
Sterowanie zespołu heterogenicznych robotów na przykładzie pchania pudła*

Cezary Zieliński¹, Wojciech Szynkiewicz¹, Piotr Trojanek¹,
Marek Majchrowski¹

Streszczenie

W artykule przedstawiono realizację zadania wspólnego pchania pudła przez zespół heterogenicznych robotów mobilnych. Zaproponowano formalny opis zadania z podziałem ról dla poszczególnych robotów z wykorzystaniem funkcji przejścia. Proponowane podejście zostało zweryfikowane zarówno w badaniach symulacyjnych, jak i na rzeczywistych robotach. W artykule zamieszczono wyniki realizacji zadania pchania dużego obiektu z wykorzystaniem trzech robotów nieholonomicznych Elektron.

1. WSTĘP

Doświadczenia eksperymentalne pokazują, że wiele prostych robotów o niskim poziomie „inteligencji maszynowej” może ze sobą współpracować w celu realizacji nawet złożonego zadania [1, 3, 4]. 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 planowania działań zależy od bardzo wielu czynników m.in. liczby i typu robotów, stopnia ich współpracy, rodzaju środowiska (statyczne/dynamiczne, znane/nieznane) itp., [1, 2, 13]. Jednym z zadań testowych dla współpracujących robotów mobilnych jest wspólne pchanie pudła (*box pushing*) przez kilka robotów [4, 11, 10]. Jest ono szczególnym przypadkiem zadania transportu obiektu. Pchanie pudła odbywa się na płaszczyźnie, podczas gdy transport obiektów zazwyczaj wymaga podnoszenia i przenoszenia przedmiotów zespołowo przez kilka robotów co istotnie zwiększa trudność zadania [5]. Wspólne przemieszczanie obiektów jest przykładem zadania testowego dla takich problemów jak koordynacja ruchów oraz utrzymanie szyku (formacji) robotów [4, 5, 11, 10]. Wśród proponowanych podejść do zadania pchania pudła można wyróżnić podejście behawioralne polegające na doborze zachowań i takim ich rozdziale między roboty aby osiągnąć pożądaný cel [4, 6, 10]. Inny sposób polega na zamknięciu obiektu, czyli otoczeniu obiektu przez trzy lub więcej robotów, a następnie zachowanie tych więzów w trakcie całego ruchu [7].

*Praca jest finansowana przez grant MEiN: 3 T11C038 29

¹Instytut Automatyki i Informatyki Stosowanej, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska, ul. Nowowiejska 15/19, 00-665 Warszawa, [C.Zielinski, W.Szynkiewicz, P.Trojanek, M.Majchrowski]@ia.pw.edu.pl

Opisane w tym artykule wspólne pchanie pudła jest reprezentantem szerszej klasy zadań wymagających kooperatywnego przemieszczania przedmiotów bez ich unieruchomienia względem agentów przemieszczających. W życiu codziennym często spotykamy się z koniecznością zmiany umebłowania pomieszczeń, a co za tym idzie poprzesuwanie ciężkich mebli, a to zazwyczaj wymaga współpracy osób pchających. W niniejszej pracy przedstawiono sformułowanie zadania, implementację systemu sterującego realizującego to zadanie oraz eksperymentalną weryfikację poprawnego jego wykonania przez grupę składającą się z trzech nieholonomicznych robotów Elektron.

2. STRUKTURA STEROWANIA ROBOTA

Do prezentacji struktury systemu sterującego zespołem robotów użyto formalizmu zdefiniowanego w [13, 12]. Formalizm ten, z jednej strony, określa sposób zachowania się pojedynczego agenta (przez zdefiniowanie jego funkcji przejścia), a z drugiej strony, opisuje przepływy informacji niezbędnych do sterowania zarówno w agencie jak i pomiędzy agentami. Z punktu widzenia implementacji oprogramowania systemu sterującego formalny zapis funkcji przejścia oraz przepływu danych wewnątrz układu stanowi specyfikację, według której projektant tworzy system. Efektem ubocznym wprowadzenia formalizmu jest stworzenie narzędzia do przeprowadzenia racjonalnych rozważań na temat funkcji poszczególnych jego elementów składowych. Wtedy dyskusja sprowadza się do określenia postaci poszczególnych funkcji przejścia – matematyczna postać tych funkcji określa zachowanie się agentów, natomiast argumenty tych funkcji określają przepływy danych pomiędzy elementami składowymi agentów jak i pomiędzy nimi. Dla zwięzłości oraz ze względu na oczywistość kontekstu oznaczenia składowych układu sterowania robota oraz ich stanów nie będą tu rozróżniane. Stan robota a_j , $j = 0, \dots, n_a$, traktowanego tutaj jako agenta upostaciowionego, jest wyrażany jako:

