

dr hab. inż. Leszek Podseǳkowski  
prof. dr inż. Leszek Kwapisz  
mgr inż. Marek Idzikowski  
mgr. Istvan Vizvary  
Instytut Obrabiarek i TBM  
Politechniki Łódzkiej.

## Sterownik ruchu robota mobilnego dla częściowo znanej przestrzeni roboczej – struktura i wyniki badań doświadczalnych generatora ścieżki.

W artykule zawarto opis systemu sterowania robotem mobilnym. System ten składa się z czterech modułów: Interfejs użytkownika, Generator ścieżki, Pilot i Skaner. Skaner jest odpowiedzialny za obróbkę danych ze skanera laserowego i za dokładne określanie położenia robota i otaczających go przeszkód. Generator ścieżki planuje optymalną (według chwilowego stanu wiedzy) trasę robota z jego bieżącego położenia do celu. Wykorzystuje on zmodyfikowaną metodę przeszukiwania grafu algorytmem  $A^*$  z kosztem heurystycznym wyznaczonym metodą propagacji kosztu. Dodatkowo metoda została wyposażona w procedurę szybkiej zmiany trasy po wykryciu zmian w przestrzeni roboczej. Pilot steruje silnikami tak, by robot jechał po wyznaczonej trasie a całość jest nadzorowana przez Interfejs użytkownika. Artykuł przedstawia strukturę sterownika ruchu, algorytmy postępowania jak i wyniki badań doświadczalnych na rzeczywistym robocie.

### 1. Wstęp

W literaturze krajowej i światowej jest opisywanych wiele metod wyznaczania trajektorii ruchu robota. Można je podzielić na potencjalne i globalne. Metody potencjalne polegają na sterowaniu robotem w kierunku celu z omijaniem przeszkód znajdujących się w bezpośrednim sąsiedztwie robota. Można w nich w prosty sposób uwzględnić ograniczenia nieholonomiczne robota. Nie zapewniają one jednak znalezienia prawidłowej (optymalnej) trajektorii w przestrzeni z wieloma przeszkodami i ślepych korytarzami (np. w labiryntach). Do grupy metod potencjalnych można zaliczyć również sterowanie robotów mobilnych za pomocą sieci neuronowych. Problem labiryntów i pomieszczeń zamkniętych jest rozwiązywany przez metody z potencjałem numerycznym, nie uwzględniają one jednak ograniczeń nieholonomicznych robotów.

Do podstawowych metod globalnych należą metody typu *roadmap* - mapy dróg i *cell decomposition* - podziału przestrzeni roboczej. Podstawowe wersje tych metod nie uwzględniają ograniczeń nieholonomicznych polegających na tym, że robot typu samochód może poruszać się tylko do przodu lub tyłu a nie może poruszać się bokiem. Spośród wielu innych metod interesującymi wydają się przedstawione w [1] i [3] metody "punkt z minimalnym promieniem skrętu" oraz "rodzic - potomkowie". Są to metody rozwiązujące problem znalezienia trajektorii w sposób globalny z uwzględnieniem ograniczeń

nieholonomicznych, jednakże czas ich obliczeń jest bardzo długi.

W Instytucie Obrabiarek i TBM PŁ od 1994 roku prowadzone są badania mające na celu stworzenie szybkiej metody planowania optymalnej trajektorii robotów mobilnych uwzględniającej ich ograniczenia nieholonomiczne. Pierwszym etapem było opracowanie metody generowania ścieżki przy wykorzystaniu grafu skoków jednostkowych, algorytmu A\* i mapy kosztu heurystycznego [5]. Drugim etapem było opracowanie algorytmu korygowania ścieżki w chwili wykrycia zmiany układu przeszkód [4] i [6]. Opracowane metody zostały zaimplementowane i został utworzony sterownik ruchu.

Na obecnym etapie badań (od 1998 roku) prowadzona jest weryfikacja doświadczalna metody na rzeczywistym robocie.

## 2. Sterownik ruchu

*Sterownik ruchu* jest systemem komputerowym, który ma za zadanie planować i kontrolować ruch robota. Aby mógł robić to poprawnie musi współpracować z innymi systemami. Przede wszystkim *sterownik ruchu* powinien znać cel do którego robot ma dotrzeć. Zasada działania systemu wykonawczego decyduje o dokładności, z jaką ten cel ma być osiągnięty. Z systemu określania położenia robota *sterownik ruchu* otrzymuje informacje o aktualnym położeniu robota, a z systemu reprezentacji wiedzy informacje o aktualnym stanie wiedzy na temat otoczenia. Na tej podstawie wyznacza optymalną i bezkolizyjną ścieżkę do celu i wysyła do systemu napędowego sygnały sterujące tak, by robot jechał po wyznaczonej ścieżce. W przypadku wykrycia nowych przeszkód *sterownik ruchu* modyfikuje ścieżkę tak, by znów była optymalna i bezkolizyjna.

