

Grzegorz Pusz
 Politechnika Wrocławska

ALGORYTMY STEROWANIA GRUPA RUCHOMYCH ROBOTÓW
 Z UWZGLĘDNIENIEM UNIKANIA KOLIZJI

Streszczenie.Praca dotyczy problemu sterowania grupą ruchomych robotów w ograniczonym obszarze, w którym znajduje się pewna liczba nieruchomych przeszkód. Uwzględnione są dynamiczne własności robotów. Zadanie polega na sterowaniu każdym z robotów w taki sposób, aby przejechał on z zadanego punktu początkowego do końcowego, unikając kolizji z przeszkodami i innymi robotami, w jak najmniejszym czasie.

1. Wprowadzenie

Będziemy się zajmować problemem sterowania grupą ruchomych robotów (wózków) w ograniczonym obszarze, w którym znajduje się pewna liczba nieruchomych przeszkód. Zakładamy, że położenia przeszkód są znane. Zadaniem dla każdego robota jest realizacja nadchodzących do systemu zadań transportowych, polegających na przejechaniu z punktu początkowego do zadanego punktu końcowego w ten sposób, aby nie doszło do kolizji z przeszkodą, lub innym robotem i aby robot nie wyjechał poza obszar dopuszczalny. Przyjmujemy, że robot posiada kołowy układ jezdny (rys.1), przy czym wielkości sterujące dla tego układu są następujące:

v - moment siły działający na kołko napędzające robot,
 w - moment siły działający na kołko sterujące zakreśleniem robota.
 Zakładamy, że zmienne sterujące v i w mogą przyjmować tylko wartości ze zbiorów trójelementowych:

$$v \in \{ -V, 0, V \}, \quad w \in \{ -W, 0, W \}.$$

Tak więc możemy na każde z kółek przyłożyć pełny moment ze znakiem plus lub minus lub nie przyłożyć nic. Nie ma możliwości przyłożenia pośrednich wartości momentu. Zbiór możliwych sterowań jest zatem dziewięcioelementowy.

Przyjmijmy następujące oznaczenia:

a - odległość kółka skrecającego od osi poprzecznej robota,

r - promień kółka napędzającego,

M_r - masa robota, k_f - współczynnik tarcia,

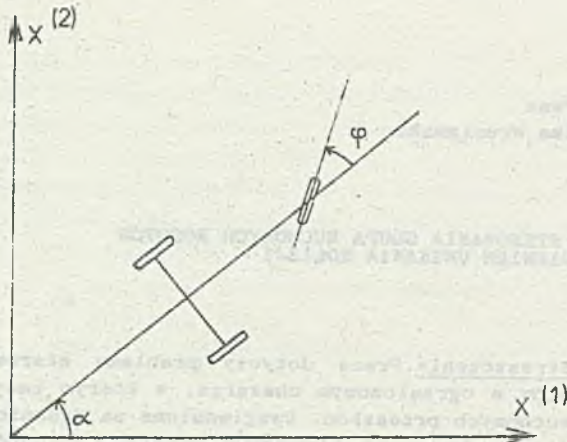
g - przyspieszenie ziemskie ($g = 9.81 \text{ m/s}^2$).

Aktualny stan (położenie i prędkość) robota jest opisany następującymi

wielkościami:

$x^{(1)}, x^{(2)}$ - współrzędne robota w układzie zewnetrznym (rys.1).

Praca była częściowo finansowana przez CPBR nr 02.13.



Kys.1. Opis położenia robota w przestrzeni
Fig.1. Description of robot location in space

- α - kąt pomiędzy osią wzdłużną robota a osią $x^{(1)}$,
 ϕ - kąt pomiędzy osią wzdłużną robota a płaszczyzną kółka skracającego,
 p - prędkość liniowa robota w aktualnym kierunku jazdy.