$$s_j = \langle c_j, e_j, V_j, T_j \rangle \quad (1)$$

- c_j – stan **podsystemu sterującego** robotem (pamięć: zmienne, program etc.),
- e_j – stan jego **efektorów**,
- V_j – zestaw odczytów **czujników wirtualnych** wykorzystywanych przez robota,
- T_j – informacja wymieniana z innymi robotami.

Zestaw odczytów czujników wirtualnych robota a_j zawiera n_{v_j} składowych:

$$V_j = \langle v_{j_1}, \dots, v_{j_{n_{v_j}}} \rangle \quad (2)$$

Odczyt każdego czujnika wirtualnego v_{j_k} , $k = 1, \dots, n_{v_j}$, tworzony jest przez agregację odczytów z jednego lub więcej eksteroreceptorów. Tak więc k -ty czujnik wirtualny agenta a_j dostarcza odczyt:

$$v_{j_k} = f_{v_{j_k}}(c_j, R_{j_k}), \quad (3)$$

gdzie R_{j_k} jest zestawem odczytów z eksteroreceptorów skojarzonych z k -tym czujnikiem wirtualnym:

$$R_{j_k} = \langle r_{j_{k_1}}, \dots, r_{j_{k_{n_r}}} \rangle, \quad (4)$$

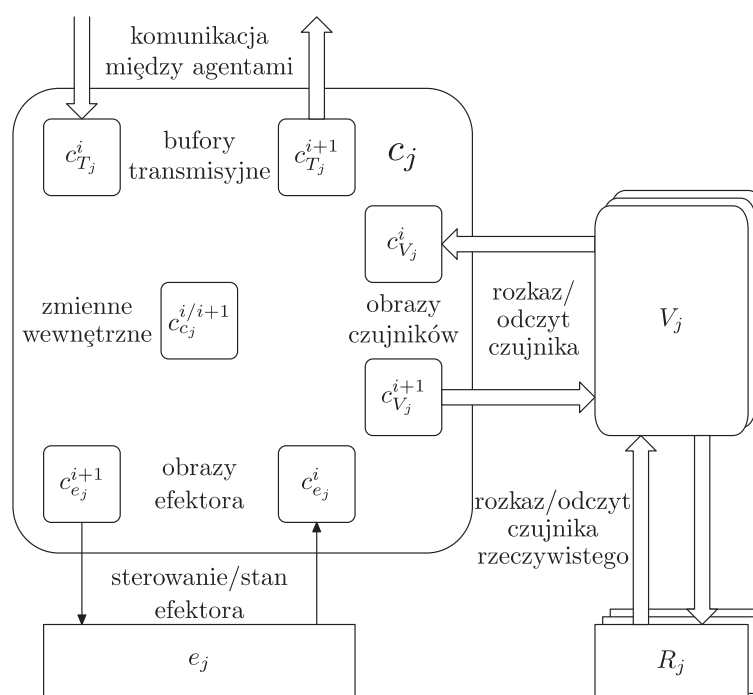
gdzie n_r jest liczbą tych eksteroreceptorów. Intereoreceptory dostarczają informacji o stanie wewnętrznym podsystemów robota, natomiast proprioreceptory określają stan jego efektorów, przez co te grupy czujników związane są z innymi podsystemami niż wyróżnione tutaj czujniki wirtualne.