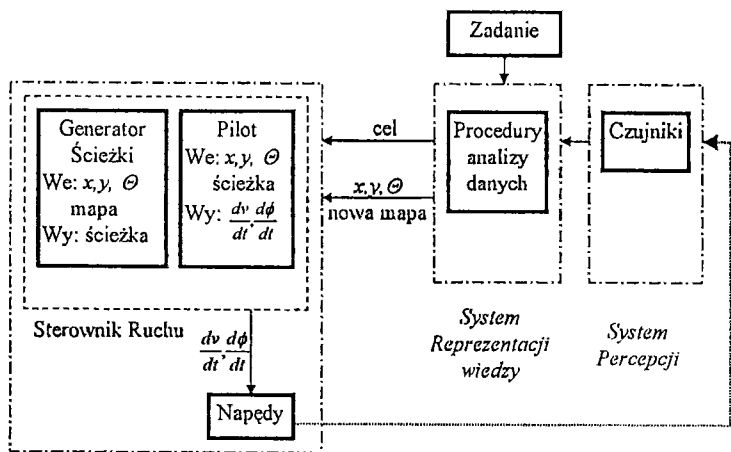
Opracowany w IOiTBM *sterownik ruchu* robota mobilnego, spełnia następujące założenia:

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

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

*Generator ścieżki* (wyższa warstwa *sterownika ruchu*) służy do planowania trasy robota. Jego sygnałami wejściowymi są: bieżące położenie robota, położenie celu oraz układ (mapa) przeszkód. Sygnałem wyjściowym (zmienną decyzyjną) *generatora ścieżki* jest ścieżka łącząca bieżące położenie z celem. Ścieżka ta jest przedstawiona w postaci sekwencji kolejnych konfiguracji robota, przez które powinien przejechać by znaleźć się w celu. Ścieżka jest wyznaczana przy uwzględnieniu więzów nieholonomicznych robota jest więc kinematycznie realizowalna. Podczas wyznaczania ścieżki *generator ścieżki* minimalizuje funkcję kosztu (wskaźnik jakości) czyli wyznacza ścieżkę optymalną. Za podstawę kosztu jest przyjęta długość ścieżki. Funkcja kosztu może jednak uwzględniać dodatkowe preferencje operatora robota zwiększające wartość tego kosztu np. w przypadku jazdy do tyłu, zmiany kierunku ruchu, jazdy blisko przeszkody lub zmiany promienia skrętu: W efekcie ścieżka optymalna może być ścieżką najkrótszą, o najkrótszym czasie przejazdu lub najmniejszej ilości energii potrzebnej do jej pokonania.

Niższą warstwą *sterownika ruchu* jest *pilot*. Jego sygnałami wejściowymi są ścieżka z *generatora ścieżki* i bieżące położenie robota. Sygnałami wyjściowymi są przyspieszenie robota  $dv/dt$  i prędkość  $dK/dt$  zmiany krzywizny toru ruchu punktu charakterystycznego



System Planowania i wykonywania działań

Rys. 1 Przepływ informacji w robocie mobilnym

robota. *Pilot*, biorąc pod uwagę ograniczenia dynamiczne napędów, tak ustala wartości sygnałów wyjściowych, by robot jak najdokładniej przemieszczał się wzdłuż ścieżki. Dodatkową funkcją *pilota* jest zatrzymanie robota w sytuacji, gdy *generator ścieżki* koryguje trasę na tyle długo, że dalsza jazda grozi kolizją.

Najniższą warstwą systemu sterowania robotem są elektroniczne sterowniki poszczególnych napędów, których sygnałami wejściowymi są odpowiednio przeliczone sygnały z *pilota*.

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 planowana jest cała droga samochodu. *Pilot* jest odpowiednikiem odruchów. Pierwszy z tych odruchów to taka praca pedałami 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.

