

Krzysztof SKRZYPCZYK, Adam GALUSZKA
Politechnika Śląska, Instytut Automatyki

ALGORYTM EKSPLOKACJI PRZESTRZENI ROBOCZEJ PRZEZ GRUPĘ ROBOTÓW MOBILNYCH, BAZUJĄCY NA GRZE O SUMIE NIEZEROWEJ

Streszczenie. W prezentowanej pracy przedstawiono problem eksploracji przestrzeni roboczej przez grupę robotów mobilnych. Rozważane jest działanie grupy robotów mobilnych działających w sztucznie stworzonym środowisku o złożonej strukturze. Zaproponowano rozwiązanie problemu w ramach hybrydowej, częściowo zdecentralizowanej struktury systemu sterowania. System złożony jest z dwóch warstw: modułu planującego oraz modułu realizacji zadań elementarnych generowanych przez nadrzędną warstwę planowania, zrealizowany z godnie z behawioralną koncepcją sterowania. Rolą systemu planowania jest generowanie sekwencji zadań elementarnych, których wykonanie prowadzi do ukończenia zdefiniowanej misji. Zaprezentowano koncepcję modelowania rozważanego problemu za pomocą sekwencji statycznych N-osobowych gier w postaci normalnej. W pracy przedstawiono wyniki przeprowadzonych eksperymentów symulacyjnych z użyciem trzech robotów mobilnych.

MULTI-ROBOT WORKSPACE EXPLORATION ALGORITHM BASED ON NON-ZERO SUM GAME

Summary. In the paper we discuss a problem of multi robot exploration of the workspace. We consider a team of robots that performs a global task in a human-made workspace of complex structure. A hybrid architecture of the team motion control system is considered. The system can be split into two layers: the planner module and the behavior based collision free motion controller, that is designed to perform several elementary navigation tasks. The role of the planner is to plan and coordinate execution of elementary tasks by individual agents to obtain performance of global task. We presents the method of elementary tasks planning based on N-person game. Simulation of the algorithm was carried out, and its result is presented and discussed in the paper.

1. Wprowadzenie

Jednym z podstawowych wymagań narzucanych systemom sterowania robotów mobilnych jest ich autonomiczność. Przez to pojęcie rozumiemy takie rozwiązanie systemowe, zapewniające samodzielną pracę robota w otoczeniu o zmiennej dynamice. Aby sprostać takiemu zadaniu, system sterowania musi posiadać zdolność do rozwiązywania szeregu istotnych problemów. Dość wymienić tutaj konieczność działania w oparciu o niepewną informację, problem budowy modelu otoczenia jak również problem planowania ruchu. Na przestrzeni lat skryształizowało się kilka rozwiązań systemowych, które z mniejszym lub większym powodzeniem realizowały wyżej naszkicowane zadania. Rozwiązania te można podzielić na trzy grupy. Pierwsza z nich to, można rzec klasyczne, systemy planujące (ang. deliberative/planning systems), realizowane w ramach sekwencyjnej struktury. Kolejna grupa to systemy bazujące na nowatorskiej koncepcji sterowania reaktywnego czy też behawioralnego (ang. reactive, behavior based systems)[1][2][13][14]. Trzecia z omawianych grup to systemy hybrydowe (ang. Hybrid systems)[5][12]. To ostatnie z omawianych rozwiązań wydaje się być rozwiązaniem najefektywniejszym, łączącym w sobie zalety dwóch poprzednich grup systemów.

Działanie robota wewnątrz przestrzeni roboczej cechującej się złożoną strukturą (pomieszczenia biurowe, hale fabryczne itd.), bez jakiegokolwiek informacji o strukturze takiej przestrzeni, powoduje, iż zadania nawigacyjne mogą być realizowane w sposób nieefektywny bądź w najgorszym z przypadków mogą nie być realizowalne. Wprowadzenie apriori informacji o topologii przestrzeni roboczej w znakomity sposób upraszcza dekompozycję globalnych zadań nawigacyjnych realizowanych przez roboty [1].

W przypadku gdy rozważamy pracę grupy robotów, realizującej w ramach wspólnej przestrzeni roboczej, konieczne staje się rozważenie problemu koordynacji działań poszczególnych jednostek grupy. Nieprawidłowa metoda koordynacji może być przyczyną nieefektywnej realizacji misji grupy lub nawet doprowadzić do sytuacji, w której niemożliwe stanie się jej wykonanie. Dlatego problemowi temu poświęcono i poświęca się stale wiele uwagi, czego dowodem jest wiele publikacji traktujących o tym zagadnieniu. Dość wspomnieć choćby kilka z nich [4][6][7][9][11][13]. W cytowanych pracach znaleźć można kilka różnych metod koordynacji. Jednakże jedną z najbardziej wyrazistych jest ta bazująca na zastosowaniu teorii gier [3]. Dzieje się tak, gdyż problem koordynacji działań poszczególnych agentów-robotów może być postrzegany jako sytuacja konfliktowa,