Zadaniem układu sterowania c_j robota a_j jest:

- zebrać informacje o stanie środowiska agenta poprzez zestaw czujników wirtualnych V_j ,
- uzyskać informacje od innych robotów $a_{j'}$ ($j' \neq j$),
- śledzić stan swych własnych efektorów e_j

i na tej podstawie oraz korzystając ze zgromadzonej informacji w wewnętrznych strukturach danych układu sterującego robota (c_j) wytworzyć:

- nowy (pożądany) stan efektorów e_j ,
- wpłynąć na przyszłe działanie czujników wirtualnych V_j ,
- wysłać informacje do innych robotów $a_{j'}$,
- ustalić nowe wartości c_j wewnętrznych struktur danych układu sterującego robotem.



Rys. 1. Ogólna struktura sterowania pojedynczym robotem

Dla efektywnego wykonania tych operacji zdekomponowano podsystem sterujący c_j w następujący sposób (rys. 1):

- c_{e_j} – obraz efektora, czyli reprezentacja efektora w podsystemie sterowania (np. położenia wałów silników, prędkość liniowa i obrotowa korpusu robota mobilnego),
- c_{V_j} – obrazy czujników wirtualnych, czyli odczyty czujników wirtualnych i ich aktualna konfiguracja,
- c_{T_j} – bufony transmisyjne, czyli informacja wzajemnie przekazywana między agentami,
- c_{c_j} – pozostałe zmienne istotne dla sterowania agentem.

Tak więc wyróżniono trzy typy komponentów podsystemu sterowania:

- komponenty dostarczające dane wejściowe: o stanie efektorów, odczytach czujników wirtualnych oraz informacji uzyskanej od innych robotów (te komponenty wyróżniono dolnym lewym indeksem x),
- komponenty przekazujące dane wyjściowe z układu sterującego do efektorów, czujników wirtualnych oraz do innych robotów (te komponenty wyróżniono dolnym lewym indeksem y),
- inne komponenty potrzebne do przetwarzania danych w podsystemie sterowania (bez wyróżniającego indeksu).

W konsekwencji wyróżniono następujące części składowe (rys. 1):

- xc_{e_j} – informacja o aktualnym stanie efektora,
- xc_{V_j} – odczyty z czujników wirtualnych,
- xc_{T_j} – informacja uzyskana od innych robotów,
- yc_{e_j} – sterowanie efekтором,
- yc_{V_j} – polecenia dla czujników wirtualnych,
- yc_{T_j} – informacja dla innych robotów,
- c_{c_j} – wewnętrzne struktury danych układu sterowania.

Stan systemu zmienia się z pewnym kwantem czasu. Niech i będzie numerem aktualnego kwantu czasu. Wtedy kolejny kwant oznaczany jest jako $i + 1$. Ponadto przyjęto, że upływ czasu i dla każdego robota jest niezależny, ale nie wprowadzono oddzielnego symbolu dla zaznaczenia tego faktu, gdyż symbol, którego indeksem jest i , zawiera również indeks j , a więc w pełni determinuje czas, o który chodzi. W prowadzonych tu rozważaniach nie wymaga się synchronizmu pracy zegarów niezależnych układów sterujących pojazdami.

Podsystem sterowania robota a_j na podstawie:

$$xc_j^i = \langle c_{c_j}^i, xc_{e_j}^i, xc_{V_j}^i, xc_{T_j}^i \rangle \quad (5)$$

wytwarza:

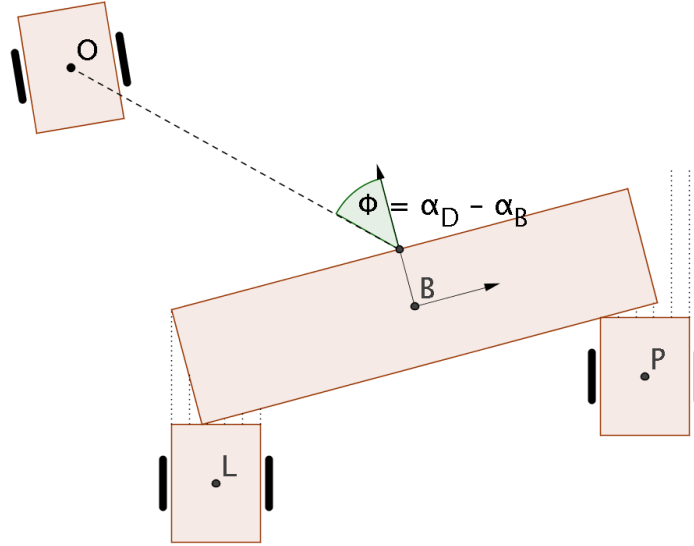
$$yc_j^{i+1} = \langle c_{c_j}^{i+1}, yc_{e_j}^{i+1}, yc_{V_j}^{i+1}, yc_{T_j}^{i+1} \rangle \quad (6)$$

