ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2006
Seria: AUTOMATYKA z. 146 Nr kol. 1739

Krzysztof SKRZYPCZYK, Adam GAŁUSZKA, Andrzej ŚWIERNIAK
Politechnika Śląska, Instytut Automatyki

METODA BEZKOLIZYJEGO STEROWANIA GRUPĄ ROBOTÓW, BAZUJĄCA NA PUNKTACH RÓWNOWAGI GIER NIEKOOPERACYJNYCH

Streszczenie. W pracy przedstawiono zastosowanie koncepcji punktów równowagi w grach niekooperacyjnych do bezkolizyjnego planowania ruchu grupy robotów mobilnych w dynamicznym, szybkozmiennym środowisku. Zaproponowano prawo sterowania wykorzystujące rozwiązanie niekooperacyjnego problemu decyzyjnego, umożliwiające syntezę systemu bezkolizyjnego sterowania grupą robotów. Rozwiązania problemu planowania ruchu, wykorzystujące koncepcję równowagi są dobrze znane w literaturze. Takie podejście jest wysoce efektywne, jednak tylko w przypadku gdy istnieją pojedyncze punkty równowagi. Jednakże nawet w prostych problemach nawigacyjnych pojawiają się sytuacje, w których występują rozwiązania wielokrotne, będące powodem istotnych trudności w syntezie praw sterowania. W pracy zaproponowano rozwiązanie problemu wyboru pojedynczego rozwiązania, bazujące na idei częściowej koordynacji. Do systemu wprowadzono moduł koordynatora, który staje się aktywny w sytuacjach gdy zostaje wykryte występowanie wielokrotnych punktów równowagi. W celu potwierdzenia efektywności proponowanej metody przedstawiono wyniki symulacji przeprowadzonych dla 2 i 5 robotów.

CONTROL OF A TEAM OF MOBILE ROBOTS BASED ON NON-COOPERATIVE EQUILIBRIA

Summary. In this work we present an application of the concept of non-cooperative game equilibria for the design of a collision free movement of a team of mobile robots in a dynamic environment. We propose the solution to the problem of feasible control synthesis, based on partially centralized sensory system. The control strategy based on the concept of non cooperative game equilibria is well known in the literature. It is highly efficient through phases where the solution is unique. However, even in simple navigation problems, it happens that multiple equilibria occur, which poses a problem for control synthesis and may lead to erroneous results. In this paper we present the solution to this problem based on the partial centralization idea. The

coordinator module is incorporated into the system and becomes active when multiple equilibria are detected. The coordination method includes a "fair arbiter" for selection of an appropriate equilibrium solution. Simulation studies of the proposed methodology were carried out for 2,3 and 5 robots and their results are presented.

1. Wprowadzenie

Przeglądając literaturę dotyczącą robotyki można zauważyć, iż dużą uwagę poświęca się problemowi planowania ruchu w środowisku wielu robotów. Występowanie wzajemnych interakcji pomiędzy robotami sprawia, że problem jest trudny do rozwiązania. Prostym dowodem na to jest fakt, iż równoczesne zaplanowanie bezkolizyjnych trajektorii dla N robotów nawet w przestrzeni wolnej od przeszkód nie jest zagadnieniem trywialnym (jak to ma miejsce w przypadku problemu dotyczącego pojedynczego robota). Jednym z narzędzi dosyć powszechnie i chętnie stosowanym do modelowania i rozwiązywania problemów interakcji pomiędzy wieloma jednostkami jest teoria gier. W kontekście jej zastosowań do problemu planowania ruchu robotów można wyszczególnić wiele prac. W [6] pokazano rozwiązanie problemu generowania bezkolizyjnych trajektorii dla dwóch robotów, bazujące na wykorzystaniu 2-osobowej gry powtarzalnej oraz kryterium „maxi-max „ do znajdowania rozwiązań. Z kolei praca [13] pokazuje sposób modelowania i rozwiązania problemu klasy ang. clean-up-collection dla grupy 2 robotów, oparty na 2-osobowej grze o sumie zerowej. Kontynuując przegląd prac poświęconych wykorzystaniu teorii gier do planowania w świecie robotów nie sposób tu pominąć szeregu prac [10][12][11]. Pierwsze dwie z wymienionych pokazują zagadnienie analizy i wyboru czasooptymalnych strategii nawigacyjnych dla grupy N robotów z użyciem równowagi Nasha. W pracy [11] pokazano z kolei algorytm wyznaczający skończony zbiór pareto-optymalnych strategii koordynacyjnych dla dwóch robotów o geometrii wieloboku poruszających się na płaszczyźnie.

Oczywiście teoria gier nie jest jedyną znaną i stosowaną metodą koordynacji i planowania w świecie wielu robotów. Warto tutaj przynajmniej wymienić dwie prace. Pierwsza z nich [2] poświęcona jest zagadnieniu generowania optymalnych, gładkich trajektorii dla grupy robotów na bazie podejścia geometrycznego. Drugą interesującą pozycją jest [5], w której autorzy prezentują metodę dynamicznej alokacji zadań dla grupy robotów mobilnych, bazującą na metodach aukcyjnych. Wykorzystując to podejście otrzymano i poddano analizie różne typy kooperacyjnych zachowań grupy.

Biorąc pod uwagę podejście bazujące na wykorzystaniu teorii gier można powiedzieć, iż przytoczone pozycje dają reprezentatywny przegląd zagadnień związanych z problemem sterowania grupą robotów. Pokazują również korzyści płynące z zastosowania omawianego podejścia. Jednakże w prezentowanych pracach można zaobserwować brak analizy bardzo ważnego aspektu tego podejścia, a mianowicie istnienia wielokrotnych rozwiązań problemów modelowanych za pomocą gier. Istnienie wielokrotnych rozwiązań jest natomiast typowe dla problemów planowania nawet w prostych przypadkach. Symulacje pokazały, iż wybór odpowiedniego, pojedynczego rozwiązania ma kluczowe znaczenie dla jakości całego procesu sterowania.

W pracy zaprezentowano metodologię modelowania i rozwiązywania problemów planowania ruchu wielu robotów, działających we wspólnej przestrzeni roboczej o nieznanym, dynamicznym modelu. Ideą dyskutowanej metodologii modelowania dynamicznego procesu sterowania ruchem robotów jest przybliżenie jej sekwencją statycznych, N-osobowych, dyskretnych problemów decyzyjnych. Pokazano również rozwiązanie problemu wyboru pojedynczego rozwiązania wykorzystujące ideę częściowej koordynacji. Wprowadzono do systemu moduł koordynatora, który staje się aktywny w momencie wykrycia wielokrotnych rozwiązań. W wyniku jego działania otrzymujemy pojedyncze rozwiązanie. Prezentowane podejście wykazuje dużą efektywność i szybkość działania, które potwierdziły wielokrotne symulacje. Przykładowe wyniki tych symulacji zaprezentowano również w niniejszej pracy.

1.1. Sformułowanie problemu

Analizowany problem polega na takim sterowaniu ruchem robotów, które zapewni ich bezkolizyjny przejazd z położeń początkowych do docelowych. Patrząc z perspektywy szybkiego, reaktywnego sterowania, problem redukuje się do wyznaczenia w każdym dyskretnym momencie czasu zbioru sterowań, które zastosowane jako wartości zadane sterowników ruchu spowodują realizację fragmentu całego zadania nawigacyjnego. W dalszej części pracy rozważać będziemy problem sterowania ruchem N robotów mobilnych, działających we wspólnej przestrzeni roboczej W, wewnątrz której znajduje się M ruchomych obiektów (przeszkód). W celu uściślenia dalszych rozważań przyjmijmy następujące założenia:

- Wektor stanu **p** zawierający informację o położeniu i orientacji każdego z robotów jest znany w każdym dyskretnym momencie działania systemu t_n, dla n=1,2,.....;
- Znane są wektory zawierające współrzędne celi **g**;
- Przeszkody posiadają kompaktową geometrię (dającą się przybliżyć za pomocą okręgu);
- Momenty zmian sterowań wszystkich robotów synchronizowane są poprzez napływającą informację ze scentralizowanego systemu sensorycznego.