Przy założeniu, że jazda odbywa się bez poślizgu, można wyprowadzić następujące przybliżone równania ruchu robota, uwzględniające dynamikę tego ruchu:

$$\begin{aligned} \frac{dx^{(1)}}{dt} &= p(t) \cos [\alpha(t) + \phi(t)] \\ \frac{dx^{(2)}}{dt} &= p(t) \sin [\alpha(t) + \phi(t)] \\ \frac{d\alpha}{dt} &= \frac{p(t) \sin [\phi(t)]}{r} \quad (1) \\ \frac{d\phi}{dt} &= w(t) \\ \frac{dp}{dt} &= \frac{v(t) / r - g \cdot p \cdot M_r \cdot k_f}{M_r} \end{aligned}$$

Jest to model uproszczony, który uwzględnia najważniejsze czynniki, tj. wpływ masy robota i tarcia kółka i przekładni na przyspieszenie w aktualnym kierunku ruchu wózka. Pominięte zostały inne siły tarcia, przyspieszenia wiążące się z faktem, że wózek nie jest układem inercyjnym, moment bezwładności wózka przy jego obrocie wokół własnej osi itp. Pełny model dynamiczny wózka można uzyskać pisząc dla niego równania Lagrange'a. Uzyskane zależności będą jednak tak skomplikowane, że uniemożliwiłyby działanie algorytmu sterowania, ponieważ bardzo znacznie wydłużyłoby się czas obliczeń przy korzystaniu z modelu. Ponadto zachowanie wózka jest wystarczająco dobrze opisane powyższymi równaniami tak, że zastosowanie modelu dokładnego nie zmieni w sposób istotny wyników. Podany wyżej model jest modelem ciągłym. Ponieważ zakładamy, że sterowanie jest dyskretne w czasie, do celów sterowania będziemy używać modelu dyskretnego - układu równań różnicowych, przy czym w chwili początkowej wartości zmiennych stanu są równe $x_0^{(1)}$, $x_0^{(2)}$, α_0 , ϕ_0 , p_0 .

Rozważamy sytuację, w której w danym zamkniętym obszarze z przeszkodami

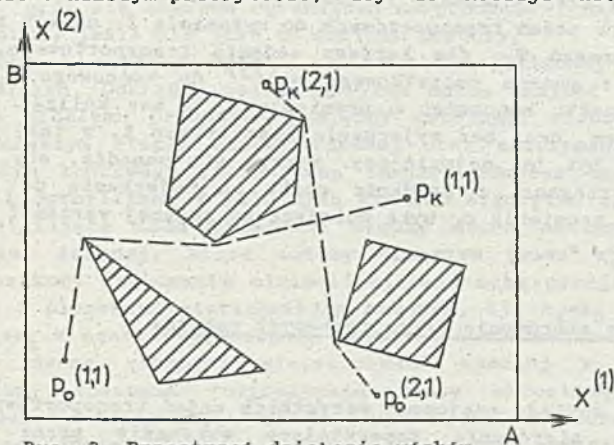
porusza się k_w wózków: w_1, w_2, \dots, w_{k_w} (rys. 2). Każdy z nich ma do wykonania szereg zadań transportowych, określonych przez punkt początkowy jazdy danego wózka p_0 , punkt docelowy jazdy p_k oraz czas rozpoczęcia jazdy t_0 . Oznaczmy przez z_{ij} j -te zadanie transportowe dla i -tego wózka i zdefiniujmy je jako

$$z_{ij} = \langle p_0^{(ij)}, p_k^{(ij)}, t_0^{(ij)}, s_{ij} \rangle,$$

gdzie:

s_{ij} - priorytet danego zadania.

Każdemu z zadań jest przyporządkowana dodatnia liczba całkowita s_{ij} , będąca jego priorytetem w stosunku do pozostałych. Jeśli w pewnej chwili nie będzie możliwe jednoczesne realizowanie wszystkich zadań transportowych, układ sterowania powinien wybrać do wykonania zadania w kolejności od najwyższego priorytetu. Oznacza to, że jeśli miałyby nastąpić kolizja dwóch wózków realizujących zadania transportowe o różnym priorytecie, układ sterowania powinien tak zmodyfikować trajektorię wózka wykonującego zadanie o niższym priorytecie, aby do kolizji nie doszło.