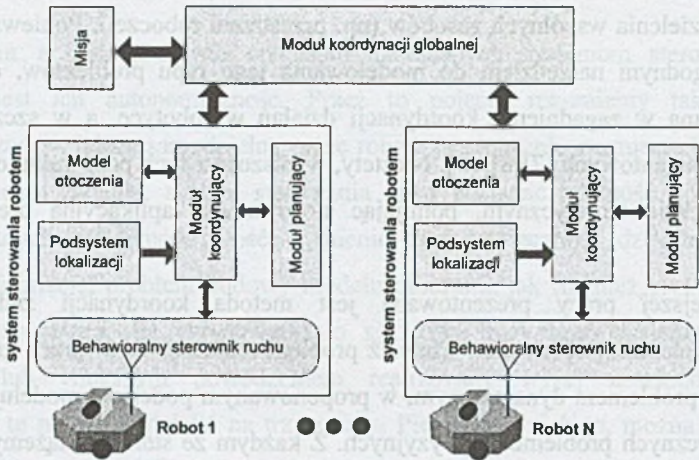
zachodząca pomiędzy nimi. Przyczyną konfliktu może być rozbieżność celów jak i konieczność dzielenia wspólnych zasobów (np. przestrzeni roboczej). Ponieważ teoria gier jest bardzo dogodnym narzędziem do modelowania tego typu problemów, dlatego jest chętnie stosowana w zagadnieniu koordynacji działań w robotyce, a w szczególności w systemach wieloagentowych [7][8][10]. Niestety, większość z tych prac traktuje ów problem jedynie w aspekcie teoretycznym, pomijając stosowalność aplikacyjną prezentowanych metod.

W niniejszej pracy prezentowana jest metoda koordynacji zadań agentów wykorzystująca elementy teorii gier. Chociaż problem realizacji misji przez grupę robotów jest w zasadzie problemem dynamicznym, w proponowanym podejściu modelujemy go jako sekwencje statycznych problemów decyzyjnych. Z każdym ze stanów wiążemy N-osobową grę dyskretną w postaci normalnej. Stosując na każdym z etapów koncepcję równowagi Nasha otrzymujemy rozwiązanie problemu cechujące się wysoką efektywnością. Opisywaną metodę wykorzystano do syntezy modułu planującego, wchodzącego w skład hybrydowego systemu sterowania. W pracy zaprezentowano wyniki symulacji potwierdzających efektywność proponowanej metody.

2. Opis systemu

Schemat ideowy systemu sterowania rozważanego w ramach niniejszej pracy przedstawiono na rys. 1.

W systemie można wyróżnić dwie zasadnicze warstwy sterowania. Pierwsza z nich, nazwana *warstwą realizacji zadań*, przeznaczona jest do implementacji na platformie mobilnej i jej rolą jest fizyczna realizacja elementarnych zadań nawigacyjnych, druga, zwana *warstwą koordynacji* służy do pośredniczenia w wymianie informacji pomiędzy poszczególnymi robotami. Z punktu widzenia tematyki niniejszej pracy ważniejsza jest pierwsza z omawianych warstw. Zrealizowana została ona zgodnie z hybrydową strukturą systemu sterowania i stanowi połączenie szybkiego, behawioralnego sterownika ruchu, którego zadaniem jest bezkolizyjna realizacja zadań nawigacyjnych oraz modułu planującego wyznaczającego ramowy plan wykonania realizowanej misji. Moduł planujący zrealizowany został na bazie N-osobowej gry niekooperacyjnej, za pomocą której modelowany jest proces interakcji pomiędzy poszczególnymi członkami grupy. Poszczególne komponenty tej warstwy systemu sterowania omówione zostaną w dalszych rozdziałach niniejszej pracy.



Rys.1. Schemat ideowy systemu sterowania

Fig.1. A diagram of the control system

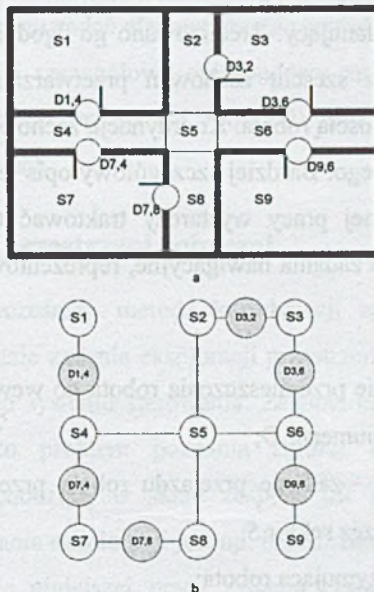
2.1. Model otoczenia

Jak już wspomniano, zakłada się, iż roboty dedykowane są do pracy w ramach sztucznie stworzonej przestrzeni roboczej o złożonej strukturze, której przykładem mogą być np. obiekty biurowe czy też hale fabryczne. Ponadto zakłada się, iż częściowa wiedza o tymże otoczeniu jest wprowadzona do systemu. Takie podejście ma na celu uproszczenie i uczynienie efektywniejszym proces nawigacji robotów w tymże otoczeniu. Rysunek 2a przedstawia plan przykładowego biura. Przestrzeń podzielona została na prostokątne obszary, zwane dalej *sektorami*. Każdy z sektorów reprezentuje powierzchnię zajmowaną przez pomieszczenia lub ich fragmenty. Przejścia pomiędzy tymi pomieszczeniami wyróżnione zostały jako *obiekty-przejścia*, najczęściej reprezentowane przez drzwi występujące w rozważanej przestrzeni roboczej. Model przestrzeni przechowywany jest i przetwarzany na dwóch poziomach: symbolicznym i geometrycznym. Pierwszy z nich ma postać grafu ważonego:

$$M = (V, W) \quad V = \{v_1, v_2, \dots, v_M\}, \quad W \subset V \times V \quad (1)$$

