść sferyczną, a więc sterowanych musi być sześć silników.

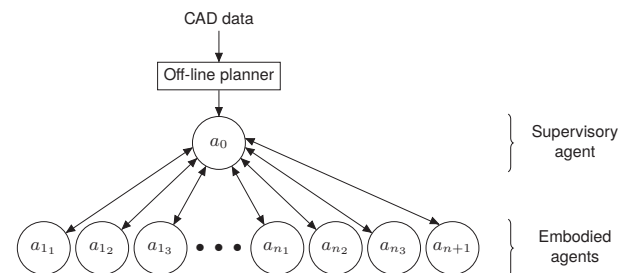
Trójkątna głowica podpierająca panele umieszczona jest na kiści manipulatora. Zawiera ona płyn magnetoreologiczny, który może przyjmować postać ciała stałego lub płynu w zależności od położenia magnesu stałego poruszanego siłownikiem pneumatycznym [1]. Ponadto sterowania wymaga wytwarzanie podciśnienia umożliwiającego przyssanie panelu do głowicy. Oba urządzenia pneumatyczne wymagają jedynie sterowania dwustanowego. Głowica, gdy nie jest przyssana do panelu, może być obracana, co wymaga sterowania pojedynczym dodatkowym silnikiem.

Ława jest odpowiedzialna za dystrybucję energii i powietrza dostarczanego pod ciśnieniem do nóg robotów. Napięcie dołączane jest do łącza znajdującego się w trzpieniu za pomocą przekaźnika. Wyspa zaworowa Festo odpowiada za dystrybucję sprężonego powietrza, między inny pośredniczy w dostarczaniu sprężonego powietrza do czyszczenia trzpieni z wiórów powstających podczas pracy maszyny CNC.

### 3. Struktura układu sterującego i zachowanie systemu

Wszystkie silniki robota połączone są z przemysłowymi sterownikami ruchu Maxon EPOS2, które z kolei przyłączone są do jego komputera pokładowego za pomocą magistrali CAN. Sterowniki EPOS2 mają podwójne zasilanie, aby zapobiec restartowi komputera podczas prawdopodobnych, ale jak dotychczas nigdy nie zaobserwowanych, krótkich przerw w zasilaniu mogących powstać przy rozłączaniu zasilania przy podnoszeniu nóg robota. Uzwojenia silników są zasilane napięciem stałym 48 V, które jest dostarczane niezależnie od stabilizowanego zasilania o napięciu stałym 24 V na potrzeby komputera pokładowego. Wszystkie urządzenia wymagające sterowania dwustanowego są obsługiwane przez wyjścia binarne sterowników EPOS2. Dziewięć takich sterowników wystarcza dla każdego robota: 6 do silników manipulatora, 2 do silników bazy mobilnej i 1 dla silnika obrotu głowicy. Sterowanie zasilaniem ławy jest obsługiwane osobno przez wejścia-wyjścia cyfrowe karty znajdującej się w komputerze nadrzędnym umieszczonym poza robotami, a zawory pneumatyczne sterujące przepływem powietrza do układu dysz czyszczących będących w podstawach trzpieni są obsługiwane przez oddzielną magistralę CAN przyłączy-

ną do tego komputera. Komputer nadrzędny i komputery pokładowe komunikują się za pośrednictwem bezprzewodowej sieci Ethernet. Oprogramowanie uruchomione w tej sieci zbudowane jest z niezależnych agentów programowych.



Rys. 3. Struktura wieloagentowego systemu sterującego  
Fig. 3. Structure of the multi-agent control system

