

Piotr SKRZYPCZYŃSKI

Politechnika Poznańska, Katedra Automatyki, Robotyki i Informatyki

## METODY WYZNACZANIA ŚCIEŻKI ROBOTA MOBILNEGO OPARTE NA PODZIALE PRZESTRZENI

**Streszczenie.** Praca prezentuje rozwiązanie zadania planowania ścieżki dla autonomicznego robota mobilnego oparte na podziale przestrzeni. Celem pracy jest przedstawienie algorytmów planowania i wygładzania ścieżki dających w efekcie jak najkrótszą ścieżkę, przy zapewnieniu bezpiecznego przejazdu robota w przypadku niedokładności modelu otoczenia. Planowanie ścieżki odbywa się na podstawie geometrycznego modelu otoczenia. Przedstawiono wyniki badań symulacyjnych.

## METHODS OF A MOBILE ROBOT PATH COMPUTING BASED ON EXACT CELL DECOMPOSITION

**Summary.** This work presents a solution of the path planning problem for an autonomous mobile robot. Path planning algorithms based on exact cell decomposition method and the geometrical world model are described. Then the path smoothing process is discussed. Some results of simulation experiments are given.

### 1. Wprowadzenie

#### 1.1. *Planer ścieżki w systemie nawigacji robota mobilnego*

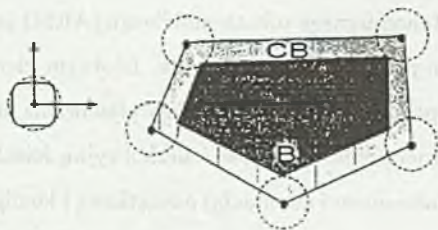
Podstawową własnością autonomicznego robota mobilnego (ARM) jest możliwość samodzielnego tworzenia i wykonywania planów działania. Istotnym elementem systemu nawigacji takiego robota jest podsystem automatycznego planowania ścieżki. Pożądane jest, by ARM był w stanie samodzielnie zaplanować bezkolizyjną ścieżkę przejazdu pomiędzy zadaną konfiguracją (położeniem i orientacją) początkową i konfiguracją końcową, lub orzekać o niemożności znalezienia takiej ścieżki. System planujący powinien znajdować ścieżkę optymalną ze względu na założone kryterium optymalizacji (zazwyczaj jest nim minimalna długość). Ścieżka planowana jest na podstawie posiadanego przez system nawigacji robota modelu otoczenia, zwanego mapą. Model taki może być predefiniowany lub budowany na podstawie danych dostarczonych przez systemy sensoryczne ARM. Mapa

zbudowana automatycznie na podstawie danych sensorycznych może być obciążona błędami wynikającymi z niedokładności systemu sensorycznego, w związku z czym planer ścieżki powinien brać pod uwagę możliwość występowania błędów w modelu otoczenia. Planowanie ścieżki może następować na podstawie znajomości całego modelu otoczenia (metody globalne) lub jedynie jego fragmentu wokół aktualnej konfiguracji robota (metody lokalne).

Powyższe założenia wyznaczają główne zagadnienia, których rozwiązaniu poświęcona jest niniejsza praca. Dodatkowym założeniem jest oparcie planowania ścieżki na modelu otoczenia w postaci mapy geometrycznej. Mapa ta otrzymywana jest na podstawie informacji sensorycznej i może być obciążona znaczną niepewnością. Czyni to szczególnie istotnym problem znajdowania ścieżki *bezpiecznej*, zapewniającej przejazd robota z dala od przeszkody, nawet w wypadku wystąpienia niedokładności w modelu otoczenia.