Wierzchołki grafu V reprezentują poszczególne obiekty wyróżnione w ramach przestrzeni roboczej W : sektory V_S oraz przejścia V_D gdzie $V = V_S \cup V_D$. Ponadto każdy z każdym z

obiektów skojarzono współczynniki opisujące w sposób bardzo przybliżony strukturę otoczenia. I tak, sektory reprezentowane są przez tzw. współczynnik zabudowy c_i , który w pewnym stopniu odzwierciedla w sposób ilościowy gęstość zabudowy określonego sektora (ilość różnych obiektów, mebli itp. znajdujących się wewnątrz sektora). Obiekty typu przejście opisywane są przez współczynnik o_j , określający stopień pewności faktu, iż dane przejście jest przejezdne (znajduje się w stanie otwarcia). Na poziomie geometrycznym i -ty sektor opisywany jest przez trójkę punktów, których pierwsze dwa: (x'_i, y'_i) i (x''_i, y''_i) definiują prostokątną ramkę sektora, natomiast trzeci z nich (x^*_i, y^*_i) jest pewnym *punktem charakterystycznym* zlokalizowanym wewnątrz prostokątnej ramki, miejscu, które z dużym prawdopodobieństwem nie jest zajęte przez jakiegokolwiek obiekt. Podobnie *obiekt-przejście*, na poziomie geometrycznym opisany jest przez okrąg o środku w punkcie (x^*_j, y^*_j) i promieniu r_j . Krawędzie grafu opisują relacje incydencji zachodzące pomiędzy poszczególnymi obiektami.



Rys.2. Plan przykładowej przestrzeni roboczej (a) oraz jej model topologiczny (b)

Fig.2. An exemplary workspace layout (a) and its topological model (b)

Współczynniki wagowe w_{ij} grafu opisują szacunkowe koszty tranzycji pomiędzy poszczególnymi obiektami, które są proporcjonalne do dystansu L_{ij} pomiędzy nimi

mierzonego wg normy euklidesowej. W niniejszej pracy zdefiniowano je następującymi zależnościami empirycznymi:

$$w_{ij} = \begin{cases} L_{i,j} (1+c_i) (1+e^{-\alpha O_i}) & \text{jeżeli } v_i \in V_S \wedge v_j \in V_D \\ L_{i,j} \frac{1}{O_i} (1+c_j) & \text{jeżeli } v_i \in V_D \wedge v_j \in V_S \\ L_{i,j} [1+0.5(c_i+c_j)] & \text{jeżeli } v_i \in V_S \wedge v_j \in V_S \end{cases} \quad (2)$$

Warto podkreślić raz jeszcze, iż model ten opisuje tylko niezmiennie cechy przestrzeni roboczej oraz daje szacunkową informację o zmianach struktury tejże przestrzeni. Model przestrzeni wewnątrz sektorów jest nieznaną, a co więcej- może ulegać dynamicznym zmianom.

2.2. Behawioralny sterownik ruchu

Zadaniem tej części systemu jest fizyczna realizacja elementarnych zadań nawigacyjnych generowanych przez moduł planujący. Zrealizowano go zgodnie z behawioralną koncepcją sterowania [2]. Składa się z sześciu zachowań przetwarzających dane sensoryczne w odpowiednie sterowania prędkością robota. Koordynacja zachowań zrealizowana jest zgodnie z zasadą arbitrażu priorytetowego. Bardziej szczegółowy opis tego modułu przedstawiono w [14]. Dla celów prezentowanej pracy wystarczy traktować ten moduł jako komponent realizujący cztery elementarne zadania nawigacyjne, reprezentowane dalej przez następujące operatory:

- **FindDoor (D)** – zadanie przemieszczenia robota do wewnątrz obszaru zajmowanego przez obiekt-przeście o numerze D ;
- **TraverseDoor (D, S)** – zadanie przejazdu robota przez przeście D do wewnątrz obszaru zajmowanego przez sektor S ;
- **Wait ()** – komenda zatrzymująca robota;
- **GoTo (S1, S2)** zadanie przemieszczenia robota z sektora $S1$ do sektora $S2$.

Wszystkie z powyższych operatorów reprezentują zadania bezkolizyjnego przemieszczenia się robota z bieżącej lokalizacji do lokalizacji docelowej zdeterminowanej przez stan obiektów reprezentowanych przez argumenty operatorów. Różnice pomiędzy

poszczególnymi operatorami, widoczne z perspektywy sterownika ruchu, leżą w parametrach przekazywanych wraz z poszczególnymi operatorami, takich jak np. dopuszczalna prędkość robota czy też wartości progowe odczytów z sensorów.