Każdy z robotów (wyróżnionych w tekście indeksem  $j$ ) jest sterowany przez trzy agenty upostaciowione ( $a_{jk}$ ,  $k = 1, 2, 3$ ). Agenty te odpowiadają za sterowanie wyżej opisanymi trzema częściami składowymi robota: manipulatorem steruje agent  $a_{j1}$ , bazą mobilną  $a_{j2}$ , natomiast głowicą  $a_{j3}$  (rys. 3). Ponieważ agenty te związane są z fizyczną postacią urządzeń mechanicznych, którymi sterują, stąd ich nazwa – upostaciowione. W ogólności robotów może być dowolnie wiele  $j = 1, \dots, n$ , ale w przedstawionym prototypowym zestawie  $n = 2$ . Można by zaprojektować monolityczny system sterowania, ale to utrudniłoby niezależny rozwój oprogramowania dla każdej z części składowych robotów. Ponadto początkowo przewidywano, że docelowy system będzie często modyfikowany wraz z rozwojem prototypu. W takich okolicznościach modularność oprogramowania jest koniecznością. Ponieważ trzy części każdego z robotów powstawały oddzielnie, trzy agenty zostały stworzone dla każdego z robotów. Całkowita liczba agentów wynika z liczby robotów w zestawie (pomnożonej przez trzy), a dodatkowo istnieją jeszcze: koordynator systemu  $a_0$  oraz agent  $a_{n+1}$  bezpośrednio sterujący ławą. Oprogramowanie systemu sterowania zostało formalnie wyspecyfikowane, przez zdefiniowanie zachowania agentów za pomocą funkcji przejścia [11]. Agenty upostaciowione sterujące poszczególnymi częściami robotów, co do zasady, mogą komunikować się ze sobą bezpośrednio, jednak w tym przypadku wystarczy komunikacja tych agentów jedynie z koordynatorem. Poniżej wyszczególniono zadania poszczególnych agentów.

Agent sterujący manipulatorem ( $a_{j1}$ ):

- 1) przemieszcza manipulator w taki sposób, że głowica zostanie ustawiona we wskazanej przez koordynatora pozycji podparcia panelu,
- 2) podpira panel,
- 3) opuszcza głowicę do pozycji początkowej. W tej konfiguracji agent pozostaje w stanie uśpienia. W tym stanie baza mobilna może bezpiecznie przemieszczać całego robota.

Pierwsza i trzecia operacja są wykonywane przez to samo zachowanie zdefiniowane pojedynczą odpowiednio sparametryzowaną funkcją przejścia. Druga nie wymaga żadnej aktywności – robot pozostaje nieruchomy.

Agent sterujący bazą mobilną ( $a_{j2}$ ):

- 1) uwalnia dwie nogi z trzpieni,

- 2) podnosi te dwie nogi,
- 3) obraca się na nodze zadokowanej do trzpienia,
- 4) opuszcza podniesione nogi,
- 5) dokuje właśnie opuszczone nogi,
- 6) stoi na wyprostowanych i przytwierdzonych do trzpieni trzech nogach.

Gdy wszystkie nogi robota przytwierdzone są do trzpieni wystających z ławy, agent sterujący bazą mobilną pozostaje w stanie uśpiania, stąd potrzebnych jest jedynie pięć zachowań. W stanie uśpiania agenta  $a_{j2}$ , agent  $a_{j1}$  może poruszać manipulatorem, a w szczególności ustawić głowicę tak, aby podierała obrabiany panel. Baza mobilna podczas pracy ma co najmniej jedną nogę wyprostowaną (cały czas unosi się na tym samym poziomie).

Agent sterujący głowicą ( $a_{j3}$ ):

- 1) obraca głowicę do pożądanej orientacji,
- 2) załącza podciśnienie,
- 3) utwardza zawartość głowicy,
- 4) podpira panel,
- 5) odłącza podciśnienie,

- 6) upłynnia zawartość głowicy.

Podpieranie panelu nie wymaga żadnych działań, a więc znowu pięć zachowań wystarcza.

Agent sterujący ławą ( $a_{n+1}$ ):

- 1) załącza lub wyłącza zasilanie dostarczane do robotów przez trzpienie i nogi,
- 2) załącza lub wyłącza sprężone powietrze używane do czyszczenia trzpieni z wiórów metalowych powstających podczas obróbki mechanicznej.