Do tego celu wykorzystuje **funkcje przejścia**:

$$\begin{cases} c_{c_j}^{i+1} &= f_{c_{c_j}}(c_{c_j}^i, xc_{e_j}^i, xc_{V_j}^i, xc_{T_j}^i) \\ yc_{e_j}^{i+1} &= f_{c_{e_j}}(c_{c_j}^i, xc_{e_j}^i, xc_{V_j}^i, xc_{T_j}^i) \\ yc_{V_j}^{i+1} &= f_{c_{V_j}}(c_{c_j}^i, xc_{e_j}^i, xc_{V_j}^i, xc_{T_j}^i) \\ yc_{T_j}^{i+1} &= f_{c_{T_j}}(c_{c_j}^i, xc_{e_j}^i, xc_{V_j}^i, xc_{T_j}^i) \end{cases} \quad (7)$$

3. PCHANIE PUDŁA

Trzy roboty a_j , $j \in \{0, L, P\}$, mają za zadanie wspólnie pchać pudło B . Robot a_0 jest robotem koordynującym. Jest on wyposażony w skaner laserowy (eksteroreceptor), dzięki czemu jest w stanie określić orientację pudła względem wybranego kierunku. Pozostałe dwa roboty faktycznie pchają pudło. W tym celu ustawiają się z jednej jego strony, tak aby czoło każdego z nich częściowo wystawało za róg pudła, robot a_L wystaje poza lewy róg, natomiast a_P – poza prawy. Każdy z robotów a_L , a_P w zderzaku przednim ma zainstalowane czujniki odległości (eksteroreceptory), na podstawie których jest w stanie ocenić jaka jest relacja pomiędzy rogiem pudła a środkiem pojazdu (rys. 2). Stosując opisany powyżej formalizm, zadania robotów mogą być określone w poniższy sposób. W tym opisie pominięto składowe stanu nieistotne z punktu widzenia realizacji zadania.



Rys. 2. Schemat ustawienia robotów w trakcie wykonywania zadania

Stan układu sterującego robota nadzorującego a_0 w chwili i jest następujący:

$$\begin{cases} c_{c_0}^i &= \langle \alpha_D^i, w_{D_0}^i, \omega_{D_0}^i, \alpha_D^{i-1}, \alpha_B^{i-1}, u^{i-1} \rangle \\ x_{e_0}^i &= \langle w_{E_0}^i, \omega_{E_0}^i \rangle \\ x_{V_0}^i &= \langle \alpha_B^i \rangle \end{cases} \quad (8)$$

gdzie:

- α_B^i – azymut pudła względem globalnego układu odniesienia odczytany za pomocą skanera laserowego w chwili i
- α_D^i – pożądaný azymut pudła w chwili $i + 1$ – wartość zadana pobrana z pamięci w chwili i , ale do wykorzystania w chwili $i + 1$
- α_B^{i-1} – azymut pudła względem globalnego układu odniesienia odczytany za pomocą skanera laserowego w chwili $i - 1$, wartość pobrana z pamięci w chwili i
- α_D^{i-1} – pożądaný azymut pudła w chwili i – wartość zadana pobrana z pamięci w chwili i
- $w_{D_0}^i$ – pożądana prędkość liniowa korpusu robota koordynującego dla chwili $i + 1$ – wartość zadana pobrana z pamięci w chwili i , ale do wykorzystania w chwili $i + 1$
- $\omega_{D_0}^i$ – pożądana prędkość obrotowa korpusu robota koordynującego dla chwili $i + 1$ – wartość zadana pobrana z pamięci w chwili i , ale do wykorzystania w chwili $i + 1$
- $w_{E_0}^i$ – odczytana prędkość liniowa korpusu robota koordynującego dla chwili i
- $\omega_{E_0}^i$ – odczytana prędkość obrotowa korpusu robota koordynującego dla chwili i
- u^{i-1} – wyjście układu regulacji prędkości liniowej robotów w chwili $i - 1$