2.3. Moduł planujący

Analizując działanie tego modułu z perspektywy niniejszej pracy, można powiedzieć, iż jest on najważniejszym elementem systemu. W jego skład wchodzi trzy komponenty: moduł koordynacji, moduł budowy modelu procesu oraz blok wyznaczania rozwiązania problemu. Działanie modułu planującego można w skrócie opisać w sposób następujący. *Moduł koordynacji* otrzymuje dane o położeniu wszystkich robotów wchodzących w skład grupy. Ponadto dostarczane są dane o aktualnym stanie otoczenia. W zależności od rodzaju zadania, położenia robotów oraz aktualnego stanu realizacji zadania budowany jest model procesu decyzyjnego zachodzącego pomiędzy agentami-robotami. Model ten w zasadzie ma postać zbioru funkcji kosztów skojarzonych z każdym z robotów, których wartość zależy od akcji przez nie podejmowanych. Następnie wyznaczane jest rozwiązanie problemu. Rozwiązanie ma postać zestawu zadań elementarnych prowadzących do optymalnej realizacji zadania globalnego. Bardziej szczegółowy opis budowy modelu procesu decyzyjnego (na przykładzie zadania eksploracji przestrzeni roboczej) oraz metod rozwiązania pokazane zostaną w kolejnych rozdziałach.

3. Zadanie eksploracji przestrzeni roboczej

Jak wspomniano wcześniej, metoda koordynacji zadań wielu robotów zostanie zaprezentowana na przykładzie zadania eksploracji przestrzeni roboczej, której topologiczny model jest wprowadzony do systemu sterowania. Zadanie eksploracji może być w sposób ogólny sformułowane jako problem poznania całości lub części przestrzeni przez poszczególne jednostki wchodzące w skład zespołu jak najniższym kosztem. Jedną z możliwych interpretacji zadania eksploracji jest np. dostarczanie części do gniazd obróbczych w dużej fabryce. W sensie niniejszej pracy zadanie eksploracji jest zdefiniowane jako odwiedzenie wszystkich lub części obiektów $M_v \subset V_s$ wchodzących w skład przestrzeni roboczej M w jak najmniejszej liczbie kroków. Proces eksploracji modelowany jest w pracy jako sekwencja, statycznych, jednoetapowych gier w postaci normalnej.

3.1. Modelowanie problemu eksploracji

Wprowadźmy najpierw notację, która będzie stosowana w dalszej części pracy. Oznaczmy stan grupy robotów jako zbiór:

$$X = \{x_i\} \quad i = 1, 2, \dots, N, \quad x_i \in V, \quad (3)$$

co jest równoznaczne z faktem, iż i -ty robot znajduje się wewnątrz obiektu przestrzeni roboczej opisanego wierzchołkiem x_i . Zbiór wszystkich możliwych akcji i -tego agenta-robota, nazywany dalej zbiorem decyzyjnym, wynikający z bieżącego stanu x_i , oznaczono przez:

$$A_i(x_i) = \{a_1, a_2, \dots, a_k\} \quad (4)$$

Poszczególne akcje odpowiadają zadaniom modelowanym za pomocą operatorów, natomiast ich zbiór, możliwy do realizacji w danym stanie zależy od struktury przestrzeni roboczej. W naszym przypadku zbiór decyzyjny i -tego robota określany jest na bazie warunków stosowalności operatorów (ang. precondition list):

FindDoor (D)

preconditions = $\{x_i \in V_S, w_{x_i, D} \neq \infty\}$

TraverseDoor (D, S)

preconditions = $\{x_i = D \in V_D, w_{D, x_i} \neq \infty\}$

Wait ()

preconditions = \emptyset

GoTo (S1, S2)

preconditions = $\{x_i = S1 \in V_S, w_{S1, S2} \neq \infty\}$

Z punktu widzenia procesu decyzyjnego akcja $a_k \subset A_i$ jest postrzegana jako operacja, której rezultatem jest zmiana bieżącego stanu x_k^n agenta w stan x_k^{n+1} :

$$a_k : x_i^n \rightarrow x_i^{n+1} \quad x_i^n \in X \quad x_i^{n+1} \in V \quad a_i \in A_i \quad (5)$$

Używając notacji przedstawionej powyżej można sformułować problem eksploracji jako odwiedzenie wszystkich obiektów należących do przestrzeni roboczej, zdefiniowanych zbiorem $M_G \subset V_S$. Wprowadźmy dodatkowy zbiór $M_V \subset M_G$ określający obiekty, które już zostały odwiedzone w trakcie realizacji zadania. I tak, zgodnie z powyższym, można

zdefiniować cel zadania jako $M_V = M_G$. Zadaniem algorytmu realizowanego przez moduł planujący jest wybór w każdym ze stanów procesu jednej z możliwych akcji, która spowoduje realizację fragmentu globalnego zadania w jak najefektywniejszy sposób, przy wzięciu pod uwagę możliwych akcji realizowanych przez resztę jednostek wchodzących w skład grupy.

W niniejszej pracy problem wyboru akcji postrzegany jest jako gra pomiędzy poszczególnymi agentami-robotami. Wynik gry jest pewną wartością liczbową, zwaną dalej kosztem, ponoszonym przez graczy w wyniku podjęcia określonych decyzji, który podlega minimalizacji. Ponadto zadanie eksploracji ma cechy, które sprawiają, iż może być postrzegane jako specyficzna odmiana gry, jaką jest tzw. problem grup decyzyjnych. Cechą szczególną takiego problemu jest to, iż optymalizacji podlega tylko jeden wskaźnik kosztów. Chociaż proces eksploracji jest w ogólności dynamiczny, w pracy jest on przybliżany za pomocą sekwencji statycznych, gier N -osobowych w postaci normalnej. Aby otrzymać model procesu (gry), musimy wyznaczyć postać minimalizowanej funkcji kosztów skojarzonej z każdym ze stanów, której wartość zależy od stanu realizacji procesu i określonych akcji podejmowanych przez poszczególnych agentów.