Oznaczmy stan i-tego robota jako wektor:

$$p_i = [x_i \; y_i \; \Theta_i] \qquad i = 1,2,\ldots N \tag{1}$$

Położenie celu i-tego robota przez wektor:

$$g_i = [x_{g,i} \; y_{g,i}] \qquad i = 1,2,\ldots N \tag{2}$$

Zakłada się, iż system jest w stanie śledzić położenie obiektów (przeszkód). Informacja o orientacji przeszkody estymowana jest na bazie znajomości bieżącego i poprzedniego położenia obiektu. Stan i-tej przeszkody oznaczmy przez:

$$b_j = [x_j \; y_j \; \hat{\Theta}_j] \qquad j = 1,2,\ldots M \tag{3}$$

Estymatę orientacji przeszkody $\hat{\Theta}_j$ otrzymujemy z formuły :

$$\hat{\Theta}_j = \arctan\left(\frac{x_j(t_n) - x_j(t_{n-1})}{y_j(t_n) - y_j(t_{n-1})}\right), \tag{4}$$

gdzie t_n i t_{n-1} oznaczają odpowiednio bieżącą i poprzednią chwilę czasu. Kolejne założenie dotyczy modelu robota. Mianowicie zakładamy, iż obiektem sterowania jest mały laboratoryjny robot mobilny Khepera o kompaktowej geometrii platformy jezdnej. Średnica platformy wynosi około 5[cm]. Robot sterowany jest różnicowo poprzez zmiany prędkości prawego i lewego koła. Na potrzeby prezentowanej metody definiujemy sterowanie i-tego robota jako:

$$u_i = [\omega_i \; v_i] \quad i = 1,2,\ldots N, \tag{5}$$

gdzie ω_i, v_i są odpowiednio prędkością kątową platformy robota mierzoną względem jej środka ciężkości oraz prędkością liniową środka ciężkości. Odpowiadające (5) wartości prędkości liniowych kół robota potrzebne jako wielkości zadane sterownika ruchu otrzymujemy z:

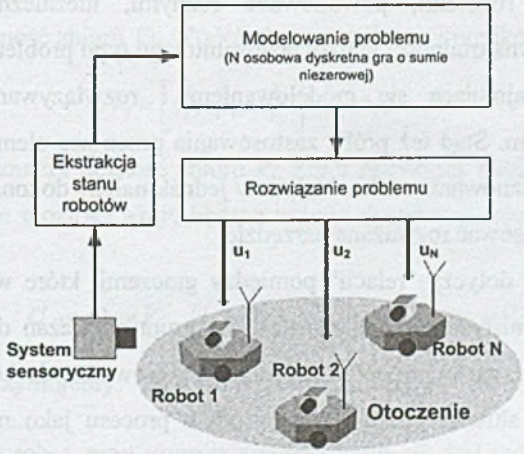
$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1 & L/2 \\ 1 & -L/2 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{6}$$

gdzie L jest rozstawem kół robota. Stosując notację jak wyżej ostatecznie problem może być postawiony w sposób następujący: w każdej chwili czasu ($t = n\Delta t$ dla $n = 0,1,2,....$), znaleźć dla każdego robota sterowanie u_i, $i=1,2,...N$, które zapewni bezkolizyjny i zorientowany na cel ruch robota. Zbiór rozwiązań otrzymany w chwili czasu t_n oznaczamy jako wektor:

$$S(t_n) = \{u_{10}, u_{20}, ..., u_{N0}\} \tag{7}$$

1.2. System sterowania

Opisywana w dalszej części metoda planowania ruchu grupy robotów działających we wspólnej przestrzeni roboczej bazuje na wykorzystaniu dyskretnych gier o sumie niezerowej. W związku z faktem, iż przestrzeń robocza jest dynamiczna, zdecydowano się na



Rys.1. Schemat ideowy scentralizowanego, reaktywnego systemu sterowania

Fig.1. A diagram of the centralized reactive control system

zastosowanie reaktywnej, scentralizowanej struktury systemu sterowania, przedstawionej na rys.1. System sensoryczny dostarcza do systemu sterowania w dyskretnych chwilach czasu informację o stanie robotów. Na bazie tych danych wypracowywany jest zbiór sterowań dla grupy robotów, optymalnych w sensie metody, która zostanie opisana w dalszej części rozdziału. Wyznaczone sterowania są natychmiast stosowane, co powoduje zmianę stanu robotów oraz zmianę stanu realizacji określonego zadania.

2. Planowanie ruchu jako problem decyzyjny

Na rozważany problem planowania ruchu wielu robotów dzielących wspólną przestrzeń roboczą można spojrzeć z perspektywy teorii gier. Utożsamiając każdy z robotów z graczem, czy też decydentem, przedstawiając jego zdanie nawigacyjne jako interes gracza, natomiast sterowanie robota kojarząc z decyzją gracza można zauważyć podobieństwo problemu planowania i problemu decyzyjnego (gry) zachodzącego pomiędzy uczestnikami procesu. Fakt, iż roboty mają niezależne cele, działają wewnątrz wspólnej przestrzeni roboczej, może powodować, że niektóre ze sterowań zastosowanych dla poszczególnych robotów prowadzić będą do kolizji, czy też stosując terminologię teorii gier - do sytuacji konfliktowej. Widzimy więc, iż nawet w przypadku gdy przestrzeń robocza W jest wolna od jakichkolwiek innych obiektów, zadanie planowania ruchu nie jest zadaniem trywialnym (w odróżnieniu od przypadku planowania ruchu pojedynczego robota). Wzajemne interakcje zachodzące pomiędzy robotami, powodowane różnymi, niezależnie podejmowanymi sterowaniami, są przyczyną trudności w rozwiązywaniu tego typu problemów. Teoria gier jest gałęzią matematyki, zajmującą się modelowaniem i rozwiązywaniem problemów o charakterze konfliktowym. Stąd też próba zastosowania przez nas elementów teorii gier do rozwiązania problemu planowania ruchu. Najpierw jednak należy dokonać analizy problemu, aby móc właściwie zastosować rozważane narzędzie.

Pierwsza kwestia dotyczy "relacji" pomiędzy graczami, które w naszym przypadku niekoniecznie są antagonistyczne, co przenosząc na grunt rozważań dotyczących robotów oznacza, iż nie wszystkie zastosowane zbiory sterowań prowadzą do kolizji (konfliktu). To sprawia, iż wydaje się słuszne poszukiwanie modelu procesu jako modelu gry o sumie niezerowej.

Kolejna uwaga związana jest z zagadnieniami numerycznymi. Mianowicie, aby umożliwić numeryczne rozwiązywanie problemu, zakładamy, iż zbiory decyzyjne graczy

posiadają skończoną liczbę elementów, co sprawia, iż kolejnym ograniczeniem nakładanym na poszukiwany model jest jego dyskretna postać.

Trzecie i ostatnie zagadnienie porusza problem dynamiki modelowanego procesu. Chociaż proces nawigacji jest problemem dynamicznym, to w naszym podejściu traktujemy go jako sekwencję problemów statycznych. Takie podejście wydaje się uzasadnione, szczególnie w przypadku, gdy odstępy czasowe pomiędzy kolejnymi chwilami podejmowania decyzji są dostatecznie małe.

W związku z tym, w proponowanym podejściu rozważa się modelowanie procesu bezkolizyjnej nawigacji wielu robotów, w każdej chwili czasu jako dyskretną, statyczną grę o sumie niezerowej w postaci normalnej.

