

Sterownik ruchu robota mobilnego dla częściowo znanej przestrzeni roboczej

Leszek Podśędkowski

Leszek Kwapisz

Marek Idzikowski

Istvan Vizvary

Przedstawiono opis systemu sterowania robotem mobilnym, składający się z modułu interfejsu użytkownika, generatora ścieżki, pilota i skanera. Przedstawiono strukturę sterownika ruchu, algorytmy postępowania i wyniki badań doświadczalnych na rzeczywistym robocie dotyczące generatora ścieżki i skanera.

W Instytucie Obrabiarek i TBM PŁ od 1994 r. są prowadzone badania, których celem jest stworzenie szybkiej metody planowania optymalnej trajektorii robotów mobilnych uwzględniającej ich ograniczenia nieholonomiczne oraz ciągły napływ danych z czujników uaktualniających mapy przestrzeni roboczej. Opracowano metodę generowania ścieżki przy użyciu grafu skoków jednostkowych, algorytmu A^* i mapy kosztu heurystycznego [1], następnie opracowano algorytm korygowania ścieżki w chwili wykrycia zmiany układu przeszkód [2, 3]. Opracowane metody zostały zaimplementowane i został utworzony sterownik ruchu.

W maju br. zakończono weryfikację doświadczalną metody na rzeczywistym robocie. Prezentowany system jest jednym z najszybszych tego rodzaju na świecie.

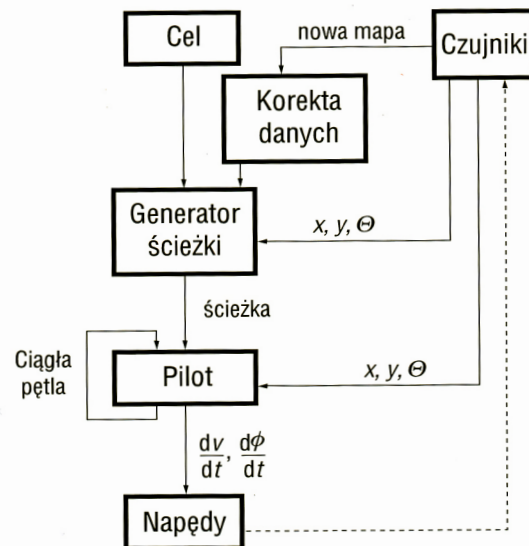
Sterownik ruchu

Opracowany w IOiTBM sterownik ruchu robota mobilnego spełnia następujące założenia:

- planuje ścieżkę uwzględniając więzy nieholonomiczne robota,
- wyznacza ścieżkę optymalną z uwagi na założone kryteria,
- działa w dowolnie skomplikowanej przestrzeni roboczej z elementami typu ślepe zaułki, korytarze, labirynty,
- czas planowania ścieżki jest relatywnie krótki,
- czas korekty ścieżki w przypadku wykrycia zmian w przestrzeni roboczej jest wystarczająco krótki, aby było można sterować robotem w czasie rzeczywistym bez konieczności zatrzymywania robota.

Aby zrealizować założenia ze sterownika ruchu zostały wydzielone dwa poziomy pracujące równolegle: generator ścieżki i pilot (rys. 1).

Podział sterownika ruchu na generator ścieżki i pilota jest analogiczny do tego, jaki występuje u kierowcy prowadzącego samochód. Generator ścieżki jest odpowiednikiem świadomego procesu myślenia, w którym jest planowana cała droga samochodu. Pilot jest odpowiednikiem odruchów. Pierwszy z tych odruchów to taka praca pe-



Rys. 1. Inteligentny sterownik ruchu

dałami i kierownicą, aby realizować drogę świadomie wcześniej wymyśloną. Drugi odruch to odruch awaryjny: jeśli nagle ktoś wybiegnie przed samochód lub za zakrętem będą roboty drogowe, to należy szybko zahamować lub skrócić w taki sposób, by uniknąć kolizji. Ten drugi odruch jest wykorzystywany tylko wtedy, gdy nie ma czasu na świadome myślenie, w przeciwnym wypadku kierowca obmyśla nową trasę i kieruje na nią samochód.