Na tej podstawie robot koordynujący oblicza sterowanie dla chwili $i + 1$.

$$\begin{cases} y_{c_{e_0}}^{i+1} = f_{c_{e_j}}(\langle w_{D_0}^i, \omega_{D_0}^i \rangle, \langle w_{E_0}^i, \omega_{E_0}^i \rangle) \\ y_{c_{T_0}}^{i+1} = f_{c_{T_j}}(\langle \alpha_D^i, \alpha_D^{i-1}, \alpha_B^{i-1}, u^{i-1} \rangle, \langle \alpha_B^i \rangle), \end{cases} \quad (9)$$

gdzie $y_{c_{e_0}}^{i+1}$ stanowi nowe sterowanie ruchem robota koordynującego, natomiast $y_{c_{T_0}}^{i+1}$ jest informacją przekazywaną robotom pchającym, jak mają się zachować w następnym kwancie czasu.

$$y_{c_{T_0}}^{i+1} = \langle y_{c_{T_{0L}}}^{i+1}, y_{c_{T_{0P}}}^{i+1} \rangle, \quad (10)$$

gdzie:

$$\left\{ \begin{array}{l} u^i = u^{i-1} + K \left(1 + \frac{T_p}{2T_i} \right) \frac{|\alpha_D^i - \alpha_B^i|}{\pi/2} + K \left(\frac{T_p}{2T_i} - 1 \right) \frac{|\alpha_D^{i-1} - \alpha_B^{i-1}|}{\pi/2} \\ \left\{ \begin{array}{l} y_{c_{T_{0L}}}^{i+1} = w_{max} (1 - u^i) \\ y_{c_{T_{0P}}}^{i+1} = w_{max} \end{array} \right. \quad \text{dla } \alpha_D^i - \alpha_B^i > 0 \\ \left\{ \begin{array}{l} y_{c_{T_{0L}}}^{i+1} = w_{max} \\ y_{c_{T_{0P}}}^{i+1} = w_{max} \end{array} \right. \quad \text{dla } \alpha_D^i - \alpha_B^i = 0 \\ \left\{ \begin{array}{l} y_{c_{T_{0L}}}^{i+1} = w_{max} \\ y_{c_{T_{0P}}}^{i+1} = w_{max} (1 - u^i) \end{array} \right. \quad \text{dla } \alpha_D^i - \alpha_B^i < 0, \end{array} \right. \quad (11)$$

gdzie w_{max} jest maksymalną prędkością, z jaką może poruszać się pojazd, natomiast $\alpha_D^i - \alpha_B^i > 0$ oznacza skręt pudła w lewo, stąd robot a_L musi poruszać się wolniej niż robot a_P . Dla $\alpha_D^i - \alpha_B^i < 0$ pudło skręca w prawo, natomiast dla $\alpha_D^i - \alpha_B^i = 0$ pchane jest na wprost (oczywiście w rzeczywistej implementacji wprowadzono ustalone

eksperymentalnie marginesy). Element u^i jest sterowaniem wyznaczonym przez regulator PI, gdzie człon całkujący jest wymagany ze względu na różnice w poślizgu kół obydwu robotów (stałe K i T_i zostały dobrane eksperymentalnie, $T_p = 100\text{ms}$).