Rys. 2. Przestrzeń działania wózków
Fig.2. Car operation space

Zakładamy, że znamy z góry wszystkie zadania transportowe, przy czym w momencie początkowym wózek znajduje się w punkcie początkowym pierwszego zadania, jakie ma wykonywać (jeśli tak nie jest, należy zadać dodatkowe zadanie dla tego wózka, które będzie wykonywane jako pierwsze i będzie polegać na dojechaniu do punktu p_0). Zatem zadania transportowe, jakie mają do wykonania wózki, są opisane przez zbiór

$$Z = \{ z_{ij} \}, \quad i \in 1, 2, \dots, k_w.$$

Aby wykonać te zadania, wózki muszą przejechać od odpowiednich punktów początkowych do końcowych, omijając przeszkody, a także unikając kolizji z innymi wózkami znajdującymi się w rozważanym obszarze. Założymy, że kolizja następuje, jeśli odległość dwóch wózków znajdujących się w punktach x_1 i x_2 jest mniejsza od zadanej wartości $2r_w$ (r_w - odległość najdalszego punktu wózka od punktu, który określa współrzędne wózka z odpowiednim marginesem bezpieczeństwa), tzn. jeśli

$$\sqrt{(x_1^{(1)} - x_2^{(1)})^2 + (x_1^{(2)} - x_2^{(2)})^2} \leq 2r_w$$

Wózki poruszają się w prostokątnej przestrzeni X (rys. 2), w której znajduje się pewna liczba l_p przeszkód $P_i \in X$, gdzie i - numer przeszkody. Zakładamy, że przeszkody są wielokątami wypukłymi. Wówczas przeszkoda P_j może być opisana przez podanie zbioru jej wierzchołków. Zakładamy również, że przeszkody nie poruszają się.

Na mapę M składają się: opis przestrzeni X i opis znanych przeszkód P :

$$M = \langle X, P \rangle.$$

Oznaczmy przez U_{ij} ciąg sterowań u_n doprowadzający i -ty wózek od punktu początkowego p_0 do punktu końcowego p_k

$$U_{ij} = [u_1, u_2, \dots, u_{Nij}],$$

Ostatecznie zadanie sterowania grupą ruchomych robotów z uwzględnieniem unikania kolizji i dynamiki ich ruchu możemy zdefiniować następująco:

Zadanie sterowania.

Dla danej mapy obszaru M , w którym poruszają się roboty, modelu (uwzględniającego dynamikę) robotów (1), ich położeń początkowych oraz danego zbioru zadań transportowych do wykonania Z , należy znaleźć:

ciąg sterowań U_{ij} dla każdego zadania transportowego, który zapewni dojechanie z punktu początkowego $p_0^{(ij)}$ do końcowego $p_k^{(ij)}$ (tzn. do otoczenia punktu końcowego o promieniu ϵ_k) bez kolizji z przeszkodą lub innym wózkiem oraz bez wyjechania poza obszar X , w taki sposób, aby czas przejazdu T był jak najmniejszy. Wymaga się ponadto, aby priorytety zadań były przestrzegane, a prędkość wózka po dojechaniu do otoczenia punktu końcowego o promieniu ϵ_k była mniejsza od zadanej wartości ϵ_v , tzn.