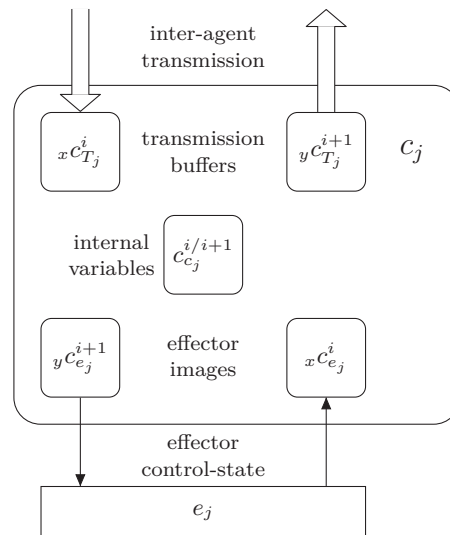
Agent ( $a_0$ ) koordynujący działania całego systemu jest odpowiedzialny za:

- 1) współdziałanie z operatorem,
- 2) czytanie planu stworzonego przez program off-line,
- 3) koordynowanie działań agentów sterujących robotami przez wydawanie odpowiednich poleceń,
- 4) sterowanie agentem sterującym ławą.

#### 4. Sterownik

Struktura każdego agenta w systemie jest podyktowana ogólnymi właściwościami agentów upostaciowionych, które sterują swymi efektorami, uzyskują informacje o otoczeniu przez swoje receptory (dokładniej eksteroreceptory) oraz kontaktują się bezpośrednio z innymi agentami, a ponadto mają własną pamięć wewnętrzną (rys. 4) [11, 13].

Generalnie każdy taki agent ma możliwość: wydawania poleceń swojemu efektorowi, konfigurowania swoich eksteroreceptorów, dostarczania informacji do innych agentów, jak również aktualizację własnej pamięci. Definiują to jego funkcje przejścia, które wykonują niezbędne obliczenia na podstawie: proprioceptywnej informacji uzyskanej z efektorów, zagregowanych odczytów z eksteroreceptorów, informacji dostarczanej przez inne agenty oraz zawartość własnej pamięci. Jednak w tym konkretnym przypadku agenty nie wykorzystują eksteroreceptorów, a ich kontakt z innymi agentami jest ograniczony jedynie do wymiany informacji z koordynatorem. Ponadto ta wymiana danych



Rys. 4. Struktura agentów upostaciowionych:  $a_{j1}, a_{j2}, a_{j3}$   
Fig. 4. Structure of embodied agents:  $a_{j1}, a_{j2}, a_{j3}$

```
// Wyznacz stan początkowy -  $i = i^0$ 
 $e_j \mapsto xC_{e_j}^i; V_j \mapsto xC_{V_j}^i;$ 
.....
loop
// Pobierz polecenie od koordynatora
 $yC_{T_{0j}}^i \mapsto xC_{T_{0j}}^i;$ 
// Zinterpretuj polecenie
.....
// Wykonaj polecenie (operację zdefiniowaną
// funkcją przejścia i warunkiem końcowym)
loop // Wykonanie operacji
// Sprawdź warunek końcowy
if  $m_{f_{T_j}}(xC_{e_j}^i) = \text{false}$  then
// Wyznacz następny stan
// podsystemu sterowania
 $yC_{e_j}^{i+1} := m_{f'_{e_j}}(xC_{e_j}^i);$ 
// Prześlij wyniki obliczeń
 $yC_{e_j}^{i+1} \mapsto e_j; yC_{V_j}^{i+1} \mapsto V_j;$ 
// Czekaj na następną iterację
 $i := i + 1;$ 
// Wyznacz bieżący stan agenta
 $e_j \mapsto xC_{e_j}^i; V_j \mapsto xC_{V_j}^i;$ 
endif
endloop; // Koniec operacji
// Poinformuj koordynatora o wykonaniu polecenia
 $yC_{T_{0j}}^i \mapsto xC_{T_{0j}}^i;$ 
endloop;
```