Stan każdego z robotów pchających a_j , $j \in \{L, P\}$ w chwili i jest określony w następujący sposób:

$$\begin{cases} c_{c_j}^i &= \langle d_{D_j}^i \rangle \\ x c_{e_j}^i &= \langle w_{E_j}^i, \omega_{E_j}^i \rangle \\ x c_{V_j}^i &= \langle d_{B_j}^i \rangle \\ x c_{T_j}^i &= \langle y c_{T_{0j}}^i \rangle \end{cases} \quad (12)$$

gdzie:

$y c_{T_{0j}}^i = w_{D_j}^i$ – zadana prędkość liniowa pojazdu obliczona przez robota a_0 zgodnie z (11)

$d_{B_j}^i$ – położenie środka zderzaka pojazdu a_j względem rogu pudła w chwili i

$d_{D_j}^i$ – pożądane położenie środka zderzaka pojazdu a_j względem rogu pudła w chwili $i+1$ – wartość zadana obliczona w kwancie czasu i na podstawie zależności (15) – wartość ta będzie wykorzystania w chwili $i+1$

Wartość $d_{B_j}^i$ określana jest za pomocą odpowiedniej funkcji agregującej (3) na podstawie odczytów czujników odległości zamontowanych na przednim zderzaku pojazdu.

Zachowanie robota pchającego zależy od polecenia uzyskanego od robota koordynującego: $y c_{T_{0j}}^i = w_{D_j}^i$ oraz położenia środka jego zderzaka względem rogu j pudła.

$$y c_{e_j}^{i+1} = f'_{c_{e_j}} \left(\langle w_{D_j}^i, \omega_{D_j}^i \rangle \right) \quad (13)$$

gdzie:

$$\omega_{D_j}^i = f''_{c_{e_j}} (d_{D_j}^i, d_{B_j}^i, w_{D_j}^i) \quad (14)$$

natomiast

$$d_{D_j}^i = \begin{cases} 0 & \text{dla } w_{D_j}^i = w_{max} \\ d_{Dm} & \text{dla } w_{D_j}^i < w_{max} \end{cases} \quad (15)$$

Wartość d_{Dm} dobrano eksperymentalnie. Dla robota a_L jest to odległość pomiędzy środkiem zderzaka, a punktem położonym pośrodku między ostatnim i przedostatnim czujnikiem odległości po lewej stronie pojazdu. Dla robota a_P odległość ta dotyczy pary czujników położonych z prawej strony pojazdu.

Funkcja $f''_{c_{e_j}}(\bullet)$ użyta w (13) ma dla pojazdu a_j następującą postać:

$$f''_{c_{e_j}} (d_{D_j}^i, d_{B_j}^i, w_{D_j}^i) = k \left(d_{D_j}^i - d_{B_j}^i \right) \quad (16)$$

Wartość współczynnika wzmocnienia k dobrano eksperymentalnie. Argument $w_{D_j}^i$ funkcji $f''_{c_{e_j}}(\bullet)$ jest potrzebny do wyznaczenia aktualnej wartości zadanej położenia $d_{D_j}^i$ – jest to robione na podstawie zależności (15).

Wszystkie trzy roboty działają w oparciu o poniższy prosty algorytm:

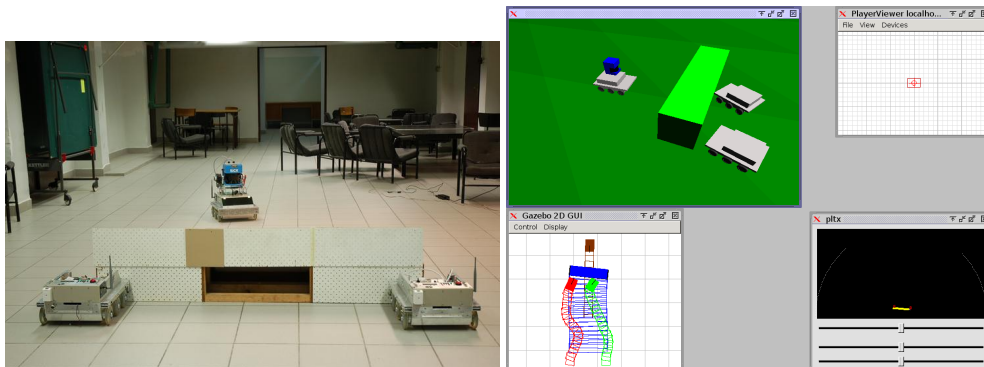
```

loop
  // Określenie aktualnego stanu robota
   $e_j \mapsto x_{e_j}^i; V_j \mapsto x_{V_j}^i; c_{T_j} \mapsto x_{c_{T_j}}^i;$ 
  // Określenie następnego stanu robota
   $y_{c_j}^{i+1} := f_{c_j}(x_{c_j}^i)$ 
  // Wysłanie rezultatów obliczeń
   $y_{e_j}^{i+1} \mapsto e_j; y_{V_j}^{i+1} \mapsto V_j; y_{c_{T_j}}^{i+1} \mapsto c_{T_j};$ 
   $i := i + 1;$ 
endloop
    
```