Jak wynika z tego podziału generator ścieżki jest odpowiedzialny nie tylko za pierwsze zaplanowanie ścieżki, ale także za jej powtórne planowanie przy zmianach przestrzeni roboczej. Zwłaszcza konieczność powtórnego planowania stawia przed generatorem ścieżki wymagania małej czasochłonności.

W proponowanym tu generatorze ścieżki problem poszukiwania ścieżki jest rozwiązywany najpierw bez uwzględniania ograniczeń nieholonomicznych, ale wynikiem poszukiwania nie jest jedna konkretna ścieżka, lecz mapa będąca jakby zbiorem drogowskazów w jakich kierunkach należy szukać drogi do celu. Mapa ta, nazywana dalej mapą kosztu heurystycznego, jest tworzona metodą propagacji szesnastokierunkowej (metoda dyfuzyjna). Drugim etapem jest tworzenie grafu dróg, którego wierzchołkami są konfiguracje robota, a krawędziami krótkie fragmenty drogi robota po łuku lub wzdłuż prostej.

Dzięki wprowadzeniu mapy kosztu heurystycznego tworzony graf ma znacznie mniej węzłów, a co za tym

Prof. dr inż. Leszek Kwapisz, dr hab. inż. Leszek Podśędkowski i mgr inż. Marek Idzikowski są pracownikami Instytutu Obrabiarek i TBM Politechniki Łódzkiej. Mgr Istvan Vizvary jest pracownikiem firmy AMG.net.

idzie czas poszukiwania ścieżki jest znacznie krótszy niż w rozwiązaniu [4], gdzie analogiczny graf jest przeszukiwany bez uwzględnienia kosztu heurystycznego. Wynikiem działania generatora ścieżki jest ścieżka łącząca bieżące położenie robota z celem, przedstawiona w postaci sekwencji bezpośrednio następujących po sobie konfiguracji robota (węzłów).

Do poprawy pracy sterownika ruchu istotna jest właściwa organizacja współpracy generatora ścieżki, pilota i czujników. By skrócić czas powtórnego planowania ścieżki powstała idea wykorzystania przez generator ścieżki informacji z poprzedniego jej planowania. Wiąże się to jednak ze zmianą sposobu budowania grafu przez generator ścieżki.

Większość autorów tworzy taki graf wychodząc z wierzchołka będącego konfiguracją startu, a kończy budowę grafu, gdy któryś z wierzchołków znajdzie się w tej samej konfiguracji dyskretnej co cel [4, 5, 6]. Wadą tego rozwiązania jest konieczność budowy całego grafu od początku gdy robot, po przejechaniu kawałka trasy, wykryje nowe przeszkody. Drugą wadą jest to, że cel jest osiągnięty z dokładnością odpowiadającą podziałce dyskretnej przestrzeni konfiguracyjnej, co w wielu przypadkach może być niewystarczające.

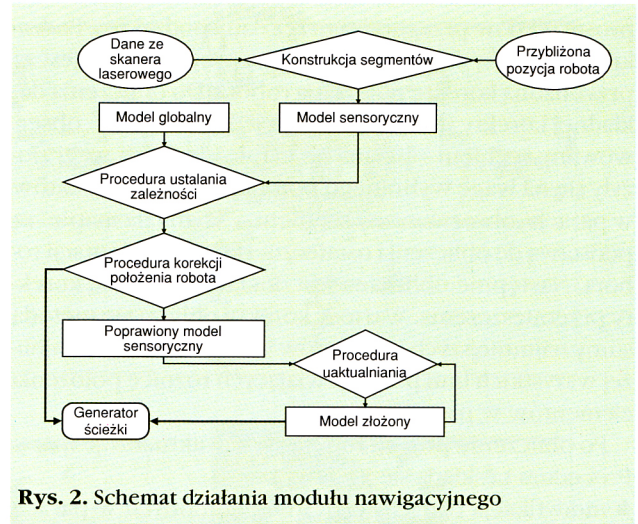
W proponowanym sterowniku ruchu (rys. 1) wyeliminowano te wady przez budowanie grafu startując z konfiguracji celu, oraz przez zapamiętywanie wszystkich połączeń w grafie, a nie tylko tych potrzebnych do utworzenia dendrytu. Dzięki temu cel jest osiągnięty ze znacznie większą dokładnością, a w przypadku natrafienia na dodatkowe przeszkody jest przebudowywana istniejąca część grafu i dobudowywana kolejna, co skutkuje znacznym skróceniem czasu powtórnego planowania ścieżki. Mogą oczywiście zaistnieć sytuacje kiedy, mimo takiej budowy sterownika ruchu, czas powtórnego planowania trajektorii jest jeszcze za długi. Jeśli będzie on dłuższy niż np. 0,4 s, pilot zatrzyma robota i będzie czekał na zaplanowanie nowej ścieżki.