$$p(T) \leq \epsilon_v.$$

2. Algorytmy sterowania grupą ruchomych robotów

Algorytm zakłada znajomość wszystkich zadań transportowych z_{ij} . Wyznacza on kolejno sterowania zapewniające wykonanie przez wózki zadań z uwzględnieniem momentów początkowych, od których może się rozpocząć wykonywanie zadania, oraz priorytetów zadań. Przy wyznaczaniu sterowania przyjmujemy, że kolizja następuje, jeśli położenia dwóch wózków różnią się mniej niż zadana wartość. Algorytm może przy tym przewidywać kolizję na kilka kroków napórzód, co zwiększa efektywność uzyskiwanego rozwiązania. Jeśli miałyby nastąpić kolizja, algorytm modyfikuje sterowanie wózkiem o niższym priorytecie w ten sposób, aby wyminąć wózek o wyższym priorytecie, uwzględniając jego kierunek jazdy i prędkość. Algorytm ma następującą postać:

Algorytm Φ

- Wybierz to z zadań, dla którego nie obliczono jeszcze sterowania, które ma największy priorytet. Oznaczmy to zadanie przez z_{ij} .
- Znajdź wózek, który ma realizować to zadanie i oblicz sterowanie U_{ij} , realizujące to zadanie, korzystając z algorytmu wyznaczania sterowania jazdą pojedynczego wózka Φ_p . Przy tym położenia, jakie mają wózki o wyznaczonych już trajektoriach, w określonych momentach czasu są traktowane w taki sam sposób, jak przeszkody stałe. Jeśli wyznaczone sterowanie miałoby doprowadzić do kolizji z innym wózkiem, którego trajektoria jest już znana, nie jest ono brane pod uwagę.

3. Powtarzaj punkty 1 i 2 aż do wyznaczenia sterowania sterowania dla wszystkich zadań transportowych do wykonania.

Algorytm Ψ_p ma za zadanie wyliczenie sterowania realizującego pojedyncze zadanie transportowe. Składa się on z dwóch kroków:

Krok 1. Polega na wyznaczeniu najkrótszej w sensie geometrycznym drogi łączącej punkt początkowy i końcowy, która omija przeszkody i wózki o już ustalonych trajektoriach. Droga ta przy przyjętych założeniach dotyczących przeszkód (reprezentowanych przez wielokąty) jest łamana o wierzchołkach w wierzchołkach przeszkód.

Krok 2. Znalazienie ciągów sterowań pozwalających w możliwie krótkim czasie przejechać między kolejnymi wierzchołkami łamanej. Przy tym jako stan początkowy przed przejechaniem kolejnego odcinka przyjmuje się stan, jaki został osiągnięty po zakończeniu przejazdu poprzedniego odcinka. Zakończenie przejazdu odcinka następuje po zbliżeniu się do punktu końcowego tego odcinka na odległość mniejszą od zadanej liczby. Zatem przyjmuje się pewną tolerancję położenia oraz nie nakłada ograniczeń na prędkość i kąty α i φ na końcu odcinka, jeśli nie jest to odcinek końcowy.

Realizacja kroku 1 została oparta na metodzie zbliżonej do metody podziałów i ograniczeń. Dokładny opis algorytmu można znaleźć w [2].

Do rozwiązania problemu przejazdu pomiędzy kolejnymi wierzchołkami wyznaczonej w pierwszym etapie metody łamanej oraz zatrzymania wózka w otoczeniu e_k punktu końcowego zastosowana została również zmodyfikowana metoda podziałów i ograniczeń. W kolejnych krokach algorytmu znajduje się sterowanie minimalizujące czas przejazdu między dwoma następującymi po sobie wierzchołkami łamanej, które możemy oznaczyć przez x_b i x_e . W pierwszym kroku szukamy sterowania minimalizującego czas przejazdu między punktem startowym i pierwszym wierzchołkiem łamanej, tj. $x_b = x_0$ i $x_e = x_1$. W ostatnim kroku $x_e = x_k$. W czasie wyznaczania sterowania minimalizującego czas przejazdu między dwoma zadanymi wierzchołkami łamanej w kroku $m+1$ rozpatrujemy zbiory sterowań rozpoczynające się określoną sekwencją sterowań o długości m . Mając dany wektor stanu wózka na początku jazdy, model wózka i sekwencję sterowań, możemy wyznaczyć położenie wózka $[x_m^{(1)}, x_m^{(2)}]$ uzyskane po wykonaniu danej sekwencji sterowań. Dla wszystkich tak określonych ciągów sterowań określamy oszacowanie q funkcji kryterialnej Q (tj. czasu przejazdu między kolejnymi wierzchołkami łamanej) jako

$$q = \Delta t_m + c \frac{\sqrt{(x_m^{(1)} - x_e^{(1)})^2 + (x_m^{(2)} - x_e^{(2)})^2}}{P_{\max}}$$

gdzie:

Δt_m - czas potrzebny na przejazd od punktu początkowego (wierzchołka łamanej) do punktu x_m (po m taktach sterowania), $\Delta t_m = m \Delta t$, Δt - długość taktu sterowania,

P_{\max} - maksymalna prędkość liniowa wózka,

$c \geq 1$ - współczynnik kary.

Dodanie współczynnika kary za odległość powoduje, że przy wyborze zbioru do podziału w kolejnym kroku przeszukiwania preferowane są trajektorie doprowadzające w pobliże punktu docelowego, nawet, jeśli rzeczywiste

oszacowanie czasu jest dla nich gorsze. Bez wprowadzenia współczynnika kary c metoda byłaby nieefektywna - przeszukiwany byłby zbyt obszerny zbiór sterowań i czas działania algorytmu byłby zbyt długi.

Dodatkowym problemem związanym z dynamiką wózka jest wymaganie osiągnięcia punktu docelowego jazdy $[x_k^{(1)}, x_k^{(2)}]$ o promieniu ϵ_k w ten sposób, aby końcowa prędkość była mniejsza od zadanej z góry wartości ϵ_v . Aby zapewnić spełnienie tego wymagania, po dojechaniu przez wózek w otoczenie punktu końcowego o promieniu ϵ_h , stosowana jest nowa postać q_1 funkcji kryterialnej:

$$q_1 = \Delta t_m + c \frac{\sqrt{(x_m^{(1)} - x_e^{(1)})^2 + (x_m^{(2)} - x_e^{(2)})^2}}{p_{\max}} + b \frac{p_m}{\sqrt{(x_m^{(1)} - x_e^{(1)})^2 + (x_m^{(2)} - x_e^{(2)})^2}}$$

gdzie:

p_m - liniowa prędkość wózka po m krokach sterowania,
 b - współczynnik kary za prędkość.

Tak więc algorytm w kolejnych krokach dokonuje podziału zbioru sterowań rozpoczynających się od określonej sekwencji $m+1$ elementowej. Jako nowy element sekwencji dla każdego nowego zbioru przyjmuje się jedno ze sterowań nie doprowadzających do kolizji z przeszłością. Danymi wejściowymi dla algorytmu są: kolejne wierzchołki łamanej x_0, x_1, \dots, x_k , wartość początkowa wektora stanu oraz model wózka.

Algorytm można zdefiniować następująco:

1. Podstaw $x_s = x_0$, $x_e = x_1$.
2. Jako oszacowanie funkcji kryterialnej przyjmujemy odległość między x_s i x_e .
3. Wyznaczamy zbiór wszystkich sterowań dopuszczalnych w danym kroku.
4. Wybieramy podzbiór o minimalnej wartości:
 - a) q , jeśli

$$\sqrt{(x_m^{(1)} - x_k^{(1)})^2 + (x_m^{(2)} - x_k^{(2)})^2} \geq \epsilon_h$$

b) q_1 w przypadku przeciwnym.

5. Jeśli ostatni punkt x_m osiągnięty przez wózek należy do otoczenia punktu x_k o promieniu ϵ_k , przejdź do 7.
6. Wybrany zbiór podziel na podzbiory sekwencji sterowań o długości $m+1$, z których pierwsze m jest identyczne jak w zbiorze wyjściowym. Dla tak uzyskanych podzbiorów wyznacz punkty x_{m+1} i oszacowania q lub q_1 . Przejdź do 3.
7. Jeśli $x_e = x_k$ i $p \leq \epsilon_v$, zakończ wykonywanie algorytmu.
 Jeśli $x_e = x_k$ i $p > \epsilon_v$, przejdź do 6.
8. Podstaw $x_s = x_e$, $x_e = x_{n+1}$ i przejdź do 2.