Rys. 5. Pseudokod agenta upostaciowionego:  $a_{j1}, a_{j2}, a_{j3}$   
Fig. 5. Pseudocode of an embodied agent:  $a_{j1}, a_{j2}, a_{j3}$

zachodzi sporadycznie – tylko wtedy, gdy koordynator zleca nowe zachowanie, po tym jak poprzednie zostało zakończone. Te założenia znacznie uprościły wewnętrzną strukturę programu sterującego pracą agentów upostacio-

wionych. Struktura agentów upostaciowionych wyraża się pseudokodem przedstawionym na rys. 5, gdzie  $\mapsto$  oznacza transmisję danych pomiędzy elementami agenta lub między agentami,  $j$  jest numerem agenta (0 dla koordynatora),  $e$  oznacza efektor,  $T$  bufor transmisji międzyagentowej,  $c$  jest podsystemem sterowania agenta,  $x$  i  $y$  odnoszą się do buforów wejściowych lub odpowiednio wyjściowych podsystemu sterowania,  $f_c$  to funkcja przejścia,  $f_\tau$  jest warunkiem końcowym, a  $m$  jest etykietą pary tych dwóch funkcji, natomiast  $i$  jest numerem dyskretnej chwili czasu (operacja rozpoczyna swoje wykonanie w chwili  $i_0$ ). Zachowanie agenta w czasie wykonania wewnętrznej pętli tego kodu jest zdefiniowane przez parę funkcji skomponowana z funkcji przejścia oraz warunku końcowego. Koordynator (agent  $a_0$ ) wskazuje do wykonania właśnie te pary funkcji. Zostały one zdefiniowane formalnie, a następnie zrealizowane zgodnie z tą specyfikacją. Zawilość specyfikacji funkcji przejścia zależy od wymaganego zachowania i sposobu reprezentacji sterowanego urządzenia. W omawianym przypadku najbardziej skomplikowanym zachowaniem był ruch manipulatora po prostej określonej w przestrzeni kartezjańskiej, biorący pod uwagę ograniczenia narzucone zarówno na prędkości jak i przyspieszenia ruchu poszczególnych silników oraz możliwości urządzenia bezpośrednio sterującego wykonaniem wygenerowanej trajektorii, czyli sterowników ruchu EPOS2 [12]. Warunki końcowe mają formę predykatów (funkcji o wartościach boolowskich), a więc są one proste do zdefiniowania i łatwe do implementacji.

Działanie koordynatora (agenta  $a_0$ ) sprowadza się jedynie do dwóch zachowań, które zostały wyspecyfikowane formalnie. Pierwsze z nich odpowiada za wydawanie poleceń podległym agentom upostaciowionym, drugie sprowadza się do zatrzymania działania systemu w przypadku awarii któregoś z jego składników. Koordynator jest pobudzany przez dwa rodzaje zdarzeń: wewnętrzny zegar taktowany adekwatnie do planu wygenerowanego off-line oraz przez informacje uzyskane od agentów upostaciowionych. Dane przesyłane z koordynatora do poszczególnych agentów składają się z identyfikatora polecenia i jego parametrów, tj.: stanu skupienia materiału magnetoreologicznego w głowicy, potrzeby generacji podciśnienia przez głowicę, jej orientacji, pozycji docelowej manipulatora, sekwencji ruchów bazy mobilnej oraz zestawu trzpieni ławy, które mają dostarczyć napięcie i sprężone powietrze. Pozostałe agenty potwierdzają pomyślnie zakończenie uprzednio zleconej operacji lub raportują awarię, w razie jej wystąpienia. Koordynator rejestruje te zdarzenia i wykrywa ewentualne przekroczenie granicznych czasów wykonania operacji zdefiniowanych w planie.

## 5. Podsumowanie

System sterujący został zaimplementowany przy użyciu programowej struktury ramowej MRROC++ [10, 13]. Modularność oprogramowania sterującego okazała się istotną zaletą podczas wstępnego badania poszczególnych części robota, a także podczas ostatecznej integracji systemu. Wybrane elementy dostępne na rynku okazały się działać poprawnie, pomimo znacznych zakłóceń elektromagnetycznych generowanych przez liczne urządzenia produkcyjne zainstalowane w pobliżu. Zrealizowane plany działań wy-

magwały do kilku pozycji podparcia i do 10 translokacji baz mobilnych. Podobne trajektorie były używane zarówno do wiercenia jak i frezowania, gdyż większość detali wymaga wywiercenia otworów wzdłuż pewnych konturów. Parametry kinematyczne systemu zostały skalibrowane przy użyciu maszyny CNC zastosowanej jako narzędzie pomiarowe.