Zaproponowano następującą postać funkcji kosztów, będącą złożeniem trzech komponentów:

$$I(a_1, \dots, a_i, \dots, a_N) = -I_R + I_E + I_D \quad (6)$$

Pierwszy z nich związany jest z „nagrodą” jaką otrzymuje grupa w wyniku podjęcia określonych decyzji, której wartość zależy od rozmiaru nowopoznanej przestrzeni. Definiujemy ją jako sumę:

$$I_R(a_1, \dots, a_i, \dots, a_N) = \sum_{i=1}^M R_i(a_i) \quad (7)$$

$$R_i = \begin{cases} \frac{1}{k} R > 0 & \text{jeżeli } x_i^{n+1} \in M_G \wedge x_i^{n+1} \notin M_V, \\ 0 & \text{w przeciwnym wypadku} \end{cases}$$

gdzie R oznacza pewną dodatnią wartość liczbową symbolizującą wspomnianą „nagrodę”. Czynniki k oznaczają liczbę robotów, które w wyniku podjęcia swoich akcji a_i $i=1,2,\dots,N$, znalazły się wewnątrz tego samego obiektu przestrzeni. Ponieważ oczekiwana wartość nagrody ulega maksymalizacji, natomiast cała funkcja kosztów minimalizacji, dlatego wartość czynnika brana jest ze znakiem minus.

Drugi ze składników funkcji (6) związany jest z kosztem (np. wydatkiem energetycznym) ponoszonym przez grupę w wyniku realizacji określonych akcji, który jest proporcjonalny do kosztów tranzycji pomiędzy poszczególnymi obiektami przestrzeni roboczej zdefiniowanymi przez wagi grafu M :

$$I_E(a_1, \dots, a_i, \dots, a_N) = \sum_{i=1}^N w(x_i^n, x_i^{n+1}), \quad (8)$$

gdzie x_i^n i x_i^{n+1} oznaczają odpowiednio obiekt opisany przez wierzchołek grafu, w którym znajduje się aktualnie i -ty robot oraz obiekt, w którym znajdzie się w wyniku realizacji akcji a_i . Trzeci z czynników wchodzących w skład funkcji kosztów związany jest z kosztem przemieszczenia się robota do najbliższego, niepoznanego obiektu. Zdefiniujemy w pierwszej kolejności ścieżkę o najniższym koszcie w grafie M pomiędzy wierzchołkami n i m jako:

$$P_{\min}(n, m) = \{v_n, \dots, v_k, \dots, v_m\} \subset V \quad (9)$$

oraz niech zbiór nieodwiedzonych obiektów będzie dany jako $M_U = \{u_i\} = M_G \setminus M_V$. Wówczas koszt I_D jest obliczany z następującej zależności:

$$I_D(a_1, \dots, a_i, \dots, a_N) = \sum_{i=1}^N D_{\min,i} \quad (10)$$

$$D_{\min,i} = \min [P_{\min}(x_i^{n+1}, u_i)], \quad (11)$$

gdzie $D_{\min,i}$ stanowi koszt przemieszczenia i -tego robota ze stanu x_i^{n+1} do najbliższego z niepoznanych obiektów określonych przez zbiór M_U .

3.2. Rozwiązanie problemu

W poprzednim rozdziale wprowadzono model przybliżający pojedynczy etap procesu eksploracji, dany jako funkcja kosztów statycznej gry N -osobowej zachodzącej pomiędzy agentami-robotami. W niniejszej sekcji poszukujemy rozwiązania powyższego problemu. Poszukiwanym rozwiązaniem będzie zbiór akcji poszczególnych robotów $S^* = \{a_{10}, a_{20}, \dots, a_{N0}\}$, których wykonanie spowoduje realizację fragmentu globalnego zadania w sposób jak najkorzystniejszy (w sensie ponoszonych kosztów) dla wszystkich agentów. Problem eksploracji ma charakter problemu kooperacyjnego, co jest zawarte w pojedynczej funkcji kosztów. Z drugiej strony fakt, iż poszczególni agenci nie komunikują się

w trakcie realizacji poszczególnych akcji, sprawia, że problem posiada również cechy problemu niekooperacyjnego. Dlatego poszukujemy rozwiązania problemu właśnie jako zagadnienia niekooperacyjnego. Klasyczne rozwiązanie tego typu problemów bazuje na koncepcji równowagi Nasha [3], która w przypadku naszego zagadnienia redukuje się do poszukiwania minimum funkcji kosztów I po poszczególnych akcjach. Stąd poszukiwane rozwiązanie ma postać:

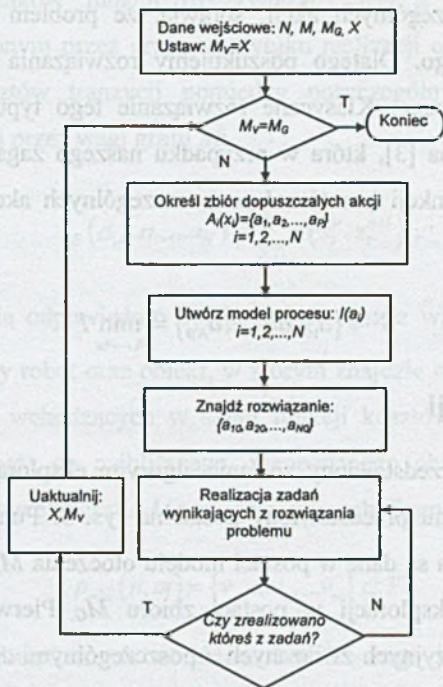
$$S^* = \{a_{10}, a_{20}, \dots, a_{N0}\} = \min_{a_1, \dots, a_N} I \quad (12)$$

4. Algorytm eksploracji

W tym rozdziale przedstawiony zostanie algorytm eksploracji przestrzeni roboczej. Schemat blokowy algorytmu przedstawiony został na rys. 3. Punktem wejścia algorytmu realizowanego przez system są dane w postaci modelu otoczenia M , stanu grupy robotów X , zdefiniowanego zadania eksploracji w postaci zbioru M_G . Pierwszy krok algorytmu to wyznaczenie zbiorów decyzyjnych związanych z poszczególnymi agentami, wynikających z bieżącego stanu X . Następnie budowany jest model procesu w postaci funkcji kosztów I zgodnie z równaniami (6-10). W dalszej kolejności poszukiwane jest rozwiązanie problemu zgodnie z (12). Zgodnie ze znalezionym rozwiązaniem do sterownika ruchu przesyłany jest kod akcji, gdzie rozpoczyna się jej fizyczna realizacja. System czeka na komunikat ze sterownika i w przypadku jego otrzymania przesyła informację do modułu koordynacji globalnej (zadanie zakończone lub nie możliwe wykonanie zadania). Po otrzymaniu komunikatu następuje uaktualnienie modelu otoczenia, stanu grupy oraz stanu realizacji zadania. Następnie algorytm przechodzi do początku. Proces ten jest powtarzany tak długo, dopóki nie nastąpi spełnienie warunku $M_G = M_v$ determinującego zakończenie zadania.

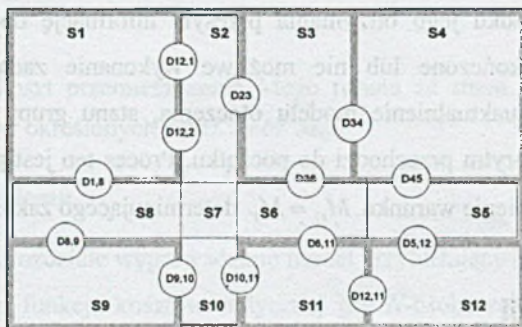
5. Wyniki symulacji

W celu weryfikacji proponowanej metody przeprowadzono eksperymenty symulacyjne z wykorzystaniem środowiska symulacyjnego M.A.S.S.[15]. Aplikacja pozwala na tworzenie modelu przestrzeni roboczej oraz symuluje kinematykę oraz układ sensoryczny robota laboratoryjnego Khepera™. Strukturę przykładowej przestrzeni roboczej zamodelowanej w symulatorze przedstawiono na rys. 4.



Rys.3. Schemat blokowy algorytmu eksploracji

Fig.3. A diagram of the algorithm of exploration



Rys. 4. Układ i struktura przestrzeni roboczej użytej do symulacji

Fig. 4. The layout of the workspace used for a simulation

Model przestrzeni roboczej składa się z 12 obiektów typu sektor $V_S = \{v_1, v_2, \dots, v_{12}\}$, które odpowiadają pomieszczeniom i fragmentom korytarzy znajdujących się wewnątrz przestrzeni. Ponadto wyróżniono w nim 13 obiektów definiujących przejścia pomiędzy