### 1.2. Model robota

W celu uproszczenia rozważań przyjęto, że robot reprezentowany jest przez punkt, a przeszkody w modelu otoczenia zostają przekształcone przez powiększenie o wymiary robota. Sprowadza to problem planowania ruchu dwuwymiarowego obiektu do planowania ruchu punktu [2]. Przestrzeń określona przez wszystkie możliwe konfiguracje robota oznaczana jest przez  $C$ , a powiększone przeszkody w tej przestrzeni —  $CB_i$ . Wolna przestrzeń konfiguracyjna zdefiniowana jest jako  $C_f = C - \bigcup_{i=1}^q CB_i$ , gdzie  $q$  jest liczbą przeszkód. Zakłada się, że przestrzeń konfiguracyjna  $C$  jest dwuwymiarowa, a przeszkody są wielobokami (niekoniecznie wypukłymi). Powyższe uproszczenie uzasadnione jest docelowym



Rys. 1. Powiększanie przeszkody

Fig. 1. Obstacle enlarging

przeznaczeniem prezentowanych metod planowania ścieżki dla robota *TRC LABMATE*. Platformę tego robota łatwo jest przybliżyć okręgiem, a układ jezdny daje możliwość

zawrócenia w miejscu. W związku z tym przeszkody są powiększane o promień okręgu opisanego na robocie, a planowanie ruchu robota względem przeszkody  $B$  zastąpiono planowaniem ruchu środka układu współrzędnych związanego z robotem względem powiększonej przeszkody  $CB$  (rys. 1).

### 1.3. Model otoczenia

Model otoczenia tworzony i przechowywany jest w postaci dwuwymiarowej (2D) mapy geometrycznej. Mapa ta budowana jest na podstawie danych uzyskiwanych ze skanera optycznego, w który wyposażony jest ARM. W geometrycznej reprezentacji otoczenia każdy wykryty przez sensory obiekt posiada odwzorowanie w mapie w postaci wieloboku lub łamanej. Elementy otoczenia, których nie udaje się sklasyfikować jako obiekty o określonym kształcie, są opisywane jako zajęte przez przeszkody-prostokąty. Takie podejście do zbierania informacji o środowisku robota charakteryzuje się wieloma zaletami, takimi jak niewielka ilość pamięci potrzebna do przechowywania mapy oraz łatwość aktualizowania mapy w wypadku stwierdzenia błędów pozycjonowania robota [3].

Każda przeszkoda w mapie posiada przypisany argument świadczący o jakości odwzorowania, tj. prawdopodobieństwo, że jej obrys jest prawidłowy. Prawdopodobieństwo to waha się od bardzo małego dla przeszkód, których nie udało się sklasyfikować jako wieloboków lub łamanych do równego jedności dla przeszkód wprowadzonych do mapy na podstawie ręcznych pomiarów otoczenia (przeszkód predefiniowanych). Pozostałe przeszkody posiadają współczynniki jakości obliczone automatycznie w czasie tworzenia mapy, a zależne od liczby zeskanowanych punktów, które utworzyły daną przeszkodę, oraz tego, czy przeszkoda ta powtarza się w kolejnych skanach (pomiarach wykonanych podczas jednego obrotu głowicy skanera). Współczynnik jakości odwzorowania przeszkody uwzględniany jest w procesie planowania ścieżki na etapie powiększania przeszkód. Każda przeszkoda powiększana jest o promień okręgu opisanego na robocie modyfikowany współczynnikiem jej jakości.

### 1.4. Wybór metody planowania ścieżki

Z literatury znanych jest kilka grup algorytmów najczęściej stosowanych do znajdowania bezkolizyjnych ścieżek [2]. Spośród wielu metod postępowania wyróżnić można następujące :



- Metoda potencjałów.
- Metody typu mapy przejść.
- Metody dekompozycji przestrzeni.