3. Opis metody

3.1. Dyskretyzacja przestrzeni sterowań

W celu otrzymania modelu dyskretnego procesu decyzyjnego należy zdyskretyzować przestrzeń sterowań poszczególnych robotów, otrzymując tym samym zbiory decyzyjne z nimi skojarzone. W tym celu dokonajmy najpierw dyskretyzacji przestrzeni zmienności prędkości kątowej ω_i i-tego robota, otrzymując zbiór:

$$\Omega_i = \left[\omega_i^1, \omega_i^2, \ldots \omega_i^{K_i}\right] \quad i = 1, 2, \ldots N, \tag{8}$$

gdzie K_i oznacza liczność zbioru Ω_i. Podobnie postępujemy z prędkością liniową v_i:

$$V_i = \left[v_i^1, v_i^2, \ldots v_i^{L_i}\right] \quad i = 1, 2, \ldots N, \tag{9}$$

gdzie L_i jak wyżej oznacza liczność zbioru V_i. Zbiór decyzyjny skojarzony z i-tym robotem, określający wszystkie możliwe akcje będące w jego dyspozycji, określony jest jako iloczyn kartezjański:

$$U_i = \Omega_i \times V_i \quad U_i = \{(\omega_i, v_i) : \omega_i \in \Omega_i \cap v_i \in V_i\} \tag{10}$$

W ten sposób otrzymujemy $D_i = \left\{d_i^k\right\}$ $k = 1, 2, \ldots, K_i L_i$ - zbiór akcji możliwych do wykonania przez i-ty robot, przy czym wykonanie akcji d_i^k jest jednoznaczne z wybraniem jednej z możliwych par sterowań (ω_i, v_i). Łatwo jednak zauważyć, iż dyskretyzując w ten sposób przestrzeń sterowań otrzymujemy duże rozmiary zbiorów decyzyjnych, a co za tym

idzie-duże rozmiary problemu. Dlatego też zaproponowany został inny sposób wyznaczenia zbioru decyzyjnego, polegający na dyskretyzacji tylko jednej zmiennej sterującej - prędkości kątowej. Druga z nich, czyli prędkość liniowa wyznaczana jest dynamicznie, na bazie pewnych heurystyk. W ten sposób redukujemy liczność zbioru decyzyjnego i-tego robota do K_i elementów:

$$U_i = \{(\omega_i, v_i) : \omega_i \in \Omega_i \cap v_i = f_{h,i}(d_T, d_{\min})\}, \tag{11}$$

gdzie $f(d_T, d_{\min})$ oznacza heurystyczną funkcję określającą wartość prędkości liniowej na bazie stanu wszystkich robotów, konfiguracji otoczenia i stopnia realizacji zadania nawigacyjnego. Prędkość liniowa zostaje ustalona przed wyznaczeniem modelu procesu decyzyjnego; tak naprawdę akcja utożsamiana będzie odtąd z wyborem jednej z możliwych wartości prędkości kątowej, a tym samym wielkości skrętu robota. Funkcja ta zbudowana została na bazie kilku reguł, które determinują pożądane działanie robota.

I tak pierwsza z reguł mówi, iż robot powinien poruszać się z określoną, możliwie dużą prędkością liniową $v_{i,opt}$ w sytuacji, gdy jego odległość od najbliższego obiektu $L_{i,min}$ jest większa niż pewna wartość graniczna L_{i0}. W przypadku gdy odległość ta jest mniejsza od wartości granicznej, prędkość liniowa powinna zostać zmniejszona w celu umożliwienia wykonania manewru omijania i zmniejszenia ryzyka kolizji.

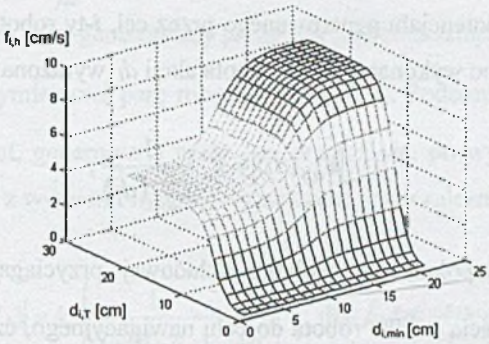
Kolejna z reguł determinuje zachowanie robota w pobliżu celu nawigacyjnego. Zgodnie z nią, po przekroczeniu pewnej granicznej odległości od celu L_{T0} prędkość robota powinna być sukcesywnie zmniejszana aby umożliwić precyzyjne osiągnięcie pożądanej lokalizacji. Biorąc pod uwagę powyższe rozważania zaproponowano następującą postać funkcji wyznaczającej wartość prędkości liniowej:

$$v_i = \frac{1}{2} v_{i,opt} w_T (1 + w_R), \tag{12}$$

gdzie w_R, w_T to funkcje wagowe, określające wpływ odpowiednio najbliższego obiektu i celu nawigacyjnego na wartość prędkości liniowej i-tego robota. Funkcje te określone są następującymi zależnościami:

$$w_T = \frac{1}{1 + e^{-\alpha(L_{i,T} - L_{T_0})}} \quad \text{oraz} \quad w_R = \frac{1}{1 + e^{-\beta(L_{i,\min} - L_{R_0})}} \tag{13}$$

W powyższych zależnościach $L_{i,T}$ i $L_{i,min}$ oznaczają odległość i-tego robota od jego celu nawigacyjnego oraz najbliższego obiektu. Współczynniki α, β determinują charakter wpływu poszczególnych funkcji na wartość prędkości robota w pobliżu wartości granicznych jego odległości od celu i najbliższego obiektu. Im większe wartości tych współczynników, tym bardziej skokowy charakter włączania się poszczególnych składników funkcji heurystycznej. Przykładowy kształt funkcji $f_{h,i}$ przedstawia rys. 5.2.



Rys.2. Przykładowy przebieg funkcji $f_{i,h}$ otrzymany dla następujących parametrów: $v_{i,opt}$=10[cm/s], L_{R0}=10[cm], L_{T0}=5[cm], α=0.9, β=0.7

Fig.2. An exemplary plot of the function $f_{i,h}$ obtained for the following set of parameters: $v_{i,opt}$=10[cm/s], L_{R0}=10[cm], L_{T0}=5[cm], α=0.9, β=0.7

3.2. Modelowanie funkcji kosztów

W celu przedstawienia modelu problemu planowania w postaci modelu dyskretnego procesu decyzyjnego należy wyznaczyć funkcje kosztów związane z poszczególnymi robotami. Wartości umownych kosztów, jakie ponoszą poszczególne roboty w wyniku podjęcia określonych akcji, określone przez ich funkcje kosztów, wyznaczane są ponownie w wyniku doświadczeń i wiedzy empirycznej. W naszym przypadku koszt ponoszony przez i-tego robota zależy od jego odległości od celu nawigacyjnego i bliskości sąsiednich obiektów. Stąd też funkcja kosztów danego robota jest określona jako:

$$I_i(d_1, d_2, ..., d_N) = f_i(d_1, d_2, ..., d_N, b_1, b_2, ...b_M) \quad \text{dla } i = 1, 2, ..., N, \tag{14}$$

gdzie d_i = 1,2,...,K_i oznacza akcję i-tego robota polegającą na wyborze d_i-tego elementu z jego zbioru dopuszczalnych akcji D_i. Natomiast b_j oznacza położenie j-tej przeszkody.