Wykonano szereg eksperymentów, aby ocenić funkcjonowanie systemu sterowania. Poszczególne agenty zostały uznane za poprawnie działające przez wykonanie poleceń ruchu przykładowego planu krok po kroku. Następnie uruchamiano kompletne sekwencje ruchów. Szczegółowe wskaźniki jakości, które odnoszą się do wymagań mechanicznych systemu (np. sztywność podczas podparcia detali lub czas translokacji bazy mobilnej) zależą od parametrów mechanicznych urządzeń napędowych oraz elementów przeniesienia napędu. System sterowania nie ma negatywnego wpływu na ich działanie. Celem eksperymentów była ocena funkcjonowania poszczególnych agentów oraz ich współdziałania. Zarówno części jak i całość funkcjonowały poprawnie.

W drugiej części tego artykułu zostanie przedstawiony automatyczny sposób generowania planów, które opisany tu układ sterowania realizuje.

## Podziękowania

Badania w ramach projektu SwarmItFIX były finansowane w ramach 7. Programu Ramowego (wspólny projekt 214678). Partnerami w ramach projektu były następujące instytucje: University of Genova (Włochy), Piaggio Aero Industries (Włochy), Exechon (Szwecja), ZTS VVÜ Košice a.s. (Słowacja), Fiat (Włochy) i Politechnika Warszawska (Polska). Publikacja ta jest finansowana ze środków statutowych Politechniki Warszawskiej.

## Bibliografia

1. Avvenente, R., Khan, M. A., Li, X., Zoppi, M., Zlatanov, D., Molino, R. M. (2010): *Development and Analysis of a Shape-Conformable Supporting Head for a Self-Reconfigurable Intelligent Swarm Fixture System*. In: ISR/ROBOTIK'10, pp. 792–799.
2. Bi, Z. M., Zhang, W. J. (2001): *Flexible fixture design and automation: review, issues and future directions*. International Journal of Production Research 39(13), 2867–2894.
3. Kang, Y., Rong, Y., Yang, J., Ma, W. (2002): *Computer-aided fixture design verification*. Assembly Automation 22(4), 350–359.
4. Molino, R., Zoppi, M., Zlatanov, D. (2009): *Reconfigurable Swarm Fixtures*. In: ASME/IFTOMM International Conference on Reconfigurable Mechanisms and Robots, pp. 730–735.
5. Neumann, K. (1988): *US Patent Number 4732525*.
6. Sela, M. N., Gaudry, O., Dombre, E., Benhabib, B. (1997): *A reconfigurable modular fixturing system for thin-walled flexible objects*. The International Journal of Advanced Manufacturing Technology 13(9), 611–617.
7. Shirinzadeh, B., Tie, Y. (1995): *Experimental investigation of the performance of a reconfigurable fixturing system*. The International Journal of Advanced Manufacturing Technology 10(5), 330–341.

8. Slonnegger, K., Kurtz, B. L. (1995): *Formal Syntax and Semantics of Programming Languages: A Laboratory Based Approach*. Addison-Wesley Publishing Company, Reading.
9. Winiarski, T., Zieliński, C. (2008): *Sterowanie interakcją manipulatora ze środowiskiem (część pierwsza i druga)*. In: X Krajowa Konferencja Robotyki – Problemy Robotyki, volume 2, pp. 473–492, Oficyna Wydawnicza Politechniki Warszawskiej.
10. Zieliński, C. (1999): *The MRROC++ System*. In: Proceedings of the First Workshop on Robot Motion and Control, RoMoCo'99, pp. 147–152.
11. Zieliński, C. (2006): *Transition-Function Based Approach to Structuring Robot Control Software*. In: Kozłowski, K. (Ed.), Robot Motion and Control, *Lecture Notes in Control and Information Sciences*, volume 335, pp. 265–286, Springer-Verlag.
12. Zieliński, C., Kornuta, T., Trojanek, P., Winiarski, T., Wałęcki, M. (2012): *Specification of a Multi-agent Robot-Based Reconfigurable Fixture Control System*. Robot Motion & Control 2011 (Lecture Notes in Control & Information Sciences) 422, 171–182.
13. Zieliński, C., Winiarski, T. (2010): *Motion Generation in the MRROC++ Robot Programming Framework*. International Journal of Robotics Research 29(4), 386–413.
14. Zoppi, M., Zlatanov, D., Molfino, R. (2010): *Kinematics analysis of the Exechon tripod*. In: Proceedings of the ASME DETC, 34th Annual Mechanisms and Robotics Conference (MR), Montreal, Canada.