Wybór metody został dokonany w kontekście przyjętych na wstępie założeń. W związku z zastosowaniem mapy geometrycznej jako modelu otoczenia wybrany algorytm powinien działać bezpośrednio w oparciu o mapę w formie geometrycznej. Założenia takiego nie spełniają algorytmy wymagające rasteryzacji modelu otoczenia, takie jak metoda potencjałów czy przybliżona metoda dekompozycji przestrzeni. Metodami, które można zaimplementować bezpośrednio na podstawie danych o przeszkodach w mapie geometrycznej, są metody typu mapa przejść, np. metoda grafu widoczności. Charakterystyczną cechą tej metody jest jednak znajdowanie ścieżki, która styka się z wierzchołkami przeszkód. Jest to niekorzystne w sytuacji, gdy chcemy otrzymać bezpieczną ścieżkę, opierając się na modelu środowiska potencjalnie obciążonym błędami. Wobec powyższych ograniczeń spośród znanych metod planowania ścieżki wybrano metodę *dokładnej dekompozycji przestrzeni*. Metoda ta umożliwia planowanie ścieżki na podstawie mapy geometrycznej, zapewniając przy tym, że znaleziona ścieżka będzie przebiegać możliwie daleko od przeszkód. Metoda dokładnej dekompozycji przestrzeni ma charakter globalny, ścieżka generowana jest na podstawie znajomości kompletnego modelu otoczenia. Istnieje jednak możliwość wykorzystania jej w planerach działających w czasie ruchu robota (*on-line*) poprzez wykorzystanie jedynie znanego w danej chwili podzbioru przestrzeni konfiguracyjnej wokół aktualnej konfiguracji robota. Ścieżka jest wówczas tworzona poprzez łączenie fragmentów ścieżek pozwalających na osiągnięcie pośrednich konfiguracji robota.

## 2. Metoda dokładnej dekompozycji przestrzeni

Metody dekompozycji przestrzeni polegają na zdekomponowaniu wolnej przestrzeni wokół robota na proste obszary, zwane komórkami, w taki sposób, że łatwo jest poprowadzić ścieżkę pomiędzy dwoma dowolnymi konfiguracjami robota w obrębie danej komórki. Metody te można podzielić na dwie grupy — metody dokładne i przybliżone. Metoda przybliżonej dekompozycji przestrzeni zakłada rasteryzację modelu otoczenia, w związku z czym nie będzie tu rozpatrywana.

W metodzie dokładnej dekompozycji przestrzeni przestrzeń  $C_f$  dekomponowana jest na zbiór nie zachodzących na siebie komórek, których suma daje dokładnie  $C_f$ . Następnie tworzy się graf połączeń, który reprezentuje relacje sąsiedztwa między komórkami. Graf ten jest przeszukiwany. W przypadku sukcesu rezultatem przeszukania jest sekwencja komórek, zwana *kanalem* łącząca komórki zawierające konfigurację początkową  $q_s$  i docelową  $q_d$  robota. Ścieżka tworzona jest na podstawie tej sekwencji.

Nie wszystkie dekompozycje przestrzeni są odpowiednie. Komórki generowane podczas dekompozycji powinny się charakteryzować następującymi cechami :

- Geometria komórki powinna być wystarczająco prosta, aby móc wyznaczyć ścieżkę między dwoma dowolnymi konfiguracjami w obrębie komórki.
- Sprawdzenie sąsiedztwa dowolnych dwóch komórek i znalezienie ścieżki przecinającej część granicy wspólnej dla dwóch sąsiadujących komórek nie powinno być trudne.

Dekompozycja wolnej przestrzeni  $C_f$  i odpowiadający jej graf połączeń są zdefiniowane następująco [2] :

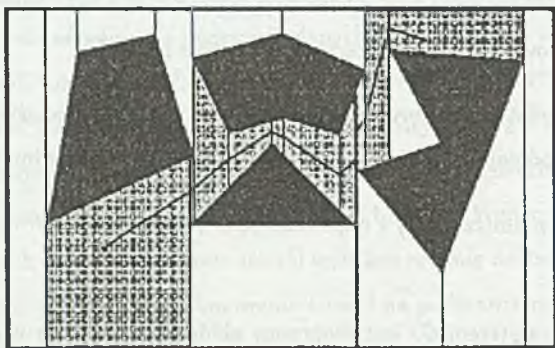
1. Dekompozycją przestrzeni  $C_f$  jest skończony zbiór  $K$  wypukłych wielokątów zwanych komórkami, takich że wnętrza dowolnych dwóch komórek nie przecinają się i suma wszystkich komórek jest równa  $C_f$ . Dwie komórki  $k$  i  $k'$  w  $K$  sąsiadują ze sobą wtedy i tylko wtedy, gdy  $k \cap k'$  jest segmentem liniowym o niezerowej długości.
2. Graf połączeń  $G$  związany z dekompozycją wypukłych wielokątów  $K$  przestrzeni  $C_f$  jest nieskierowanym grafem określonym następująco :
  - Węzłami  $G$  są komórki w  $K$ .
  - Dwa węzły w  $G$  są połączone połączeniem wtedy i tylko wtedy, gdy korespondujące komórki sąsiadują ze sobą.