W celu zbudowania funkcji kosztów wykorzystano koncepcję pól potencjalnych [8], stosowaną bardzo często do bezkolizyjnego sterowania robotem mobilnym w środowisku o szybkozmiennej dynamice [4][9]. W swojej oryginalnej wersji [8] koncepcja ta polega na wypełnieniu przestrzeni roboczej polem potencjalnym generowanym przez cel i przeszkody. Robot poruszający się wewnątrz tego pola poddawany jest działaniu sztucznych sil potencjalnych, które przyciągają go do celu i odpychają od przeszkód. Dalej omówimy zastosowanie tej koncepcji do wyznaczenia funkcji kosztów. Zgodnie z tą koncepcją w wyniku wirtualnego potencjału generowanego przez cel, i-ty robot przyciągany jest do niego z siłą, której wartość po wykonaniu przez robota akcji d_i wyrażona jest zależnością:

$$\left|\mathbf{F}_{a,i}(d_i)\right| = k_a \frac{1}{\hat{L}_{g,i}^2(d_i)}, \tag{15}$$

gdzie k_a oznacza współczynnik wpływu składowej przyciągającej. Z kolei $\hat{L}_{g,i}$ jest predykowaną odległością środka robota do celu nawigacyjnego, czyli taką, w jakiej znajdzie się w wyniku wykonania akcji d_i po czasie Δt:

$$\hat{L}_{g,i}(d_i) = \sqrt{\left(x_{g,i} - \hat{x}_i^{d_i}\right)^2 + \left(y_{g,i} - \hat{y}_i^{d_i}\right)^2}, \tag{16}$$

gdzie $x_{g,i}, y_{g,i}$ oznaczają współrzędne celu nawigacyjnego określone przez pożądaną konfigurację docelową robota $p_{i,cel}$. Natomiast $\hat{x}_i^{d_i}, \hat{y}_i^{d_i}$ oznaczają predykowane położenie i-tego robota, wyznaczane z zależności:

$$\begin{aligned} \hat{x}_i^{d_i} &= x_i + v_i^{t_{n-1}} T_o \cos(\Theta_i + \omega_i^{t_{n-1}} T_o) + v_i\,(\Delta t - T_o)\cos(\Theta_i + \omega_i^{t_{n-1}} T_o + \omega_i^{d_i}(\Delta t - T_o)) \\ \hat{y}_i^{d_i} &= x_i + v_i^{t_{n-1}} T_o \sin(\Theta_i + \omega_i^{t_{n-1}} T_o) + v_i\,(\Delta t - T_o)\cos(\Theta_i + \omega_i^{t_{n-1}} T_o + \omega_i^{d_i}(\Delta t - T_o)), \end{aligned} \tag{17}$$

gdzie $\omega_i^{d_i}$ oznacza wielkość prędkości kątowej robota, ustawioną w wyniku podjęcia akcji d_i. W niniejszym opracowaniu dopuszczalne jest istnienie opóźnień pomiędzy chwilą rozpoczęcia procesu obliczenia modelu problemu decyzyjnego a chwilą uruchomienia sterowania. W systemie rozważany czas opóźnienia T_0 modelowany jest jako górne

oszacowanie czasu obliczeń oraz opóźnień w transmisji sygnałów. Wartości $\omega_i^{t_{n-1}}, v_i^{t_{n-1}}$ oznaczają poprzednie sterowania, które wciąż sterują ruchem robota przez okres T_0.

Składowa odpychająca jest wypadkową wszystkich sił wirtualnych generowanych przez pozostałe roboty oraz przeszkody. Stąd wyznaczana jest jako suma wektorowa:

$$\mathbf{F}_{\mathbf{r},\mathbf{i}}(d_1,\ldots d_N) = \sum_{\substack{j=1\\ j\neq i}}^{N} \mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}\left(d_i,d_j\right) + \sum_{j=1}^{M} \mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i) \quad \text{dla } i=1,2,\ldots,N, \tag{18}$$

gdzie $\mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}(d_i,d_j)$ oznacza siłę generowaną przez j-ty robot działającą na robot i-ty, po wykonaniu przez wyżej wymienioną parę robotów akcji d_j, d_i. Podobnie, $\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)$ oznacza siłę działającą na i-ty robot, generowaną przez j-tą przeszkodę, po wykonaniu przez robot akcji d_i. Wartość pierwszej z wymienionych sił wyznaczana jest z zależności:

$$\begin{cases} \left|\mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}(d_i,d_j)\right| = k_r\left(\dfrac{1}{\hat{L}_{ij}(d_i,d_j)} - \dfrac{1}{L_0}\right)^2 & \text{dla } \hat{L}_{ij}(d_i,d_j) < L_0 \\ \left|\mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}(d_i,d_j)\right| = 0 & \text{w przeciwnym wypadku} \\ \left|\mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}(d_i,d_j)\right| = \infty & \text{dla } \hat{L}_{ij}(d_i,d_j) < r_i + r_j, \end{cases} \tag{19}$$

gdzie L_0 oznacza promień strefy, wewnątrz której na robot oddziałują siły wirtualne, r_i, r_j są promieniami platform jezdnych robotów, natomiast $\hat{L}_{i,j}\left(d_i,d_j\right)$ to predykowana odległość robota i-tego od robota j-tego, wyznaczana z zależności:

$$\hat{L}_{i,j}(d_i,d_j) = \sqrt{\left(\hat{x}_i^{d_i} - \hat{x}_j^{d_j}\right)^2 + \left(\hat{y}_i^{d_i} - \hat{y}_j^{d_j}\right)^2} \tag{20}$$

Predykowane położenie robotów $(\hat{x}_i^{d_i}, \hat{y}_i^{d_i}), (\hat{x}_j^{d_j}, \hat{y}_j^{d_j})$ wyznaczamy z (17). Podobnie obliczmy wartość siły $\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)$:

$$\begin{cases} \left|\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)\right| = k_{rb}\left(\dfrac{1}{\hat{L}_{ij}(d_i)} - \dfrac{1}{L_0}\right)^2 & \text{dla } \hat{L}_{ij}(d_i) < L_0 \\ \left|\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)\right| = 0 & \text{w przeciwnym wypadku} \\ \left|\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)\right| = \infty & \text{dla } \hat{L}_{ij}(d_i) < r_i + r_{b,j}, \end{cases} \tag{21}$$

gdzie $r_{b,j}$ jest promieniem okręgu opisanego na j-tej przeszkodzie, natomiast $\hat{L}_{i,j}(d_i)$ oznacza predykowaną odległość i-tego robota od j-tej przeszkody po czasie Δt:

$$\hat{L}_{ij}(d_i) = \sqrt{\left(\hat{x}_i^{d_i} - \hat{x}_{o,j}\right)^2 + \left(\hat{y}_i^{d_i} - \hat{y}_{o.j}\right)^2} \tag{22}$$

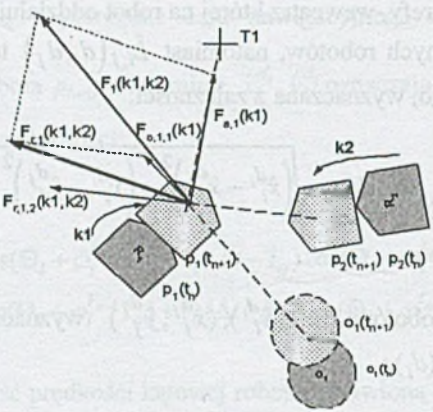
Estymowane położenie j-tej przeszkody wyznaczane jest na bazie znajomości aktualnego jej stanu $\mathbf{b}_j$ oraz stanu wcześniejszego $\mathbf{b}_j^{t_{n-1}}$:

$$\begin{aligned} \hat{x}_{b,j} &= 2x_{b,j} - x_{b,j}^{t_{n-1}} \\ \hat{y}_{b,j} &= 2y_{b,j} - y_{b,j}^{t_{n-1}} \end{aligned} \tag{23}$$

Współczynniki k_r w zależości (19) i k_{rb} w (21) regulują wpływ sił odpychających na wypadkową siłę, której działaniu poddany jest robot. Wreszcie siła wypadkowa oddziałująca na i-ty robot wyznaczana jest jako suma wektorowa:

$$\mathbf{F_i} = \mathbf{F_{a,i}} + \mathbf{F_{r,i}} \tag{24}$$

Interpretacja geometryczna opisywanych sił, których działaniu poddany jest robot, przedstawiona została na rys. 3.