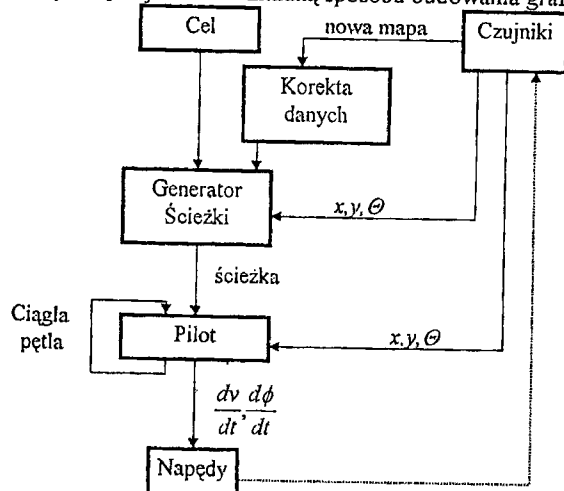
Jak wynika z tego podziału *generator ścieżki* jest odpowiedzialny nie tylko za pierwsze zaplanowanie ścieżki, ale także za jej powtórne planowanie przy zmianach przestrzeni roboczej. Zwłaszcza konieczność powtórnego planowania stawia przed *generatorem ścieżki* wymagania niskiej czasochłonności.

By skrócić czas obliczeń wielu autorów [2] znajduje najpierw ścieżkę nie uwzględniając więzów nieholonomicznych, a następnie wygładza ją biorąc je pod uwagę. W większości przypadków wygładzanie ścieżki odbywa się na poziomie nawigatora lub pilota. W proponowanym tu *generatorze ścieżki* problem poszukiwania ścieżki również jest rozwiązywany najpierw bez uwzględniania ograniczeń nieholonomicznych, ale wynikiem poszukiwania nie jest jedna konkretna ścieżka, lecz mapa będąca jakby zbiorem drogowoskazów w jakich kierunkach należy szukać drogi do celu. Mapa ta, nazywana dalej mapą kosztu heurystycznego jest tworzona metodą propagacji szesnastokierunkowej. Drugim etapem jest tworzenie grafu dróg, którego wierzchołkami są konfiguracje robota, a krawędziami krótkie fragmenty drogi robota po łuku lub wzdłuż prostej. Graf taki tworzy się startując z jednego wierzchołka i znajdując kolejnych jego następników. Taki graf jest

nazywany dendrytem, a początkowy wierzchołek korzeniem tego dendrytu. Podczas budowy grafu należy uwzględnić więzy nieholonomiczne robota. Jak wynika z analizy literatury istnieją dwa podstawowe sposoby tworzenia krawędzi i węzłów grafu. Pierwszy z nich zakłada, że krawędzie są krótkimi skokami o stałej długości i stałych wartościach zmiennych sterujących. Drugi tworzy długie krawędzie składające się z wzajemnie stycznych odcinków i łuków łączące charakterystyczne konfiguracje (znajdujące się w wierzchołkach przeszkód). Z uwagi na to, że w drugim sposobie konieczne jest zapisanie mapy przeszkód w postaci wieloboków zdecydowano się na pierwsze rozwiązanie (z przestrzenią roboczą zapisaną w postaci mapy rastrowej).

Dzięki wprowadzeniu mapy kosztu heurystycznego tworzony dendryt ma znacznie mniej węzłów a co za tym idzie czas poszukiwania ścieżki jest znacznie krótszy niż w rozwiązaniu [1] 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).

Dla poprawnej 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 planowania ścieżki. Wiąże się to jednak ze zmianą sposobu budowania grafu przez *generator ścieżki*.



Rys. 2. Inteligentny sterownik ruchu

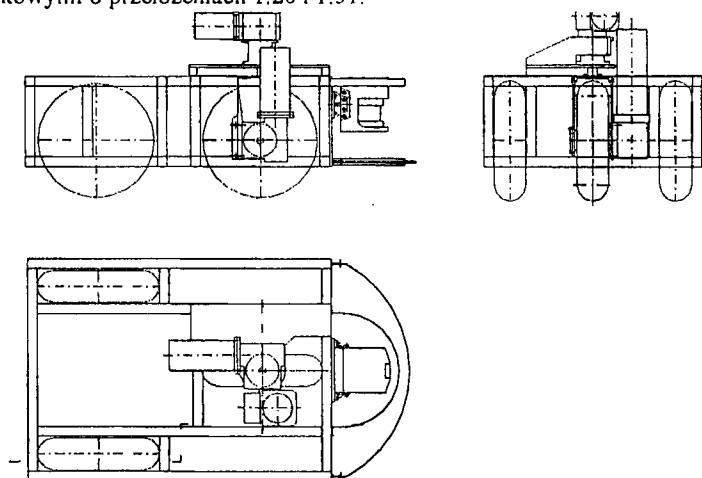
Większość autorów tworzy taki graf wychodząc z wierzchołka będącego konfiguracją startu, a kończy budowę grafu gdy któryś z wierzchołków znajdzie się w tej samej konfiguracji dyskretnej co cel. Wadą tego rozwiązania jest konieczność budowy całego grafu od początku gdy robot, po przejechaniu kawałka trasy, 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. 4) 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. Wówczas, jeśli będzie on dłuższy niż np. 0,4 sekundy *pilot* zatrzyma robota i będzie czekał na zaplanowanie nowej ścieżki.