Ogólny algorytm metody dokładnej dekompozycji przestrzeni przedstawia się następująco :

- 1 — Wygeneruj dekompozycję wypukłych wielokątów  $K$  przestrzeni  $C_f$ .
- 2 — Skonstruuj graf połączeń  $G$  związany z dekompozycją  $K$ .

- 3 — Przeszukaj  $G$  w celu znalezienia sekwencji sąsiadujących komórek między  $q_s$  i  $q_g$ .
- 4 — Jeżeli przeszukanie skończyło się sukcesem, zwróć wygenerowaną sekwencję komórek, w przeciwnym wypadku zwróć znacznik niepowodzenia.

Wynikiem działania algorytmu jest sekwencja komórek  $k_1, \dots, k_p$ , takich że  $q_s \in k_1$  i  $q_g \in k_p$  i dla każdego  $j \in [1, p-1]$ ,  $k_j$  i  $k_{j+1}$  sąsiadują ze sobą. Wnętrze kanału leży całkowicie we wnętrzu wolnego obszaru. Najprostszym sposobem wygenerowania wolnej ścieżki zawartej we wnętrzu kanału otrzymanego w wyniku przeszukania grafu  $G$  jest wyznaczenie punktów środkowych każdego segmentu i połączenie  $q_s$  z  $q_g$  linią łamaną.



Rys. 2. Ścieżka wyznaczona metodą dokładnej dekompozycji przestrzeni  
Fig. 2. Path computed with exact cell decomposition method

Optymalna wypukła dekompozycja jest możliwa do wyznaczenia w czasie wielomianowo zależnym od liczby wierzchołków  $n$ , pod warunkiem że przeszkody są wypukłymi wielobokami [2]. Istnieje możliwość dokonania dekompozycji nieoptymalnej, która może być wykonana w sposób efektywny bez nakładania takich ograniczeń na model otoczenia [1]. Nieoptymalna dekompozycja przestrzeni  $C_f$  tworzona jest przez przemiatanie przestrzeni  $C_f$  prostą równoległą do jednej z osi układu współrzędnych (np. osi  $y$ ). Proces przemiatania jest przerywany, kiedy prosta napotyka wierzchołek obszaru  $CB$ . Maksymalnie tworzone są dwa pionowe segmenty liniowe łączące ten wierzchołek z krawędziami  $CB$ , które znajdują się bezpośrednio nad i bezpośrednio pod nim. Granica  $CB$  i utworzone liniowe segmenty wyznaczają trapezoidalną dekompozycję  $C_f$ . Każda komórka dekompozycji jest trapezoidem lub trójkątem.

Przeszukiwanie grafu  $G$  jest realizowane za pomocą algorytmu  $A^*$  [2] (znanego też jako algorytm Dijkstry [4]). Węzłami grafu są  $q_s$ ,  $q_g$  oraz punkty środkowe granic komórek two-



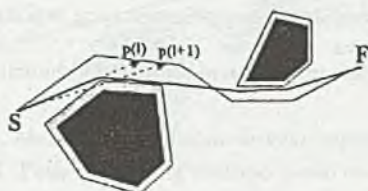
rzających kanał. Dwa węzły są połączone gałęzią tylko wtedy, gdy można poprowadzić między nimi odcinek całkowicie zawarty w komórce. Połączenia są rozpatrywane ze względu na długość odcinków łączących dwa węzły. Złożoność obliczeniowa przeszukiwania wyraża się zależnością  $O(n \log n)$ . Przykładową ścieżkę otrzymaną metodą dekompozycji przestrzeni przedstawia rys. 2.