Moduł nawigacyjny

Do zbierania informacji o stanie przestrzeni roboczej użyto dalmierza laserowego. Pomiar odległości może być wykonany tylko w osi optycznej przyrządu. Do odchylenia wiązki lasera zastosowano obrotowe lustro umożliwiająca skanowanie dookoła. Zasięg urządzenia wynosi około 15,5 m, dokładność pomiaru 2,54 mm. Skaner pracuje z maksymalną częstotliwością 50 kHz. Umożliwia to pracę systemu z dużymi częstotliwościami odświeżania, przy zachowaniu dużej rozdzielczości kątowej.

Sposób określania położenia robota opiera się na założeniu istnienia płaskich (prostych) fragmentów ścian. Schemat działania modułu nawigacyjnego jest przedstawiony na rys. 2. Informacje o otoczeniu przechowywane są w dwóch mapach. Mapa rastrowa jest używana przez moduł planowania trajektorii, a wektorowa służy do określania położenia i orientacji robota mobilnego.

Z pomiarów odometrycznych otrzymujemy przemieszczenie robota w stosunku do poprzedniego cyklu. Na tej



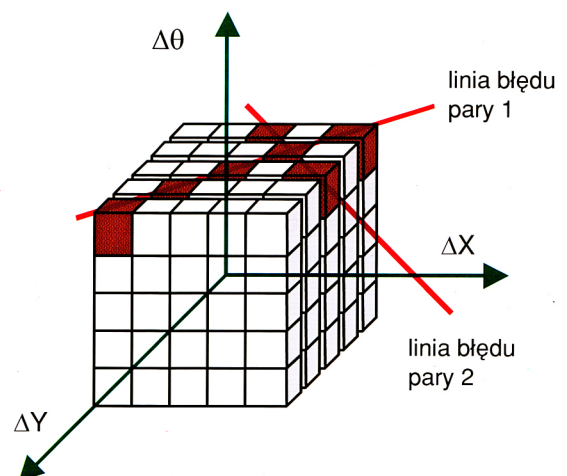
Rys. 2. Schemat działania modułu nawigacyjnego

podstawie obliczamy aktualne położenie i orientację robota. Zakładamy, że obliczenia te są obciążone pewnym błędem oraz, że błąd ten nie może przekroczyć pewnych wartości. Ponieważ używane są trzy współrzędne do opisu położenia robota, wszystkie możliwe kombinacje błędów X, Y i θ tworzą przestrzeń trójwymiarową zwaną przestrzenią błędów.

Do obliczenia korekty położenia wykorzystywane są informacje z mapy wektorowej otoczenia: „ściany na mapie” oraz dane z aktualnego skanu. Na podstawie danych ze skanera wychwytuje się serie punktów pomiarowych tworzących odcinki proste: obserwowane segmenty. Następnie na podstawie przybliżonego położenia robota ściany na mapie i obserwowane segmenty łączone są w pary. Każda taka para ma przypisaną wagę zależną od wagi ściany i długości segmentu obserwowanego oraz od liczby ścian, z którymi dany segment tworzy parę.