3. Omówienie wyników badań symulacyjnych przebiegu sterowania przy użyciu opisanych algorytmów

Jak wspomniano, za pomocą opracowanej metody otrzymuje się rozwiązanie suboptymalne. Miara straty optymalności wynikającej z przybliżonego rozwiązania drugiego etapu metody może być następujący współczynnik:

$$\eta = \frac{t_{\text{met}} - t_{\text{opt}}}{t_{\text{opt}}}$$

gdzie: t_{opt} - czas przejścia wg sterowania optymalnego,
 t_{met} - czas uzyskany przy stosowaniu sterowania
 wyznaczonego przez algorytm.

Niestety, sterowanie optymalne, a zatem i czas t_{opt} nie są znane. Współczynnik można jednak oszacować od góry, zamieniając czas t_{opt} przez jego dolne ograniczenie

$$t_{\text{opt}} = \frac{s}{p_{\text{max}}}$$

gdzie: s - długość najkrótszej geometrycznej drogi od punktu startowego do celu,

p_{max} - maksymalna prędkość, jaką może osiągać robot.

Obecnie przedstawimy spostrzeżenia odnośnie wyników otrzymanych dla przykładowych przypadków sterowania jednym wózkiem.

Wartość współczynnika w bardzo dużym stopniu zależy od konfiguracji przeszkód i położenia punktów startowego i końcowego. Przeprowadzono badania dla różnych konfiguracji przeszkód i otrzymano η w zakresie 0.01 - 0.8. Dla zadanej konfiguracji można stwierdzić pewną zależność współczynnika i czasu realizacji algorytmu wyznaczenia sterowania od długości taktu sterowania i doboru współczynnika kary c . Zwiększanie długości taktu sterowania powoduje zwiększenie wartości współczynnika, ale równocześnie zmniejsza się czas obliczeń. Przykładowe wyniki dla określonej konfiguracji przeszkód przedstawionej na rysunku 3 i stałej wartości pozostałych parametrów przedstawia tabela 2.1.

ε	:	0.07	0.20	0.7	0.9
Δt	:	0.1	0.3	1.0	1.5
η	:	0.06	0.07	0.38	0.79
t	:	37	21	15	13

Tab 2.1 Zależność współczynnika η i czasu realizacji algorytmu t od długości taktu sterowania Δt

Wartość pozostałych parametrów algorytmu była następująca: $a=0.5\text{m}$, $r=0.05\text{m}$, $V=0.35\text{Nm}$, $w=0.7\text{rd/s}$, $a=0.2\text{m}$, $k_f=0.4$, $c=5$, $\alpha_c=0$, $\varphi_c=0$. Zwiększanie taktu sterowania wymaga proporcjonalnego zwiększania ε . Jeśli takt sterowania jest duży, a ε mały, to znacznie wydłuża się czas obliczeń. Eksperymentalnie stwierdzono, że zależność pomiędzy taktem sterowania Δt a ε powinna jak

poprzednio, być następująca:

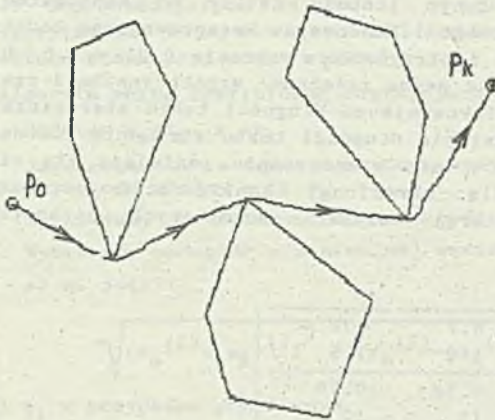
$$\varepsilon \approx \Delta t \cdot v.$$