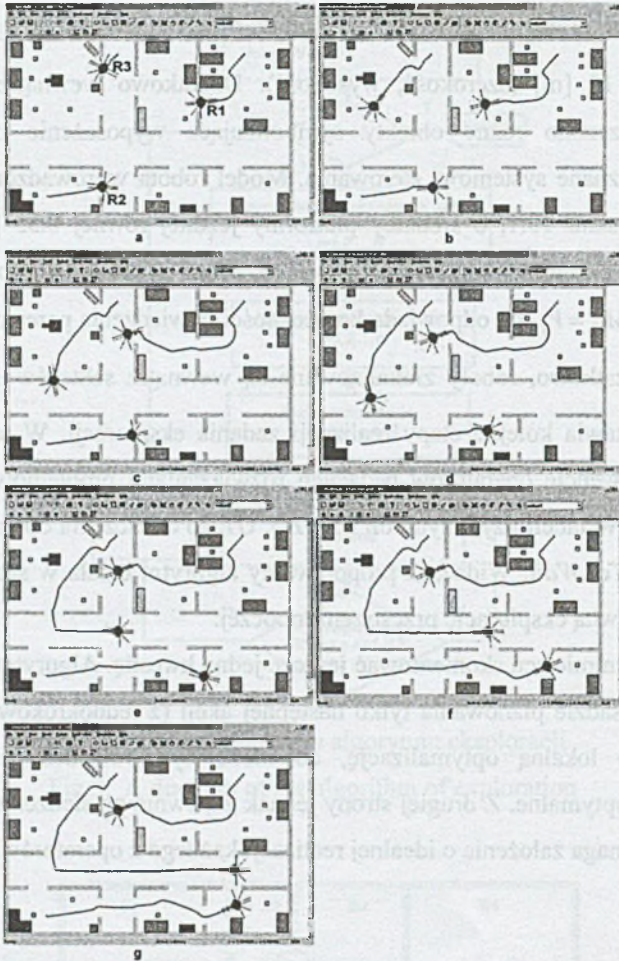
poszczególnymi sektorami $V_d = \{v_1, \dots, v_{25}\}$. Obszar przestrzeni roboczej to prostokąt o wymiarach 15×10 [m] (szerokość, wysokość). Dodatkowo wewnątrz poszczególnych sektorów rozmieszczono różne obiekty symbolizujące wyposażenie biurowe, których położenie nie jest znane systemowi sterowania. Model robota wprowadzony do systemu to robot *Khepera* w skali 10:1, o średnicy platformy jezdnej równej 0.55 [m]. Rozważamy realizację zadania eksploracji przestrzeni roboczej przez grupę $N=3$ robotów. Zatem cel definiujemy jako $M_G = V_S$, co odpowiada konieczności odwiedzenia przez roboty wszystkich 12 sektorów. Początkowo, roboty zlokalizowane są wewnątrz sektorów $X_{ini} = \{v_4, v_9, v_2\}$. Rysunek 5 przedstawia kolejne etapy realizacji zadania eksploracji. W tabeli 1 natomiast przedstawiono sekwencje operatorów będących rozwiązaniami problemów decyzyjnych na każdym z etapów. W tabeli użyto symboli *FD*, *TD*, *GT* do oznaczenia operatorów *FindDoor*, *TraverseDoor*, *GoTo*, *Wait*. Widać, iż proponowany algorytm działa w sposób prawidłowy, zapewniając efektywną eksplorację przestrzeni roboczej.

Warto w tym miejscu skomentować jeszcze jedną kwestię. Algorytm prezentowany w pracy działa na zasadzie planowania tylko następnej akcji (z jednokrokovym horyzontem), zapewniając tylko lokalną optymalizację, co może być powodem, iż nie wszystkie rozwiązania będą optymalne. Z drugiej strony jednak zapewnia on śledzenie zmian dynamiki otoczenia i nie wymaga założenia o idealnej realizacji każdego z operatorów.

Tabela 1

Sekwencje operatorów na każdym etapie realizacji zadania eksploracji

n	Robot 1	Robot 2	Robot 3
1	FD ('D3, 4')	FD ('D9, 10')	FD ('D1, 2 1')
2	TD (D3, 4, S3)	TD (D9, 10, S10)	TD (D1, 2 1, S1)
3	FD (D2, 3)	FD (D10, 11)	FD (D1, 8)
4	TD (D2, 3, S2)	TD (D10, 11, S11)	TD (D1, 8, S8)
5	W ()	FD (D11, 12)	GT (S7)
6	W ()	TD (D10, 11, S12)	GT (S6)
7	W ()	W ()	GT (S5)



Rys. 5. Wynik przykładowej symulacji

Fig. 5. The result of the exemplary simulation

6. Podsumowanie

W pracy zaprezentowano problem planowania i koordynacji zadań w środowisku wielu robotów. Rozważano pracę grupy robotów, której celem jest eksploracja przestrzeni roboczej o złożonej strukturze. Zaproponowano użycie hybrydowej struktury systemu sterowania oraz metodę modelowania problemu koordynacji działań robotów. Ponadto zaprezentowano algorytm eksploracji przestrzeni roboczej, zgodnie z którym system realizuje zdefiniowane zadanie. Kluczowym elementem algorytmu jest proces budowy modelu zadania

eksploracji w postaci N -osobowej gry w postaci normalnej, do której rozwiązania użyto koncepcji równowagi Nasha. Chociaż w pracy zaprezentowano wyniki pojedynczego eksperymentu symulacyjnego, to faktycznie wykonano wiele innych symulacji, w różnych konfiguracjach przestrzeni roboczej stosując różne zestawy parametrów modelu. Na tej podstawie można stwierdzić, iż algorytm działa w sposób zadowalający, generując w większości przypadków optymalne rozwiązania problemu, nawet w otoczeniu o bardzo złożonej strukturze. Jednakże, jak już wspomniano, nie można zagwarantować uzyskania optymalnych wyników we wszystkich przypadkach. Powodem tego jest jednokrokowa metoda planowania działań. Takie podejście ma jednak ważną zaletę – pozwala śledzić zmiany otoczenia i nie wymaga założenia o idealnej realizacji operatorów. Przedstawione podejście wydaje się być obiecujące, dlatego przyszłe prace skoncentrowane będą na zastosowaniu metody na stanowisku eksperymentalnym złożonym z dwóch rzeczywistych robotów *Khepera*.

Praca finansowana z grantu MNiI nr 3T11 029 28/2006.