### 3. Propozycje modyfikacji przedstawionej metody

Jak zaprezentowano to powyżej, metoda dokładnej dekompozycji przestrzeni znajduje ścieżkę ARM zgodnie z przyjętymi na wstępie założeniami. Niestety, znaleziona ścieżka jest często daleka od optymalnej w sensie najkrótszej możliwej drogi między konfiguracją początkową i końcową. Wynika to głównie z uproszczonej metody prowadzenia ścieżki robota przez znaleziony kanał. Przeprowadzone symulacje wykazały (rys. 2), że metoda dekompozycji przestrzeni prowadzić może do niepotrzebnego „nakładania drogi” przez robota. W celu wyeliminowania tej istotnej niedogodności zaproponowano metody wygładzania znalezionej ścieżki. Celem proponowanych rozwiązań jest znalezienie nowej ścieżki, która zachowywałaby zalety oryginalnej ścieżki otrzymanej metodą dekompozycji przestrzeni, będąc jednocześnie krótszą.

#### 3.1. Wygładzanie ścieżki metodą Tchouchenkova

Jest to metoda wygładzania [5] zastępująca fragmenty znalezionej ścieżki najdłuższymi odcinkami możliwymi do poprowadzenia przy danej konfiguracji przeszkód (rys. 3). Algorytm ten jest częścią opracowanej przez Tchouchenkova metody znajdowania ścieżki bazującej na mapie rastrowej, może być jednak łatwo przeniesiony na mapę geometryczną. Zapewnia ona znalezienie najkrótszej ścieżki leżącej w danym kanale przejścia.

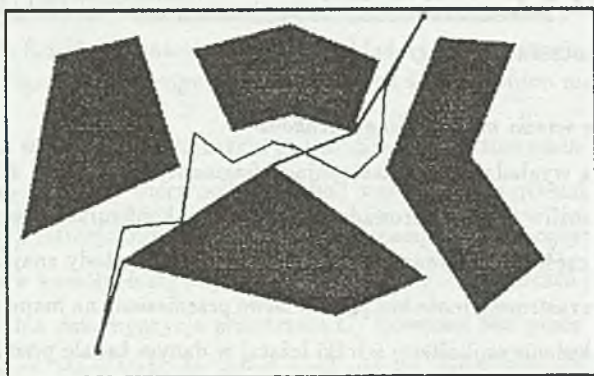


Rys. 3. Wygładzanie ścieżki metodą Tchouchenkova  
Fig. 3. Path smoothing with Tchouchenkova's method

Algorytm można przedstawić następująco :

- 1 — Z punktu początkowego  $S$  będącego punktem pierwotnej ścieżki są odkładane odcinki, które kończą się w dowolnym punkcie  $P_i$  ścieżki. Jeżeli ten odcinek przechodzi przez przeszkodę, próbuje się połączyć z punktem  $S$  poprzedni punkt ścieżki  $P_{i-1}$ .
- 2 — Jeżeli utworzony odcinek do punktu  $P_i$  jest wolny od kolizji, a odcinek poprowadzony do punktu  $P_{i+1}$  przechodzi przez przeszkodę lub jej dotyka, odcinek  $SP_i$  wybierany jest jako część nowej, wygładzonej ścieżki.
- 3 — Punkt  $P_i$  staje się punktem początkowym. Jeżeli nowy punkt początkowy jest równy punktowi docelowemu  $F$ , algorytm kończy działanie, w przeciwnym wypadku wraca do początku.

Jak wykazały badania symulacyjne (rys. 4), ścieżka wygładzona za pomocą tego algorytmu jest podobna do uzyskanej metodą grafu widoczności, tzn. styka się z wierzchołkami przeszkód. Jest to sprzeczne z przedstawionym założeniem o poszukiwaniu bezpiecznej ścieżki. Tak więc pomimo skrócenia wynikowej ścieżki zastosowania tego algorytmu nie można uznać za rozwiązanie docelowe.