### Multi-Robot-Based Reconfigurable Fixture – Control System

**Abstract:** Machining fixtures must fit exactly the work piece to support it appropriately. Even slight change in the design of the work piece renders the costly fixture useless. Substitution of traditional fixtures by a programmable multi-robot system supporting the work pieces requires a specific control system and a specific programming method enabling its quick reconfiguration. The multi-robot control system has been designed following a formal approach based on the definition of the system structure in terms of agents and transition function definition of their behaviour. Thus a modular system resulted, enabling software parameterisation. This facilitated the introduction of changes brought about by testing different variants of the mechanical structure of the system. The shop-floor experiments with the system showed that the work piece is held stiffly enough for both milling and drilling operations performed by the CNC machine. If the number of diverse work piece shapes is large the reconfigurable fixture is a cost-effective alternative to the necessary multitude of traditional fixtures. Moreover, the proposed design approach enables the control system to handle a variable number of controlled robots and accommodates possible changes to the hardware of the work piece supporting robots. The first part of the paper introduces the fixturing problem and presents the control system of the designed multi-robot fixture, while the second part presents the planer deciding where and when the supports should be located.

**Keywords:** reconfigurable fixtures, robotic fixtureless assemblies, multi-robot systems, multi-agent systems, control systems

#### prof. dr hab. inż. Cezary Zieliński

Jest profesorem na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. W latach 2002-2005 sprawował na tym wydziale funkcję prodziekana ds. nauki i współpracy międzynarodowej, 2005-2008 zastępcy dyrektora Instytutu Automatyki i Informatyki Stosowanej (IAiIS) ds. naukowych, a od 2008 pełni funkcję dyrektora tego instytutu. Od roku 1996 jest kierownikiem Zespołu Robotyki w IAiIS. Od 2007 roku jest członkiem Komitetu Automatyki i Robotyki Polskiej Akademii Nauk. Od 2008 roku współpracuje z Przemysłowym Instytutem Automatyki i Pomiarów. Jego zainteresowania badawcze koncentrują się na zagadnieniach związanych z programowaniem i sterowaniem robotów.

e-mail: [c.zielinski@ia.pw.edu.pl](mailto:c.zielinski@ia.pw.edu.pl)



#### mgr inż. Tomasz Kornuta

Absolwent Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. W 2003 roku uzyskał tytuł inżyniera, w 2005 tytuł magistra inżyniera. Od 2008 roku pracuje na etacie asystenta w Instytucie Automatyki i Informatyki Stosowanej (IAiIS), w ramach którego prowadzi zajęcia dydaktyczne, a od 2009 roku pełni funkcję Kierownika Laboratorium Podstaw Robotyki. Od 2005 roku w ramach doktoratu prowadzi badania związane z projektowaniem systemów robotycznych wykorzystujących paradygmat aktywnego czucia do analizy otoczenia. Jego główne zainteresowania naukowe obejmują wykorzystanie informacji wizyjnej w robotyce.

e-mail: [tkornuta@ia.pw.edu.pl](mailto:tkornuta@ia.pw.edu.pl)



#### mgr inż. Piotr Trojanek

Jest doktorantem oraz pracuje jako konstruktor w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Swoje doświadczenie zdobywał pracując nad układami sterowania robotów mobilnych. Obecnie prowadzi badania dotyczące systemów wieloagentowych oraz zastosowań metod inżynierii oprogramowania w robotyce. Od lat związany ze studenckim kołem naukowym Bionik.

e-mail: [piotr.trojanek@gmail.com](mailto:piotr.trojanek@gmail.com)



#### dr inż. Wojciech Szykiewicz

Jest adiunktem na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. W latach 1999-2003 był zastępcą dyrektora i sekretarzem rady naukowej Centrum Automatyki i Technik Decyzyjnych. Jego zainteresowania badawcze dotyczą problematyki planowania ruchu robotów mobilnych i manipulatorów, a także manipulacji robotów.