LITERATURA

1. Althaus, P., Christensen H.I.: Behaviour coordination in structured environments. *Advanced Robotics*, 17(7), 2 657–674
2. Arkin, R.C. : *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998
3. Basar, M., Olsder, G.J.: *Dynamic Noncooperative Game Theory*. Mathematics in Science and Engineering, Academic Press Inc. Ltd, London, 1982
4. Burgard, W. et al. 2000. Collaborative Multi-Robot Exploration. Proc. IEEE International Conference on Robotics and Automation (ICRA), San Francisco CA
5. Fox, D.: Burgard W. and Thrun S.: A Hybrid Collision Avoidance Method For Mobile Robots. Proc. of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998
6. Gerkey, B. and M.J. Mataric.: Pusher-watcher: An Approach to Fault-Tolerant Tightly-Coupled Robot Coordination, Proc. of the IEEE International Conference on Robotics and Automation (ICRA2002), Washington DC, 2002

7. Golfarelli, M. and Meuleu N.: Multi Agent Coordination in Partially Observable environments, Proceedings of the 2000 IEEE Int. Conf. of Robot and Automation, San Francisco, 2000, 2777- 2782
8. LaValle, S.: Robot motion planning: A game-theoretic foundation. *Algorithmica*. 26(3), 2000, 430–65
9. Lawton, J.R. et al.: "A Decentralized Approach to Formation Maneuvers." *IEEE Transactions on Robotics and Automation*, Vol 19, No 6, 2003, 933–941
10. Li, Q. and S. Payandeh.: On Coordination of Robotics Agents Based on Game Theory, ICRA 2001, Seoul, Korea, 2001, (private communication)
11. Sequiera, J. and M. I. Ribeiro.: "A Negotiation Model for Cooperation Among Robots", Proc. of European Control Conference, Porto, Portugal, 2001
12. Shim, H.S. et al.: "A hybrid control structure for vision based soccer robot system." *Int. J. Intelligent Automation and Soft Computing*, 6: (1), 2000, 89–101.
13. Schneider-Fontan, M. and Mataric, M.J.: Territorial Multi-Robot Task Division, *IEEE Transactions on Robotics and Automation*, Vol 15, N.5, 1998, 815-822
14. Skrzypczyk, K.: Behavior-based approach to a synthesis of a mobile vehicle control system." IV-th Domestical PhD Workshop OWD, Istebna-Zaolzie, Poland, 2002
15. Skrzypczyk, K.: Hybrydowy system sterowania robotem mobilnym. Rozprawa doktorska, Gliwice 2005

Recenzent: Prof. dr hab. inż. Zbigniew Banaszak

Abstract

In this paper we discussed a problem of planning and coordination of tasks in a multi robot system. We considered a team of robots that was intended to perform a task of exploration of a human-made workspace of complex structure. We proposed both the hybrid architecture of a control system and method of coordination of multiple robotic agents. In the paper we also presented an algorithm of exploration of workspace. The core of the algorithm is the model of the process that is stated as a non-cooperative game in a normal form. We applied the Nash equilibrium concept to generate a solution of the problem. Although the result of only one simulation was presented, we had made a number of simulation

experiments using both various parameters and workspace configurations. In all cases we obtained correct task execution. On the basis of simulations we carried out we can conclude that this algorithm works well and provides effective exploration of even very complex-structured environments. However, the algorithm can not guarantee optimal task performance. It is caused the algorithm uses only one-step-ahead planning method. But this approach on the other hand has other advantage - it allows to track dynamical changes of the environment and it does not need the assumption that a given action is always executed in a perfect way.

OPTIMALNE ROZMIERZCZENIE CZĘSTOTLIWÓW PARAMETRYCZNYCH DLA LINIOWEGO UKŁADU O PARAMETRACH ROZKŁADANYCH

Streszczenie: W pracy przedstawiany został problem optymalnego rozmieszczenia czynnika promiennych dla liniowego układu o parametrach rozłożonych wzdłuż równoległym rozmiarowym ciągłym typu parabolicznego ze zmienną wykładniczą wznoszący oraz zmiennymi równymi. Rozwiązany jest problem optymalnego przybliżenia wzmocnienia.

THE OPTIMUM EXPERIMENTAL DESIGN PROBLEM FOR LINEAR DISTRIBUTED PARAMETER SYSTEMS

Summary: The optimum experimental design problem for linear distributed parameter system described by linear parabolic equations with exponential increasing gain and constant coefficients is considered. The approach, based on approximation of the gain, allows how to construct and determine the initial state.

1. Wstęp

Procesy opisane liniowymi parabolicznymi równaniami różniczkowymi dwukrotnym z różnego typu warunkami brzegowymi są charakterystycznym przykładem nie tylko od czasu, ale również od przestrzeni przestrzennych - bardziej właściwym określeniem z punktu widzenia teorii sterowania jest więc procesy takie jak systemy rozłożone, które dysponując jedynie określonym czasem obserwacji pomiarowych nie można sterować tym systemem w dowolnym punkcie przestrzeni. W tym przypadku koniecznym jest określenie optymalnego układu systemu o parametrach rozłożonych na podstawie wyników obserwacji z odpowiednio rozmieszczonych czynnika promiennych. W tym artykule problem optymalnego rozmieszczenia czynnika promiennych jest sformułowany w sposób ogólny.