Na podstawie jednej pary można określić jedynie korektę w kierunku prostopadłym do kierunku obserwowanego segmentu. Graficzną tego reprezentacją jest linia prosta w przestrzeni błędów (rys. 3), zwana prostą błędów.

W celu obliczenia dokładnej wartości błędów przestrzeni błędów jest dzielona na podzakresy tworząc macierz trójwymiarową. Następnie, wagi odpowiadające każdemu z podzakresów są zwiększane o wagi tych par, których



Rys. 3. Przestrzeń błędów

proste błędów przechodzą przez dany podzakres. Podzakres, który uzyskał najwyższą wagę przyjmowany jest za przybliżoną korektę położenia robota. Do obliczenia dokładnej korekty uwzględniane są jedynie te pary „obserwowany segment – ściana na mapie”, których wagi złożyły się na wagę wybranego podzakresu. Różnice kątów w parach „obserwowany segment – ściana na mapie” są podstawą do obliczenia ostatecznej korekty orientacji robota. Następnie obliczana jest dokładna wartość korekty przemieszczenia. Wartość korekty oblicza się metodą sumy najmniejszych kwadratów jako punkt leżący najbliżej wszystkich linii przedstawiających różnicę położenia elementów w parach.

Po obliczeniu pozycji robota system aktualizuje mapy. Procedura ta składa się z kilku faz:

- modyfikacja danych ścian uwzględnionych w parach „obserwowany segment – ściana na mapie”,
- tworzenie nowych ścian na podstawie niesparowanych segmentów,
- określenie obszaru wolnego od przeszkód,
- usunięcie nieistniejących już ścian (znajdujących się w obszarze wolnym).

Następnie jest uaktualniana mapa przeszkód. Poziom pewności komórek, w których obserwowano przeszkody jest podnoszony. Jeżeli poziom ten przekroczy wartość progową, to system zaznacza komórkę jako zajętą. Jeśli cela na mapie przeszkód jest oznaczona jako zajęta i znajduje się w obszarze wolnym od przeszkód, to jej stopień pewności jest obniżany i gdy spadnie poniżej progu, cela jest oznaczana jako wolna.

Po uaktualnieniu pozycji robota oraz map moduł nawigacyjny jest gotowy do obróbki danych z następnego cyklu.

Rozpoznanie przestrzeni roboczej

Ze względu na przyjęty algorytm konieczne jest zdefiniowanie mapy wzorcowej (model globalny), aby umożliwić nawigację. Można to przeprowadzić na kilka sposobów. Pierwszy z nich polega na wprowadzeniu danych o istotnych przeszkodach, takich jak np. ściany pomieszczeń z dokumentacji. Ściany na mapie ścian zdefiniowanej w ten sposób powinny mieć maksymalny i niezmienny stopień pewności. Następnie system buduje na tej podstawie bitową mapę przeszkód. Metoda ta daje najlepszą dokładność tworzonej mapy, zakładając, że budynek został wykonany zgodnie z planem.

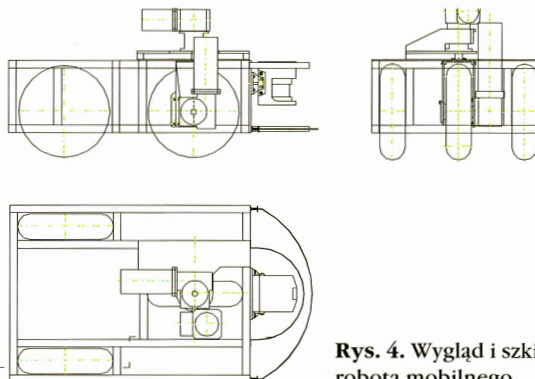
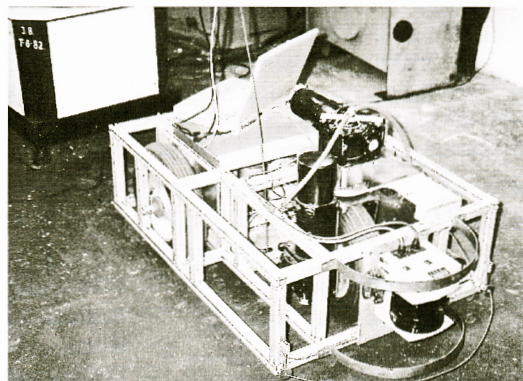
Drugi sposób polega na tym, że robot tworzy samodzielnie mapę ścian i przeszkód. W pozycji początkowej, tak dobranej aby widoczna była jak największa liczba płaskich fragmentów ścian, robot wykonuje serię pomiarów. Następnie przemieszcza się o niewielką odległość, zatrzymuje się i wykonuje następną serię pomiarów. Taką procedurę pozwala na precyzyjne utworzenie mapy ścian, a na jej podstawie mapy przeszkód.