Rys. 4. Ścieżka wyznaczona metodą dokładnej dekompozycji przestrzeni i wygładzona algorytmem Tchouchenkova

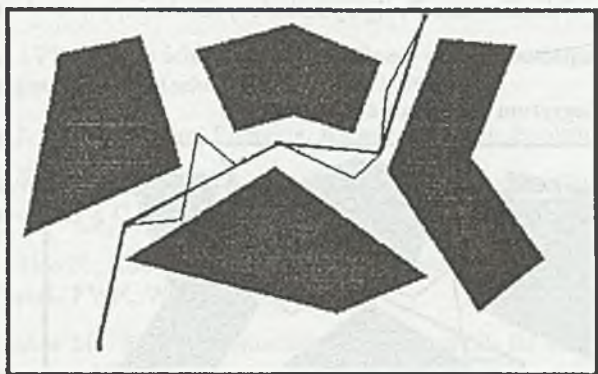
Fig. 4. Path computed with cell decomposition method and Tchouchenkov's smoothing algorithm

### 3.2. Metoda hybrydowa

Ponieważ zastosowana metoda wygładzania ścieżki nie przyniosła akceptowalnych wyników, konieczne stało się zaproponowanie innych rozwiązań. Ścieżka uzyskana wskutek



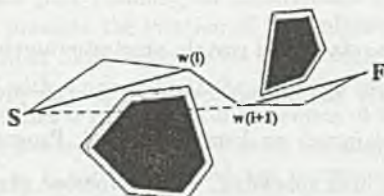
realizacji metody dekompozycji przestrzeni nie jest najkrótszą z możliwych, może być jednak uznana za bezpieczną. Natomiast ścieżka wygładzona jest krótsza, choć nie spełnia wymogu bezpieczeństwa. Wobec tego zrealizowano metodę tworzącą ścieżkę hybrydową, mającą w możliwie dużym stopniu łączyć korzystne cechy obu prezentowanych rozwiązań. Algorytm wykorzystuje jako bazę ścieżkę wygładzoną i znajduje dla każdego jej węzła najbliższy węzeł ścieżki pierwotnej, leżący w bezpiecznej odległości od przeszkody. Z tak wyznaczonych punktów powstaje nowa ścieżka (rys. 5).



Rys. 5. Ścieżka hybrydowa  
Fig. 5. Hybrid path

### 3.3. Metoda redukcji węzłów

Algorytm bazuje na pierwotnej ścieżce otrzymanej metodą dekompozycji przestrzeni. Znajduje on najkrótsze przejście pomiędzy węzłami ścieżki pierwotnej, redukując liczbę węzłów. Algorytm ten jest podobny do metody Tchouchenkova, brane są jednak pod uwagę tylko węzły ścieżki pierwotnej, a nie wszystkie jej punkty (rys. 6).



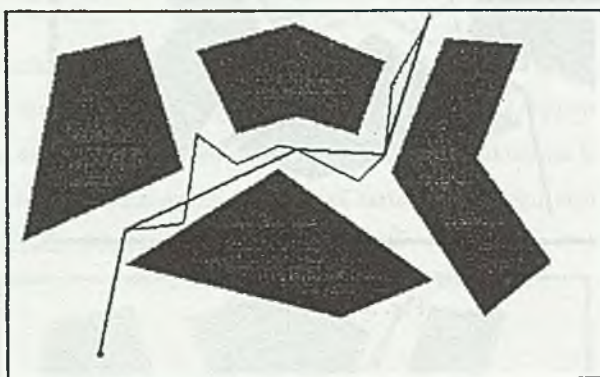
Rys. 6. Metoda wygładzania ścieżki poprzez redukcję węzłów  
Fig. 6. Path smoothing method based on reduction of nodes

1 — Z punktu początkowego  $S$  będącego węzłem pierwotnej ścieżki prowadzone są odcinki, które kończą się w dowolnym węźle  $W_i$  ścieżki pierwotnej. Jeżeli ten odcinek

przechodzi przez przeszkodę, próbuje się połączyć z punktem  $S$  poprzedni węzeł ścieżki  $W_{i-1}$ .

- 2 — Jeżeli utworzony odcinek do punktu  $W_i$  jest wolny od kolizji, a odcinek poprowadzony do punktu  $W_{i+1}$  przechodzi przez przeszkodę lub jej dotyka, odcinek  $SW_i$  wybierany jest jako część nowej, wygładzonej ścieżki.
- 3 — Punkt  $W_i$  staje się punktem początkowym. Jeżeli nowy punkt początkowy jest równy punktowi docelowemu  $F$ , algorytm kończy działanie, w przeciwnym wypadku wraca do początku.