Rys. 3. Geometryczna interpretacja wirtualnych sił oddziałujących na robot

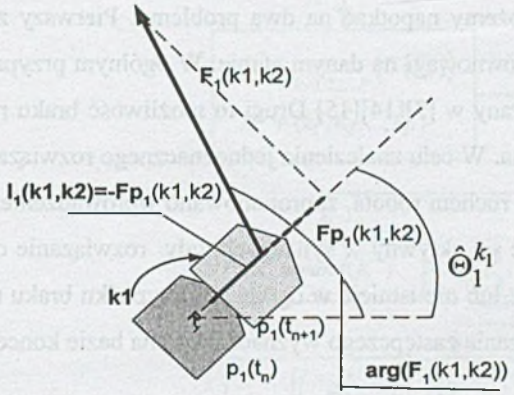
Fig.3. A geometrical interpretation of virtual forces applied to the robot

Kierunek i zwrot wektora siły (24) określa kierunek, w którym powinien poruszać się robot, aby bezkolizyjnie przemieszczać się w kierunku celu. Wartość funkcji kosztów i-tego robota, będąca rezultatem podjętych akcji $\{d_1,...,d_i,...,d_N\}$, wyznaczona została jako rzut siły, której działaniu poddany jest robot na kierunek, zdeterminowany przez orientację robota uzyskaną w wyniku akcji d_i (rys. 4). Wartość I_i wyznaczana jest więc z zależności:

$$I_i\left(d_1, d_i, ..., d_N\right) = -\left|\mathbf{F_i}\right| \cos\left(\arg\left(\mathbf{F_i}\right) - \hat{\Theta}_i^{d_i}\right), \tag{25}$$

gdzie

$$\hat{\Theta}_i^{d_i} = \Theta_i + \omega_i^{t_{n-1}} T_0 + \omega_i^{d_i}\left(\Delta t - T_0\right) \tag{26}$$



Rys. 4. Geometryczna interpretacja wirtualnych sił oddziałujących na robot

Fig.4. A geometrical interpretation of virtual forces applied to the robot

4. Rozwiązanie problemu decyzyjnego

W poprzedniej sekcji otrzymaliśmy model procesu planowania ruchu wielu robotów wyznaczony jako N-osobowa, dyskretna gra w postaci normalnej. Model ten opisuje sytuację, w której konflikt jest powodowany możliwością kolizji, do której mogą doprowadzić niektóre

kombinacje decyzji podejmowanych przez graczy na etapie zdeterminowanym w chwili t_n. Poszukujemy zatem rozwiązania problemu, w postaci zbioru decyzji $S_0=\{d_{10},d_{20},...,d_{N0}\}$, który zapewni każdemu z graczy jak najniższe koszty. Jak już powiedzieliśmy wcześniej, ze względu na brak możliwości komunikacji pomiędzy robotami rozsądne wydaje się poszukiwanie rozwiązania problemu jako niekooperacyjnego. Najbardziej znaną koncepcją rozwiązania tego typu problemów jest koncepcja punktu równowagi Nasha, zgodnie z którą poszukujemy rozwiązania z zależności:

$$\begin{aligned} &I_1\left(d_{10},d_{20},...d_{N0}\right)\le I_1\left(d_1,d_{20},...d_{N0}\right)\\ &I_2\left(d_{10},d_{20},...d_{N0}\right)\le I_2\left(d_{10},d_2,...d_{N0}\right)\\ &\vdots\\ &I_N\left(d_{10},d_{20},...d_{N0}\right)\le I_N\left(d_{10},d_{20},...d_N\right)\end{aligned} \tag{27}$$

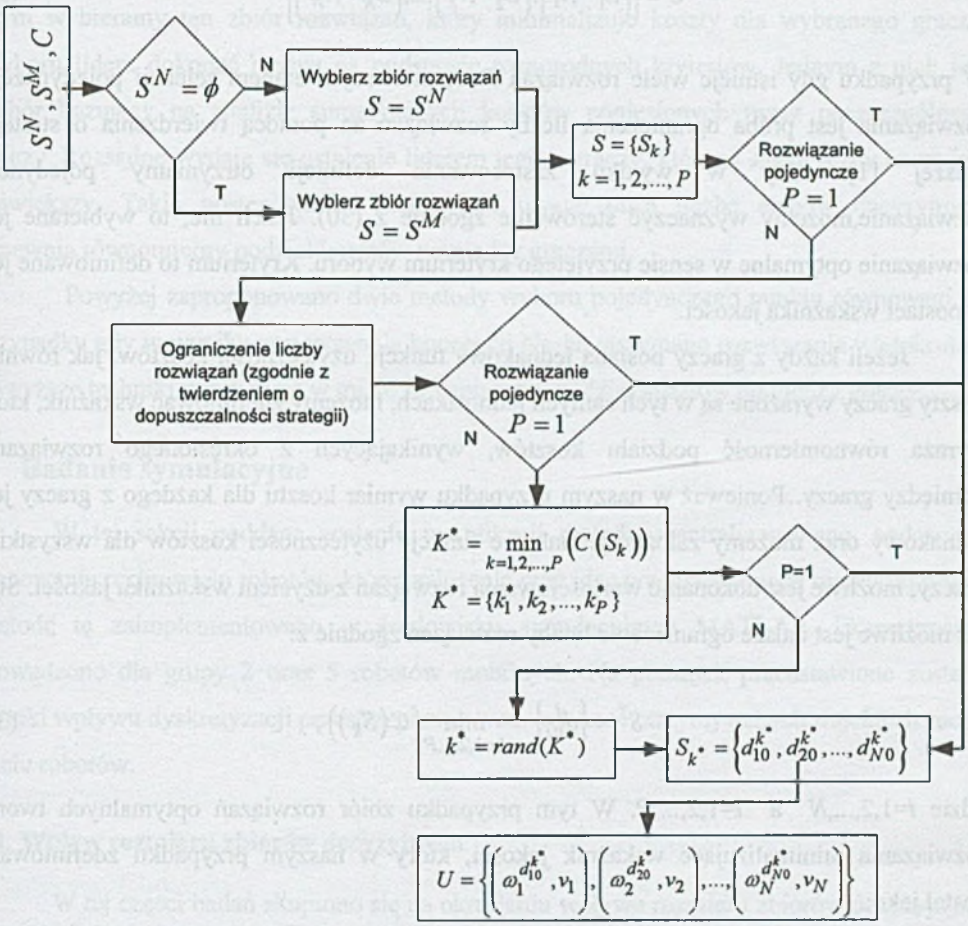
Stosując tą koncepcję możemy napotkać na dwa problemy. Pierwszy z nich to możliwość istnienia wielu punktów równowagi na danym etapie. W ogólnym przypadku jest to problem trudny, szeroko dyskutowany w [7][14][15] Drugi to możliwość braku rozwiązań problemu, zgodnie z koncepcją Nasha. W celu znalezienia jednoznacznego rozwiązania, koniecznego do prawidłowego sterowania ruchem robota, zaproponowano wprowadzenie do systemu modułu arbitrażowego, który staje się aktywny w sytuacjach, gdy rozwiązanie otrzymane zgodnie z (27) nie jest jednoznaczne lub nie istnieje w ogóle. W przypadku braku rozwiązań zgodnych (27) poszukujemy rozwiązania zastępczego wyznaczanego na bazie koncepcji *minimaksowej*:

$$\begin{aligned} &d_{10}=\min_{d_1}\max_{d_2,...d_N} I_1\left(d_{10},d_2,...,d_N\right)\\ &\vdots\\ &d_{N0}=\min_{d_N}\max_{d_1,...d_{N-1}} I_N\left(d_1,d_2,...,d_{N0}\right)\end{aligned} \tag{28}$$

Oznaczmy najpierw zbiory rozwiązań otrzymanych zgodnie z (27) jako S^N, natomiast zbiory uzyskane za pomocą (28) jako S^M:

$$S^N=\{d_{i0}^n\}\quad S^M=\{d_{i0}^m\}\quad n=1,2,...,P_N,\ m=1,2,...,P_M,\ i=1,2,...N \tag{29}$$

Schemat blokowy algorytmu arbitrażu przedstawiono na rys. 5. Danymi wejściowymi algorytmu arbitrażu są zbiory rozwiązań (29) oraz dodatkowe kryteria wyboru w postaci wskaźnika jakości C.