Ostatnia z metod jest podobna do poprzedniej. Robot z pozycji początkowej zaczyna przemieszczać się i wykonuje pomiary bez zatrzymywania się. W wyniku powstaje mapa ścian o nieco gorszej dokładności niż w metodzie z wymuszonym zatrzymaniem.

Autorzy przetestowali drugą z metod. Podczas nawigacji przy tak stworzonej mapie ścian pomiary położenia robota były stabilne, a dokładność określenia położenia nie przekraczała 10 mm.

Badania doświadczalne

W celu przeprowadzenia testów został zbudowany w Instytucie Obrabiarek i TBM Politechniki Łódzkiej trójkołowy robot mobilny o wymiarach 0,8 × 1,3 m. Na rys. 4 jest przedstawiony projekt tego robota oraz jego fotografia. Robot ma dwa tylne koła obracające się swobodnie wokół osi prostopadłej do osi robota. Przednie koło jest zarówno napędzane jak i sterowane (obracane wokół pionowej osi).



Rys. 4. Wygląd i szkic robota mobilnego

Robot jest sterowany z zamontowanego na nim przenośnego komputera Toshiba Satellite Pro 440 z procesorem Pentium 133 MHz, a zasilany z sieci 220 V lub z zestawu akumulatorów.

Przed wykonaniem właściwych testów robot przejechał kilka razy po terenie badań i utworzył wstępną jego mapę. Podczas testów przyjęto 44 dyskretne położenia katowe oraz dziesięcioelementową dyskretną przestrzeń sterowania (do przodu, do tyłu) × (w lewo z minimalnym promieniem skrętu, w lewo z dwa razy większym od minimalnego promieniem skrętu, prosto, w prawo z dwa razy większym od minimalnego promieniem skrętu, w prawo z minimalnym promieniem skrętu).

Przeprowadzono kilkaset testów, trwających łącznie kilka godzin. Część zmian kształtu przeszkód wynikała z drobnych niedokładności systemów określania położenia i identyfikacji przeszkód powodujących czasami „pływanie” granic przeszkód, inne celowo dostawiano lub usuwano np: kanister po oleju lub człowiek.



Rys. 5. Przejazd z wykryciem zamknięcia korytarza



Rys. 6. Przejazd z wykryciem otwartego przejścia

Podczas badań można było wyróżnić trzy podstawowe typy zachowań sterownika ruchu:

- sprawdzenie, że zgłoszone zmiany przestrzeni roboczej nie wymagają zmiany trasy,
- lokalna korekta trasy wynikająca z małych zmian przestrzeni roboczej,
- zmiany wynikające z zablokowania przejścia, którym robot miał przejechać,
- zmiany wynikające z pojawienia się dodatkowego przejścia umożliwiającego dotarcie do celu bardziej optymalną trasą.