Stwierdzono, że zmiana współczynnika kary ma niewielki wpływ na osiągnięty czas sterowania. Charakterystyczne jest, że przy zmniejszaniu współczynnika kary c poniżej 1.5 następuje gwałtowne zwiększenie czasu realizacji algorytmu powyżej wartości praktycznie akceptowalnych. Niekiedy okazuje się nawet, że zwiększenie współczynnika kary poprawia wynik. Przykładowe wyniki pokazuje tabela 2.2.

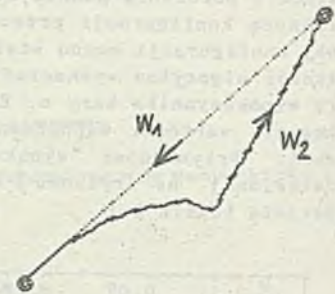
c	1.05	1.1	1.2	3	5
t_{met}	18.2	18.7	19.4	20.1	20.2
	0.12	0.12	0.26	0.28	0.28
t	215	97	53	34	34

Tab.2. Zależność czasu sterowania t_{met} i czasu realizacji algorytmu wyznaczania sterowania t od współczynnika kary c

Problemem związanym z dynamiką jest wybór takiego otoczenia punktu końcowego x_k , po osiągnięciu którego rozpoczyna się hamowanie robota tak, aby predkość końcowa była równa zero. Przyjęcie za dużej wielkości powoduje straty związane ze zbyt długim czasem jazdy i obliczeń, a zbyt małej prowadzi do przejechania poza punkt końcowy.



Rys.3. Wymijanie przeszkód stałych
Fig.3. Avoidance of fixed obstacles



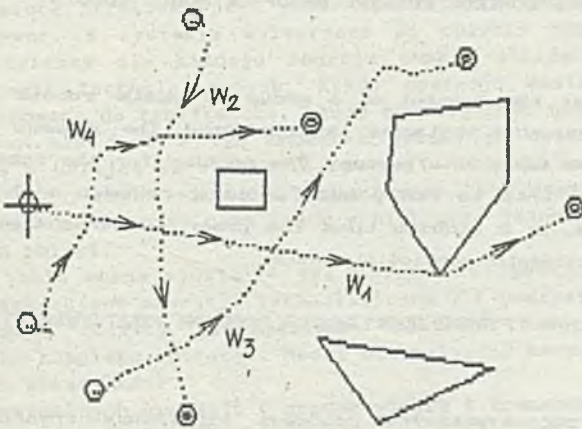
Rys.4. Unikanie zderzenia czołowego z innym wózkiem
Fig.4. Avoidance of head-on collision with another car

Obecnie przedstawione zostaną wnioski dotyczące sterowania grupą wielu robotów. Na rysunkach przedstawiono trajektorie robota dla dwóch charakterystycznych przypadków możliwości kolizji: pierwszy, kiedy roboty jedą dokładnie naprzeciw sobie i mogłyby się zderzyć czołowo, a drugi, gdy zbliżają się pod kątem zbliżonym do prostego. W porównaniu z algorytmem sterowania jednym robotem wydłuża się czas obliczeń, przy czym w pewnych

sytuacjach wydłużenie jest znaczne i stanowi problem. Jest to spowodowane faktem, że algorytm przeszukuje bardzo szerokie zbiory sterowań we wszystkich kierunkach, w tym w kierunkach, w których jedzie drugi robot, co prowadzi do następnej możliwości kolizji w jednym z najbliższych kroków. Z tego względu algorytm został uzupełniony w ten sposób, aby nie wszystkie kierunki uniknięcia kolizji były brane pod uwagę. Algorytm wylicza kąt między kierunkami jazdy robotów i zabronione są sterowania prowadzące poza najkorzystniejszy kierunek unikania kolizji. W ten sposób uzyskano lepsze unikanie kolizji przy jednoczesnym zmniejszeniu czasu obliczeń. Algorytm unikania kolizji można przy tym próbować dalej udoskonalić tak, aby uzyskać bardziej płynny przebieg trajektorii przy omijaniu drugiego robota, niż przebiegi przedstawione na rysunkach. Jest to zadanie do rozwiązania w dalszych pracach.