gdzie symbol “ \mapsto ” określa przekazanie informacji pomiędzy podsystemami, natomiast podwójny ukośnik poprzedza komentarz. Ponieważ algorytm (17) dotyczy wszystkich robotów, niektóre przesłania mogą nie być wykorzystywane w konkretnym przypadku.

4. EKSPERYMENTY

Zaproponowana metoda pchania została wstępnie zweryfikowana z wykorzystaniem symulatora trójwymiarowego *Gazebo* (rys. 3) – części projektu *Player/Stage*, gdzie do modelowania dynamiki obiektów wykorzystywana jest biblioteka *Open Dynamics Engine* [8, 9]. Zaletą takiego podejścia jest to, że symulator udostępnia ten sam interfejs programistyczny co rzeczywiste roboty (w ten sposób nie ma potrzeby duplikowania kodu dla obydwu tych przypadków).



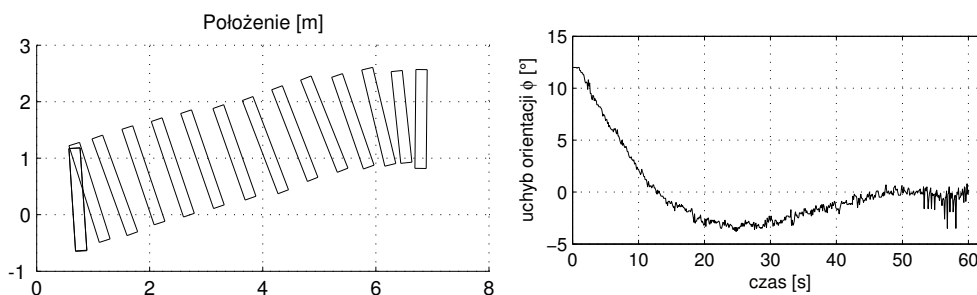
Rys. 3. Zadanie transportu pudła – widok rzeczywistego eksperymentu oraz obraz z symulatora

Pchanie pudła zostało zrealizowane przy wykorzystaniu nieholonomicznych robotów mobilnych typu „czołg” – gdzie napęd jest przenoszony za pomocą pasków

z dwóch silników na koła po lewej i prawej stronie. Taki układ może być z dobrym przybliżeniem traktowany jako układ różnicowy. Roboty są wyposażone m.in. w 6 czujników odległości o zakresie działania do 50 cm. Robot kontrolujący położenie obiektu jest wyposażony w dalmierz laserowy SICK model LMS200. Komunikacja między robotami zapewniona jest przez łącze WLAN.

W trakcie eksperymentów robot nadzorujący był nieruchomy i jedynie przekazywał robotom pchającym informację o aktualnej wartości kąta ϕ pomiędzy prostą prostopadłą do pudła, poprowadzoną przez jego środek, a półprostą poprowadzoną z punktu środkowego ściany pudła zwróconej w kierunku robota nadzorującego, a środkiem tego robota (rys. 2). Zadaniem robotów pchających była realizacja zleconej przez robota nadzorującego prędkości liniowej, a w konsekwencji minimalizacja kąta ϕ , oraz utrzymanie się na krawędzi pudła przez odpowiednie zmiany swej prędkości kątowej. Poniżej zamieszczono uzyskaną trajektorię transportowanego obiektu oraz wykres kąta ϕ traktowanego w tym przypadku jako uchyb regulacji (rys. 4). Konsekwencją zastosowania regulatora PI do likwidacji stałej różnicy kąta ϕ jest widoczne na wykresie przeregulowanie przebiegu.