Na uwagę zasługuje fakt, że te typy zachowań wynikają z rodzaju wykrytych zmian przestrzeni roboczej i nie są narzucone w programie. Na rys. 5 i 6 są przedstawione slajdy z dwóch przejazdów robota po terenie laboratorium. Kolorem czarnym oznaczone są przeszkody oraz planowana trasa (ścieżka) robota. Kolorem szarym zaznaczony jest obszar, na którym zbudowano graf podczas pierwszego planowania ścieżki, a kolorem żółtym – podczas kolejnych. Kolorem niebieskim jest zaznaczona trasa przejechana przez robota. Na rys. 5 pierwsze planowanie ścieżki trwało 0,67 s. W trakcie jazdy sterownik ruchu 294 razy reagował na zmiany układu przeszkód. W większości przypadków czas reakcji wynosił 0,15 – 0,2 s, a ścieżka nie była zmieniana. Lokalne korekty trasy trwały od 0,2 s do 0,4 s. Po wykryciu zamkniętego przejścia robot planował nową trasę przez 1,67 s.

Na rys. 6 pierwsze planowanie ścieżki trwało 1,53 s, a liczba reakcji na zmiany układu przeszkód wynosiła 153. Po wykryciu otwartego przejścia robot planował nową trasę przez 1,2 s.

Badania potwierdziły poprawność działania inteligentnego sterownika ruchu. Znalaziona przez niego ścieżka prowadziła robota po trasie bezkolizyjnej tak, że robot

dojeżdżał do celu z dokładnością limitowaną jedynie dokładnością czujników. Na podstawie oceny wzrokowej można było stwierdzić, że ścieżka była optymalna, bądź bliska optymalnej na tyle, że operator sterując robotem ręcznie nie pokierowałby go lepiej.

Podczas badań przeprowadzono również testy inteligentnego sterownika ruchu w sytuacjach z ruchomymi przeszkodami. W sytuacjach, gdy ruchome przeszkody znajdowały się w polu widzenia robota i poruszały się na tyle wolno, że można je było uznać za chwilowo nieruchome, sterownik działał poprawnie. Jeżeli jednak przeszkoda poruszała się w taki sposób, że robot w celu uniknięcia kolizji musiałby przewidywać jej ruchy, to następowało zderzenie robota z przeszkodą.

Bibliografia

1. Podśędkowski L.: Very well informed A* searching algorithm and its application for nonholonomic mobile robot motion planning. International Journal of Science and Technology. USA 1999.
2. Podśędkowski L.: Sterownik ruchu dla częściowo znanej przestrzeni roboczej. VI Krajowa Konferencja Robotyki Świeradów-Zdrój 1998, s. 187 – 194.
3. Podśędkowski L.: Dynamiczne planowanie trajektorii robotów mobilnych w zmiennej przestrzeni roboczej. Zesz. Nauk. PŁ, Nr 809, Rozpr. Nauk., z. 254, 1999 r.
4. Barraquand J., Latombe J.-C.: On Non-Holonomic Mobile Robots and Optimal manoeuvring. Revue d'intelligence artificielle. Vol. 3, no. 2, 1989, s. 77 – 103.
5. Fraichard T., Laugier C.: On line reactive planning for a non-holonomic mobile in a dynamic world. IEEE Intern. Conf. on Robotics and Automation, Sacramento, 1991, s. 432-437.
6. Latombe J. C.: Robot Motion Planning, Kluwer Academic Publishers 1991.

Streszczenia artykułów naukowych

Sterownik ruchu robota mobilnego dla częściowo znanej przestrzeni roboczej, Leszek Podśędkowski, Leszek Kwapisz, Marek Idzikowski, Istvan Vizvary – s. 18

Przedstawiono system sterowania robotem mobilnym. System ten składa się z czterech modułów: Interfejs użytkownika, Generator ścieżki, Pilot i Skaner. Omówiono strukturę sterownika ruchu, algorytmy postępowania i wyniki badań doświadczalnych na rzeczywistym robocie dla modułów Generatorsa ścieżki i Skanera

Mobile robot controller for partially defined working space

Leszek Podśędkowski, Leszek Kwapisz, Marek Idzikowski, Istvan Vizvary – p. 18

A mobile robot control system is described. The whole system consists of four modules: User Interface, Path Generator, Pilot and Scanner. There are presented structure of the motion controller, procedure algorithms and results of experiments conducted with prototype mobile robot for Path Generator and Scanner modules.