### 3. Badania doświadczalne

W celu przeprowadzenia testów został zbudowany w Instytucie Obrabiarek i TBM Politechniki Łódzkiej trójkołowy robot mobilny o wymiarach 0,8 x 1,3 m. Na rysunku 3 przedstawiony jest projekt tego robota a na rysunku 4 jego zdjęcie. 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). Do napędu i obrotu wykorzystano komutatorowe silniki prądu stałego PZTK 10-08 i PZTK 88-35 odpowiednio z przekładniami ślimakowymi o przełożeniach 1:20 i 1:51.

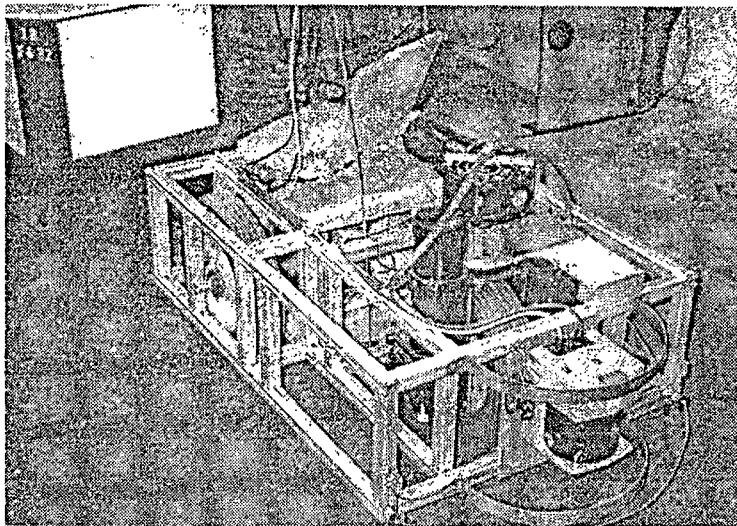


Rys.3. Szkic robota mobilnego

Maksymalny kąt obrotu przedniego koła wynosi blisko  $\pi$ , ale w celu uwypuklenia ograniczeń nieholonomicznych został on programowo ograniczony tak, by minimalny promień skrętu robota (mierzony w punkcie dyferencjału) wynosił 1m. Silniki są sterowane za pośrednictwem karty UCNC 3 i sterowników 25A8 i 50A8. W celu zamknięcia pętli sprzężeń zwrotnych wykorzystywane są zainstalowane na silnikach prądnice tachometryczne i przetworniki obrotowo-impulsowe. Przed przednim kołem umieszczony jest skaner laserowy zabezpieczony sztywnym zderzakiem. Dodatkowo zamocowany jest z przodu zderzak podatny z wyłącznikami krańcowymi zatrzymujący awaryjnie robota w chwili zetknięcia z przeszkodą. Robot sterowany jest z zamontowanego na nim przenośnego komputera Toshiba Satellite Pro 440 z procesorem Pentium 133MHz, a zasilany jest z sieci 220V lub z zestawu akumulatorów.

Do identyfikacji położenia przeszkód i określania położenia robota względem przeszkód został wykorzystany skaner laserowy SICK PLS. Mierzy on odległość od przeszkód w zakresie 0-180° z rozdzielczością 0,5°, dokładnością  $\pm 15$  mm i częstotliwością 25 Hz. Pomiar jest dokonywany w płaszczyźnie równoległej do podłoża i odległej od niego o 0,25 m. Badania wykonywano w Laboratorium Robotyki Instytutu Obrabiarek i TBM Politechniki Łódzkiej. Położenia istniejących tam przeszkód nie zmieniano, dołożono jedynie kilka osłon tekturowych na przeszkody znajdujące się poniżej lub powyżej płaszczyzny w której dokonywany jest pomiar skanerem. Teren badań przedstawiono w postaci mapy rastrowej o

wymiarze 128×128 i o długości boku jednego rastra 0,1 m. Przed przystąpieniem do właściwych testów robot przejechał kilka razy po terenie badań i utworzył wstępną jego mapę.