Rys. 5. Schemat arbitrażowy

Fig.5. An arbitration algorithm diagram

W pierwszym kroku sprawdzane jest, czy zbiór rozwiązań S^N nie jest pusty. Jeśli nie jest pusty, wówczas wybierany on jest jako zbiór rozwiązań preferowanych $S = S^N = \{S_k\}$ $S_k = \{d_{i0}^k\}$ $k = 1, 2, ..., P, i = 1, 2, ...N$. W przeciwnym przypadku wybierany jest zbiór rozwiązań minimaksowych $S = S^M$. Następnie sprawdzana jest liczność P zbioru

S_k. W przypadku gdy P=1, rozwiązaniem problemu jest zbiór decyzji $S^*=S_1$. Wówczas zbiór sterowań wyznaczany jest jako:

$$U=\left\{\left(\omega_1^{d_{10}},v_1\right),\left(\omega_2^{d_{20}},v_2\right),\ldots,\left(\omega_N^{d_{N0}},v_N\right)\right\} \tag{30}$$

W przypadku gdy istnieje wiele rozwiązań P>1, pierwszym etapem selekcji pojedynczego rozwiązania jest próba ograniczenia liczby rozwiązań za pomocą twierdzenia o strategii lepszej [1]. Gdy w wyniku zastosowania definicji otrzymamy pojedyncze rozwiązanie,możemy wyznaczyć sterowanie zgodnie z (30). Jeżeli nie, to wybierane jest rozwiązanie optymalne w sensie przyjętego kryterium wyboru. Kryterium to definiowane jest w postaci wskaźnika jakości.

Jeżeli każdy z graczy posiada jednakową funkcję użyteczności kosztów, jak również koszty graczy wyrażone są w tych samych jednostkach, możemy zdefiniować wskaźnik, który wyraża równomierność podziału kosztów, wynikających z określonego rozwiązania pomiędzy graczy. Ponieważ w naszym przypadku wymiar kosztu dla każdego z graczy jest jednakowy oraz możemy założyć jednakowe funkcje użyteczności kosztów dla wszystkich graczy, możliwe jest dokonanie wartościowania rozwiązań z użyciem wskaźnika jakości. Stąd też możliwe jest dalsze ograniczenie liczby rozwiązań zgodnie z:

$$S^*=\left\{d_{i0}^l\right\}=\min_{k=1,2,\ldots P}\left(C(S_k)\right), \tag{31}$$

gdzie i=1,2,...,N a l=1,2,...,P. W tym przypadku zbiór rozwiązań optymalnych tworzą rozwiązania minimalizujące wskaźnik jakości, który w naszym przypadku zdefiniowany został jako:

$$\begin{aligned} C(S_k)&=\sum_{i=1}^{N}\left(I_i(S_k)+\left|I_i(S_k)-\bar{I}\right|\right)\\ \bar{I}&=\frac{1}{N}\sum_{i=1}^{N}I_i(S_k)\end{aligned} \tag{32}$$

Jeżeli tak otrzymany zbiór jest jednoelementowy $\overline{\overline{S^*}}=P^*=1$, wówczas wyznaczamy sterowanie zgodnie z (30). W przeciwnym wypadku musimy wybrać jeden z elementów otrzymanego zbioru. W naszym przypadku zastosowaliśmy metodę losową, zgodnie z którą

numer determinujący ostateczne rozwiązanie pozyskiwany jest w wyniku losowania liczby całkowitej z przedziału $<1;P^*>$. Innym rozwiązaniem może być metoda tzw. hierarchii wtórnej, zgodnie z którą ustalamy gracza, tzw. lidera, dla którego koszt ma być najniższy, po czym wybieramy ten zbiór rozwiązań, który minimalizuje koszty dla wybranego gracza. Wyboru lidera dokonać można na podstawie różnorodnych kryteriów. Jednym z nich jest wybór bazujący na analizie sumarycznych kosztów poniesionych przez poszczególnych graczy. Rozsądne wydaje się ustalenie liderem tego z graczy, którego sumaryczny koszt jest największy. Takie postępowanie, biorąc pod uwagę dużą liczbę etapów decyzyjnych zapewnia równomierny podział kosztów pomiędzy graczami.

Powyżej zaproponowano dwie metody wyboru pojedynczego punktu równowagi, w przypadku gdy w wyniku zastosowania koncepcji Nasha otrzymano rozwiązania wielokrotne. Powyższe techniki zapewniają w miarę równomierny podział kosztów pomiędzy graczy.

5. Badania symulacyjne

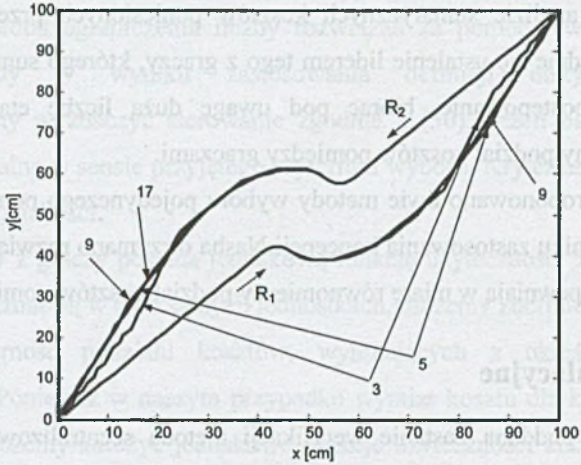
W tej sekcji poddana zostanie weryfikacji metoda scentralizowanego, nadążnego planowania ruchu wielu robotów, której założenia oraz ideę przedstawiono w niniejszej pracy. Metodę tę zaimplementowano w środowisku symulacyjnym MATLAB. Eksperymenty prowadzono dla grupy 2 oraz 5 robotów mobilnych. Na początek przedstawione zostaną wyniki wpływu dyskretyzacji przestrzeni sterowań na jakość otrzymywanych trajektorii ruchu wielu robotów.

5.1. Wpływ rozmiaru zbiorów decyzyjnych

W tej części badań skupiono się na określeniu wpływu rozmiaru zbiorów decyzyjnych na jakość procesu sterowania ruchem dwóch robotów. Przeprowadziliśmy badania dla zbiorów decyzyjnych o rozmiarach od 3 do 17 elementów. Przyjęto następujące parametry modelu: Okres decyzyjny $\Delta t = 0.1\,[s]$. Pożądana prędkość liniowa robotów $v_{i,opt}$ ustalona została dla obydwu robotów na poziomie $v_{1,opt} = v_{2,opt} = 25\,[cm/s]$. Rysunek 6 przedstawia trajektorie otrzymane dla poszczególnych rozmiarów zbiorów decyzyjnych. Widać na rysunku, iż wraz ze wzrostem liczności zbiorów decyzyjnych rośnie gładkość otrzymanych trajektorii. Jednakże aby ocenić w sposób bardziej dokładny jakość otrzymywanych trajektorii zaproponowano użycie wskaźnika jakości zdefiniowanego w sposób następujący:

$$J_i = \frac{1}{H}\sum_{n=1}^{H}\left(u_i(t_n) - u_i(t_{n-1})\right)^2 \quad for \quad i = 1,2,...N, \tag{33}$$

gdzie $u_i\ (t_n\ )$ i $u_i\ (t_{n-1})$ oznaczają odpowiednio sterowania zastosowane w n-tej i poprzedzającej ją chwili n-1-szej dyskretnej chwili czasu.