e-mail: [W.Szykiewicz@elka.pw.edu.pl](mailto:W.Szykiewicz@elka.pw.edu.pl)



**dr inż. Tomasz Winiarski**

Jest adiunktem w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Sprawuje funkcje kierownika laboratorium robotyki w macierzystym instytucie, a także opiekuna studenckiego koła naukowego Bionik, które współtworzył i z którym organizował imprezy popularyzujące robotykę oraz realizował granty badawcze. W 2010 r. otrzymał za osiągnięcia naukowe nagrodę indywidualną drugiego stopnia rektora PW, w 2011 r. wyróżnienie w konkursie innowator mazowska, a także pierwszą nagrodę w konkursie młodzi innowacyjni. Jego zainteresowania naukowe dotyczą z jednej strony konstrukcji i nawigacji robotów mobilnych dedykowanych do zadań usługowych, z drugiej strony specyfikacji i implementacji zadań manipulatorów i chwytaków ze szczególnym uwzględnieniem hybrydowego sterowania pozycyjno-siłowego oraz sterowania impedancyjnego.

*e-mail: [tmwiniarski@gmail.com](mailto:tmwiniarski@gmail.com)*



**mgr inż. Michał Wałęcki**

W latach 2004–2010 studiował na Wydziale Elektroniki i Technik Informatycznych PW, uzyskując tytuł magistra inżyniera ze specjalnością Elektronika i Inżynieria Komputerowa. W 2010 rozpoczął studia doktoranckie w Instytucie Automatyki i Informatyki Stosowanej PW. Uczestniczył w wielu projektach związanych z robotyką – zarówno naukowych, jak i komercyjnych. W latach 2010-2011 pracował w IAiS PW na stanowisku Konstruktora, obecnie od 2011 zajmuje stanowisko asystenta. Jego zainteresowania naukowe obejmują zagadnienia związane z planowaniem i modyfikacją on-line trajektorii manipulatora. Interesuje się również



specjalizowanymi konstrukcjami robotyki mobilnej oraz sprzętowymi sterownikami do zadań robotycznych.

*e-mail: [m.walecki@elka.pw.edu.pl](mailto:m.walecki@elka.pw.edu.pl)*

**prof. nzw. dr hab. inż. Włodzimierz Kasprzak**

Jest pracownikiem naukowo-dydaktycznym Politechniki Warszawskiej od 1997 r. W latach 1988-1996 prowadził badania naukowe w Niemczech i Japonii w zakresie wizji komputerowej i analizy sygnałów. Z zawodu jest inż. informatyki, w 2002 r. uzyskał stopień doktora habilitowanego ze specjalnością Automatyka i Robotyka, od 2005 r. jest profesorem nadzwyczajnym w Politechnice Warszawskiej. Specjalizuje się w zagadnieniach rozpoznawania obrazów i sygnałów mowy oraz metod sztucznej inteligencji. Jest autorem ponad 100 prac naukowych z tej dziedziny, w tym 3 książek.

*e-mail: [W.Kasprzak@elka.pw.edu.pl](mailto:W.Kasprzak@elka.pw.edu.pl)*



**prof. dr hab. inż. Teresa Zielińska**

Profesor, pracownik Wydziału Mechanicznego Energetyki i Lotnictwa Politechniki Warszawskiej. W latach 2002-2008 kierownik Zakładu Teorii Maszyn i Robotów na tym Wydziale, od 2008 pełni funkcję prodziekana ds. ogólnych. Jest Sekretarzem Generalnym International Federation for the Promotion of Mechanism and Machine Science (IFToMM), oraz opiekunem międzynarodowych studiów European Master on Advanced Robotics (EMARO) programu Erasmus Mundus. Członek Komitetu Automatyki i Robotyki Polskiej Akademii Nauk. Jej zainteresowania badawcze koncentrują się na zagadnieniach związanych z maszynami kroczącymi, z planowaniem i sterowaniem ruchu robotów oraz z biorobotyką.

*e-mail: [teresaz@meil.pw.edu.pl](mailto:teresaz@meil.pw.edu.pl)*