Wymijanie innych robotów powoduje, że długość trajektorii robota L_1 dla jazdy od punktu początkowego do końcowego jest dłuższa (czasami znacznie) niż długość trajektorii L dla jazdy pojedynczego robota. Wydłużenie to silnie zależy od układu punktów początkowych, końcowych, przeszkód stałych i ilości robotów.

Drugim podstawowym problemem, jaki pojawia się przy sterowaniu grupą wózków, jest problem znalezienia efektywnego algorytmu heurystycznego przeszukiwania drzewa rozwiązań. Algorytm taki powinien zapewnić możliwość wycofania się z drogi, która mimo długotrwałych poszukiwań nie daje polepszenia rezultatu. Także ten algorytm będzie doskonalony w dalszych pracach. Ma on podstawowe znaczenie, ponieważ jego polepszenie pozwala na zmniejszenie czasu obliczeń oraz wymagań co do wielkości pamięci komputera realizującego algorytm.



Rys. 5. Unikanie kolizji przez grupę ruchomych robotów.

Fig.5. Avoidance of collisions in a group of moving robots

Podsumowując uzyskane wyniki, opracowany algorytm pozwala na sterowanie grupą ruchomych robotów z unikaniem kolizji w obecności przeszkód, przy

czym można jeszcze poprawić jakość sterowania ulepszając rozwiązanie problemu wymijania robotów.

LITERATURA

- [1] Bubnicki Z., Józefczyk J., Podstawy adaptacyjnych systemów sterowania ruchem w zastosowaniu do sterowania maszyn, pojazdów i manipulatorów. Raporty ISTS, seria SPR, nr 17/87, Politechnika Wrocławska, Wrocław 1987.
- [2] Martowski K., Pusz G., Sas J., Podstawy adaptacyjnych systemów sterowania w zastosowaniu do sterowania maszyn, pojazdów i manipulatorów, Raporty ISTS, seria SPR, nr 18/88, Wrocław 1988.
- [3] Chalam V.V., Adaptive Control Systems - Techniques and Applications, Marcel Dekker Inc., New York and Basel, 1987.
- [4] Vincent T.L., Stricht D.L., Peng W.Y., Aircraft Missile Avoidance, Journal of Operations Research, 4, 1976.
- [5] Lee C.S.G., Lee B.H., Resolved Motion Adaptive Control for Mechanical Manipulators, Trans. of the ASME, vol. 106, June 1984.

Recenzent: Prof.dr inż.H.Kowalowski

Wpłynęło do Redakcji do 1990-04-30.

COLLISION AVOIDANCE CONTROL FOR THE GROUP OF MOBILE ROBOTS

Summary

In the paper the control of a group of mobile robots moving in the constrained space with obstacles is considered. The dynamic properties of robots motion are taken into account. The problem for the robots is to move from the start points to end points, avoiding collision with the obstacles and other robots, in a minimum time. The paper describes the properties of the worked out heuristic control algorithm.

АЛГОРИТМ УПРАВЛЕНИЯ ГРУППОЙ ДВИЖУЩИХСЯ РОБОТОВ ИЗБЕГАЮЩИЙ СТОЛКНОВЕНИЙ

Резюме

В работе рассматривается проблема управления группой движущихся роботов в ограниченном пространстве, в котором находятся известные препятствия для движения. Учитываются динамические свойства движения роботов. Задача для роботов состоит в перемещении от начальной точки до конечной, избегая столкновений с препятствиями и другими роботами в минимальное время. В работе описываются свойства разработанного алгоритма управления роботами.