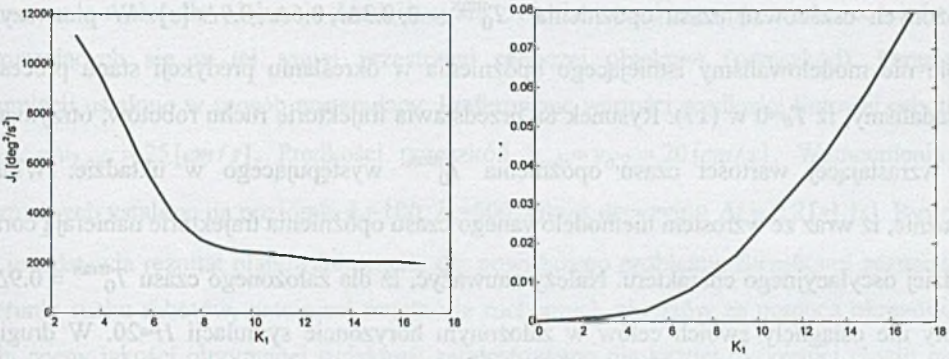


Rys. 6. Wyniki symulacji pokazujących wpływ rozmiaru zbiorów decyzyjnych na jakość trajektorii

Fig. 6. Result of enlarging the size of decision sets. Simulations were carried out for 3,5,9, and 17 possible decisions

Ponadto jako sterowanie przyjmujemy tutaj prędkość kątową i-tego robota, wynikającą z wyboru określonego elementu zbioru decyzyjnego, otrzymanego jak rozwiązanie problemu na n-tym i n-1-szym etapie decyzyjnym, przy czym H oznacza całkowitą liczbę etapów decyzyjnych prowadzących do osiągnięcia celu. Wykres zmienności wskaźnika jakości w zależności od liczby elementów zbioru decyzyjnego, otrzymany dla powyższego eksperymentu, dla robota 1 przedstawia rys. 7a. Na rysunku 7b z kolei przedstawiono zależność czasu wyznaczania rozwiązania problemu dla różnych liczności zbiorów decyzyjnych. Widać wyraźnie, iż jakość otrzymywanych trajektorii stabilizuje się bardzo szybko. Dalszy wzrost liczby dyskretnych wartości sterowań, w przeciwieństwie do czasu obliczeń, nie wpływa znacząco na jakość sterowania. Wynika z tego, iż nie jest konieczne zakładanie dużej liczby dopuszczalnych, dyskretnych wartości sterowań. Poprzez właściwy dobór wielkości sterowań możemy osiągnąć zadowalające rezultaty już dla 5 elementowych

zbiorów decyzyjnych, czego wyraźną zaletą jest stosunkowo mały czas obliczeń konieczny do uzyskania rozwiązania.



Rys.7. Wpływ rozmiaru zbioru decyzyjnego na jakość sterowania (a) i czas obliczeń (b)

Fig.7. The comparison of control quality of the robot 1defined by performance index J_1 (a) and the computation time (b)

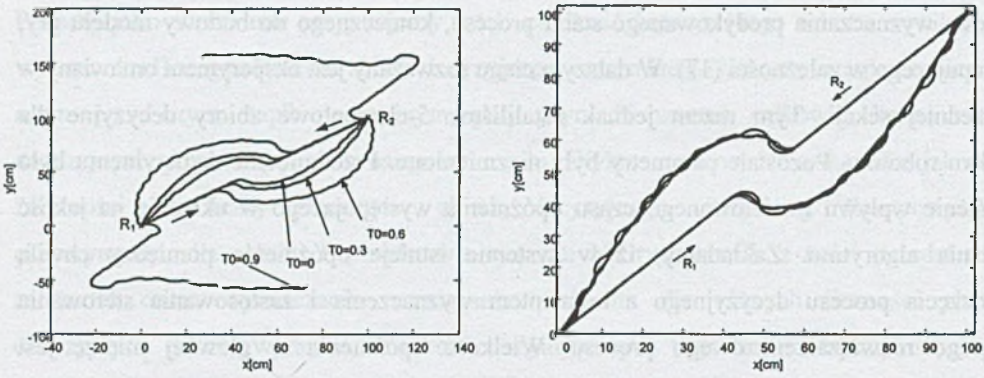
Czasy obliczeń uzyskano na komputerze klasy PC z zegarem 800 MGHz. Należy przy tym zauważyć, iż implementacja algorytmu poza środowiskiem MATLAB daje nieporównywalnie niższe czasy obliczeń.

5.2. Badanie wpływu czasu opóźnienia

Kolejnym zagadnieniem było określenie znaczenia modelowania czasu opóźnienia w procesie wyznaczania predykowanego stanu procesu, koniecznego do budowy modelu gry, występującego w zależności (17). W dalszym ciągu rozważany jest eksperyment omawiany w poprzedniej sekcji. Tym razem jednak ustaliliśmy 5-elementowe zbiory decyzyjne dla obydwu robotów. Pozostałe parametry były niezmienione. Przedmiotem eksperymentu było określenie wpływu modelowanego czasu opóźnienia występującego w układzie na jakość działania algorytmu. Zakładamy, iż w systemie istnieje opóźnienie pomiędzy chwilą rozpoczęcia procesu decyzyjnego a momentem wyznaczenia i zastosowania sterowania będącego rozwiązaniem owego procesu. Wielkość opóźnienia w głównej mierze jest proporcjonalna do czasu obliczeń, koniecznych do wyznaczenia sterowania. W trakcie eksperymentu czas opóźnienia występujący w systemie symulujemy zgodnie z modelem:

$$T_0^{syst} = \frac{1}{2} T_0^{\max} \left(1 + \delta t\right), \quad \delta t \in \langle 0,1 \rangle, \tag{34}$$

gdzie δt jest liczbą losową, natomiast T_0^{max} jest maksymalnym oszacowaniem czasu opóźnienia, jakie może wystąpić w układzie. Symulacje przeprowadziliśmy dla czterech założonych oszacowań czasu opóźnienia $T_0^{max} = 0, 0.3\Delta t, 0.6\Delta t, 0.9\Delta t\,[s]$. W pierwszym etapie nie modelowaliśmy istniejącego opóźnienia w określaniu predykcji stanu procesu, zakładaliśmy, iż T_0=0 w (17). Rysunek 8a przedstawia trajektorie ruchu robotów, otrzymane dla wzrastającej wartości czasu opóźnienia T_0^{max} występującego w układzie. Widać wyraźnie, iż wraz ze wzrostem niemodelowanego czasu opóźnienia trajektorie nabierają coraz bardziej oscylacyjnego charakteru. Należy zauważyć, iż dla założonego czasu $T_0^{max} = 0.9\Delta t$ roboty nie osiągnęły swoich celów w założonym horyzoncie symulacji H=20. W drugim etapie eksperymentu modelowaliśmy czas opóźnienia przyjmując w (17) $T_0 = T_0^{max}$. Rezultat symulacji przedstawia rys. 8b. Można zauważyć wyraźną poprawę jakości działania algorytmu. Trajektorie stają się mniej oscylacyjne. W przypadku krytycznym występującym w poprzednim eksperymencie, metoda w znakomity sposób radzi sobie z istniejącym opóźnieniem, generując poprawne trajektorie pozwalające na osiągnięcie celu przez roboty w założonym horyzoncie czasowym symulacji. Stąd nasuwa się wniosek, iż modelowanie opóźnienia, które zawsze towarzyszy rzeczywistym układom, poprawia zdecydowanie jakość działania algorytmu.