Zadanie było kończone, gdy pudło znalazło się w pobliżu robota nadzorującego – wtedy dodatkowo zmieniano jego orientację tak, aby linie środkowe pudła i robota wiodące były się równoległe.



Rys. 4. Położenie pudła w trakcie pchania oraz wykres uchybu regulacji

5. WNIOSKI

W wyniku przeprowadzonych prac została zaproponowana metoda transportu obiektu przez roboty nieholonomiczne wykorzystująca warstwowy układ regulacji – zadaniem każdego z robotów jest utrzymywanie odpowiedniej orientacji względem obiektu, natomiast pożądaný ruch obiektu uzyskiwany jest dopiero przez odpowiednią koordynację prędkości robotów. Zaprezentowany algorytm został zweryfikowany zarówno w środowisku symulacyjnym jak i na rzeczywistych robotach. Bez utraty ogólności możliwe jest zastosowanie takiego podejścia do manipulacji bardziej złożonymi obiektami przez większe grupy robotów.

LITERATURA

- [1] R. C. Arkin. *Behavior-Based Robotics*. Cambridge, Mass., MIT Press 1998.
- [2] G. Dudek, M. Jenkin, E. Milius. A taxonomy of multirobot systems. In: *Robot teams: From diversity to polymorphism*, T. Balch and L. E. Parker, Eds. Welle-
sley, MA: AK Peters 2002.
- [3] O. Khatib. Mobile manipulation: The robotic assistant. *Robotics and Autono-
mous Systems*, February, 1999, Vol. 26, No. 2, s. 175–183.
- [4] C. R. Kube, E. Bonabeau. Cooperative transport by ants and robots. *Robotics
Autonomous Systems*, 2000, Vol. 30, No. 1–2, s. 85–101.
- [5] N. Miyata et al. Cooperative transport by multiple mobile robots in unknown sta-
tic environments associated with real-time task assignment. *IEEE Transactions
on Robotics and Automation*, October, 2002, Vol. 18, No. 5, s. 769–780.
- [6] L.E. Parker. Alliance: An architecture for fault tolerant multirobot cooperation.
IEEE Transactions on Robotics and Automation, April, 1998, Vol. 14, No. 2,
s. 220–240.
- [7] G. A. S. Pereira, M. F. M. Campos, V. Kumar. Decentralized algorithms for
multi-robot manipulation via caging. *International Journal Robotics Research*,
Jul. Aug., 2004, Vol. 23, No. 7–8, s. 783–795.
- [8] Russell Smith. Open dynamics engine v0.5 user guide. <http://www.ode.org/>,
February, 2006.
- [9] Richard T. Vaughan, Brian Gerkey, Andrew Howard. On device abstractions for
portable, reusable robot code. In: *Proceedings of the IEEE/RSJ International
Conference on Intelligent Robot Systems (IROS2003)*. *Proceedings*, October,
2003, s. 2121–2427.
- [10] Z. Wang, E. Nakano, T. Takahashi. Solving function distribution and behavior
design problem for cooperative object handling by multiple mobile robots. *IEEE
Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*,
September, 2003, Vol. 33, No. 5, s. 537–549.
- [11] S. Yamada, J. Saito. Adaptive action selection without explicit communication
for multirobot box-pushing. *IEEE Trans. on Systems, Man and Cybernetics-Part
C*, September, 2001, Vol. 31, No. 3, s. 398–404.
- [12] 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, Vol. 53, No. 1, s. 57–67.
- [13] C. Zieliński. Formalizacja opisu struktur ramowych do programowania syste-
mów wielorobotowych. In: *Przemysłowe i medyczne systemy robotyczne*. *Pro-
ceedings* Red. K. Tchoń. WKŁ, 2005, s. 53–66.

CONTROL OF A TEAM OF HETEROGENEOUS ROBOTS: EXAMPLE OF COOPERATIVE BOX PUSHING

The paper presents the implementation of the task in which two robots push a box under the supervision of a third robot. The pushing robots and the supervisory robot use different sensors, thus the system is heterogeneous. The task is described formally. Its execution has been verified both in simulation and on Electron non-holonomic robots.