Wyniki działania algorytmu przedstawia rys. 7.



Rys. 7. Ścieżka wygładzona metodą redukcji węzłów  
Fig. 7. Path computed with node reduction smoothing method

#### 4. Wnioski

Przedstawione metody planowania ścieżki zostały przetestowane na drodze symulacji komputerowych. Symulacje zostały zaimplementowane w postaci programów komputerowych w języku PASCAL działających na komputerze PC. Programy ten pozwalają prześledzić działanie algorytmów oraz sprawdzić, na ile tworzone przez nie ścieżki spełniają przyjęte na wstępie założenia. Zaprezentowane algorytmy pozwalają na znalezienie ścieżki ARM suboptymalnej w sensie długości, lecz charakteryzującej się bezpiecznym przebiegiem z dala od przeszkód. Zapropozowane metody wygładzania — metoda hybrydowa i redukcji węzłów znajdują ścieżkę bezpieczną, krótszą niż otrzymana bezpośrednio

przy zastosowaniu metody dekompozycji przestrzeni. Algorytm redukcji węzłów posiada jednak znacznie mniejszy koszt obliczeniowy, nie trzeba w tym wypadku przeprowadzać pełnego wygładzania ścieżki. Metoda dekompozycji przestrzeni oraz jej zaproponowana modyfikacja charakteryzują się niewielkim nakładem obliczeń. Uzasadnia to zastosowanie ich w systemie ARM pracującym w czasie rzeczywistym.

## LITERATURA

1. Gorzeń J. : Planowanie ścieżki robota mobilnego na podstawie mapy geometrycznej. Praca magisterska, Politechnika Poznańska, 1995.
2. Latombe J. : Robot Motion Planning. Kluwer Academic Publishers 1991.
3. Skrzypczyński P. : Building Geometrical Map of Environment Using IR Range Finder Data. Proc. IAS 4 Conf. Karlsruhe 1995.
4. Sysło M., Deo N., Kowalik J. : Algorytmy optymalizacji dyskretnej z programami w języku Pascal. PWN, Warszawa 1995.
5. Tchouchenkov M. : Eine effeziente Berechnungsmethode für die kürzeste Trajektorie Robotersysteme 1991.

Recenzent: Prof. dr inż. Henryk Kowalowski

Wpłynęło do Redakcji do 30.06.1996 r.

## Abstract

In this paper the path planning for an autonomous mobile robot (AMR) is considered. First chapter presents the location of path planner in the whole navigation system of AMR and describes basic assumptions about vehicle and environment model. Robot vehicle is replaced with a point, while the obstacles are enlarged, so the movement of the robot towards obstacles is replaced with movement of the point towards enlarged obstacles. The geometric (vector) map is used as an environment model. This map is obtained from sensor data, so it can have significant uncertainties. By this reason the planned path must be safe, it should be located as far from obstacles as possible. The choice of path planning method in context of previous assumptions is shown. In second chapter the chosen method i.e. exact cell decomposition is described in details. The third chapter is devoted to the methods of path smoothing. At first, the use of well-known Tchouchenkov's method is described. From the experimental study it has been found that the resulting path obtained with this method does not fulfil the above mentioned criteria of safety. By



this reason two other algorithms are proposed. Both methods compute safe path, though shorter than these obtained with exact cell decomposition method. The first is a hybrid method, which computes new path on the basis of these obtained with Tchouchenkov's algorithm, and ensures that the path is safe. The second method is node reduction algorithm. Both methods are illustrated with results of simulation experiments.



Fig. 1. Path in a 2D environment with obstacles.

Abstract

In this paper the algorithm of the safe path in a 2D environment with obstacles is presented. The algorithm is based on the exact cell decomposition method (ECD) in which the environment is decomposed into cells. The path is found by searching for a sequence of cells from the start cell to the goal cell. The path is then smoothed to avoid sharp turns. The algorithm is illustrated with results of simulation experiments.