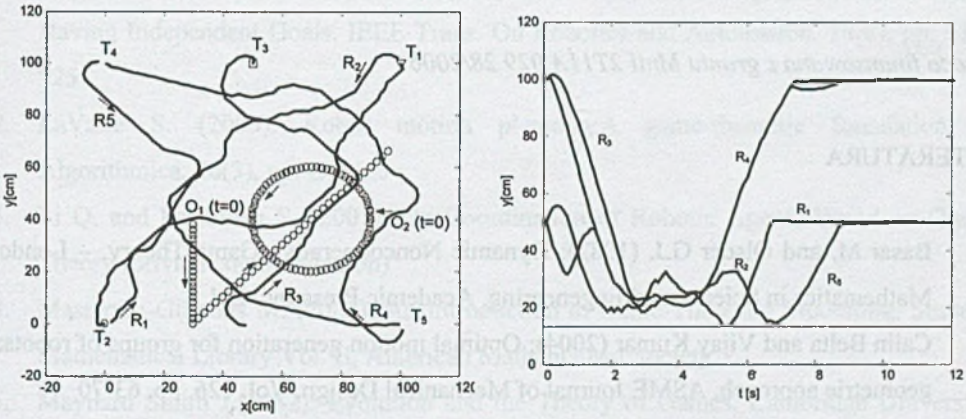
(a) (b)

Rys.8. Wpływ uwzględnienia (a) i nie uwzględnienia (b)czasu opóźnienia w modelu

Fig.8. Comparison of results of simulations made for two robots in case when T_0 was not modeled (a) and when it was (b)

5.3. Planowanie ruchu wielu robotów w obecności ruchomych przeszkód

Ostatni z prezentowanych eksperymentów pokazuje rezultat działania metody planowania ruchu dla większej, składającej się z pięciu robotów grupy, w obecności dwóch poruszających się w tej samej przestrzeni roboczej obiektów (przeszkód). Parametry symulacji ustalono w sposób następujący. Preferowane wartości prędkości liniowej robotów: $v_{1,opt} = v_{2,opt} = 25\,[cm/s]$. Prędkości przeszkód $v_{o,1} = v_{o,2} = 20\,[cm/s]$. Wzmocnienia sił wirtualnych ustalono na poziomie k_r=100, k_a=500. Okres decyzyjny $\Delta t = 0.2\,[s]$ [s]. Rysunek 9a przedstawia rezultat planowania ruchu dla powyższego problemu. Strzałkami zaznaczono kierunki ruchu robotów, natomiast trajektorie ruchomych obiektów za pomocą okręgów. W celu oceny jakości otrzymanej trajektorii zarejestrowano dla każdej dyskretnej chwili czasu odległość i-tego robota do najbliższego obiektu otoczenia (innego robota czy też przeszkody). Przebieg tych wielkości pokazano na rys. 9b. Widać, iż najmniejsza z zarejestrowanych odległości jest na poziomie 13 [cm], co dla założonych wielkości robotów i przeszkód (r=5.5 [cm]) jest potwierdzeniem bezkolizyjnego ruchu robotów.



(a) (b)

Rys.9. Przebieg bezkolizyjnych trajektorii otrzymanych w eksperymencie (a). Wykres odległości poszczególnych robotów do najbliższych obiektów otoczenia (b)

Fig.9. Collision free trajectories obtained in a simulation for 5 robots and 2 obstacles (a). Time plot of the distance to the nearest object (b)

6. Podsumowanie

W prezentowanej pracy przedyskutowano koncepcję i działanie scentralizowanego, nadążnego systemu planowania ruchu wielu robotów, którego ideą było wykorzystanie do modelowania procesu sterowania dyskretnych gier w postaci normalnej. Eksperymenty wykazały, iż system radzi sobie bardzo dobrze z problemem sterowania ruchem wielu robotów w dynamicznym środowisku. Pomimo iż na każdym z etapów decyzyjnych sterowanie odbywa się na bazie znajomości stanu bieżącego oraz jednokrokowej predykcji, to trajektorie otrzymane w wyniku działania systemu możemy uznać za satysfakcjonujące. Dodatkową zaletą jednoetapowego planowania ruchu jest wynikająca z tego faktu duża zdolność nadążania systemu za zmianami otoczenia i odporność na niepewność budowanego modelu procesu jak również, co ma niebagatelne znaczenie, mały koszt obliczeniowy prezentowanych algorytmów. Chociaż koncepcja takiego sterowania była początkowo tylko etapem przejściowym do implementacji omawianych algorytmów w ramach hybrydowej [16] struktury systemu, to jednak sama w sobie dostarczyła ciekawych rezultatów, co spowodowało, iż celowe stało się poświęcenie jej badaniu większej uwagi.

Praca finansowana z grantu MniI 2T11A 029 28/2006

LITERATURA

1. Basar M. and Olsder G.J. (1982): Dynamic Noncooperative Game Theory. – London: Mathematics in Science and Engeneering, Academic Press Inc. Ltd.
2. Calin Belta and Vijay Kumar (2004): Optimal motion generation for groups of robots: a geometric approach, ASME Journal of Mechanical Design, Vol. 126, pp. 63-70
3. Esposito J. and Kumar V. (2000): Closed loop motion plans for mobile robots. - Proc. IEEE Intl Conf. On Robotics and Automation (ICRA), pp.1020-1025
4. Ge S.S and Cui Y.J.(2002): Dynamic motion planning for mobile robots using potential field method. - Autonomous robot vol. 13, Kluwer Academic Publisher, p. 207-222, Netherlands

5. Gerkey B. and Mataric M.J. (2002): Sold!: Auction methods for multi-robot coordination. IEEE Transactions on Robotics and Automation, special issue on Advances in Multi-Robot Systems, 18(5), pp. 758-786.
6. Golfarelli M.(1998): A game theory Approach to Coordination in MAS. Proc 13 th Europe Conference on Artificial Intelligence, Brighton UK
7. Harsanyi J.C. and Selten R.(1998): A general Theory of Equilibrium Selection in Games. Massachusetts: The MIT Press
8. Khatib O.(1986): Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, The International Journal of Robotics Research 5(1), pp. 90-98
9. Koren Y. and Borenstein J: (1991): Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation. - Proc. of the IEEE Conference on Robotics and Automation, Sacramento, CA
10. LaValle S. and Hutchinson S. (1994): Path selection and coordination for multiple robots via Nash equilibria. – Proc. 1994 IEEE Intl Conf. Robot. & Automation, pp. 1847-1852
11. LaValle S. and Hutchinson S. (1998): Optimal Motion Planning for Multiple Robots Having Independent Goals. IEEE Trans. On Robotics and Automation, 14(6), pp. 912-925
12. LaValle S. (2000): Robot motion planning:A game-theoretic foundation. – Algorithmica, 26(3), pp.430-465
13. Li Q. and Payandeh S. (2001): On Coordination of Robotic Agents Based on Game Theory (private communication)
14. Masterton-Gibbons M. (2001): An Introduction to Game-Theoretic Modelling, Student Mathematical Library Vol. II, American Mathematical Society
15. Maynard Smith J.(1982): Evolution and the Theory of Games. Cambridge University Press, Cambridge
16. Skrzypczyk K. (2005): Hybrydowy system sterowania robotem mobilnym. Rozprawa doktorska, Wydział Automayki, Elektroniki i Informatyki, Instytut Automatyki, Politechnika Śląska, Gliwice

Recenzent: Prof. dr hab. inż. Ryszard Tadeusiewicz

Abstract

In this paper we propose a solution to multiple equilibria problem based on the idea of partial coordination. The coordinator module is added to the navigation system and it becomes active when multiple equilibria are reported by the moving robot agents. For the coordinator we will also, synonymously, use the term arbiter. The task the arbiter, apart from choosing the best one from several equilibria, is also resolving the situation when no equilibria exists. The partially centralized design allows efficient and collision free control for motion of robots. We present a series of simulations that prove effectiveness of our approach. We investigate the influence of primary algorithm parameters for two differential driven mobile robots.