Rys.4. Wygląd robota mobilnego

Podczas testów przyjęto 44 dyskretne położenia kątowe 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). Wielkość skoku jednostkowego była równa 0,285 m. Wprowadzono współczynniki kosztu dotyczące kształtu znalezionej drogi decydujące o preferencjach użytkownika odnośnie unikania jazdy tyłem i nawrotów.

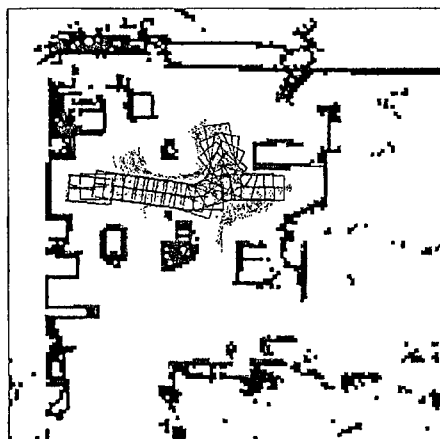
Przeprowadzono kilkaset testów o łącznym czasie trwania ponad kilka godzin. Pierwsza część testów związana była z usuwaniem błędów programowych i dobraniem właściwych wartości parametrów planowania i śledzenia ścieżki. Później proces planowania przebiegał bezawaryjnie i działanie robota było prawidłowe. Jedynymi wyjątkami były sytuacje gdy robot napotykał na przeszkody których nie był w stanie wcześniej wykryć (na przykład podczas jazdy tyłem). Zmiany kształtu przeszkód w trakcie badań były dwojakiemu rodzaju. Część wynikała z drobnych niedokładności systemów określania położenia i identyfikacji przeszkód powodujących czasami „pływanie” granic przeszkód. Inne były to celowo dostawiane lub usuwane przeszkody: kanister po oleju lub człowiek.

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

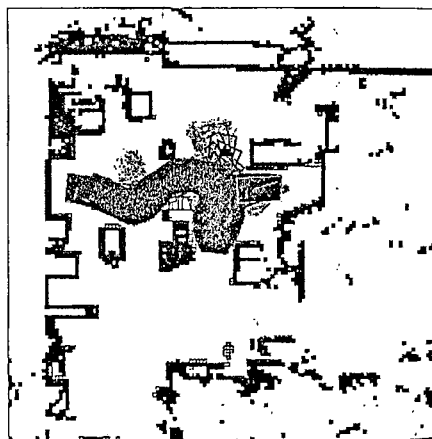
- lokalna korekta trasy wynikająca z drobnych zmian przestrzeni roboczej,
- zmiany wynikające z zablokowania przejścia którym robot miał przejechać,
- zmiany wynikające z pojawienia się dodatkowego przejścia umożliwiającego dotarcie do celu bardziej optymalną trasą.

Te trzy podstawowe sytuacje zostały zaprezentowane na rysunkach 5-10. Na rysunkach 5, 7 i 9 jest przedstawiona pierwotnie zaplanowana ścieżka, a na pozostałych ostateczny kształt trajektorii.

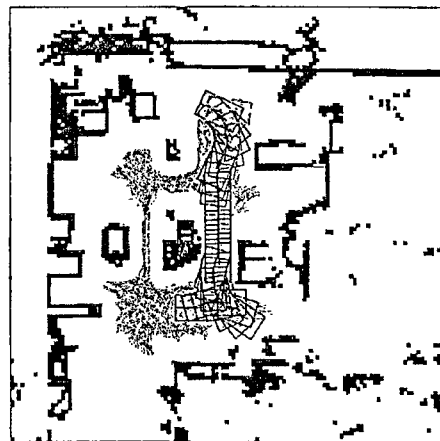
W pierwszym przypadku (rys. 5 i 6) czas pierwszego planowania ścieżki wynosił 0,66 s. a czasy 23 korekt ścieżki wynosiły od 0,07 do 0,4 s. Ponieważ były one krótsze od założonego progu 0,4 s. robot nie zatrzymywał się podczas jazdy. Położenie docelowe zostało osiągnięte z dokładnością 0,02 m. i 0,02 rad. W drugim przypadku (rys. 7 i 8) czas pierwszego planowania ścieżki wynosił 1 s. a czasy 32 korekt ścieżki wynosiły od 0,09 do 1 s. Po wykryciu zablokowania zaplanowanej ścieżki robot zatrzymał się na 0,6 sekundy i podjął jazdę już wzdłuż nowej trasy. Położenie docelowe zostało osiągnięte z dokładnością 0,06 m. i 0,14 rad. W trzecim przypadku (rys. 9 i 10) czas pierwszego planowania ścieżki wynosił 1,5 s. a czasy 39 korekt ścieżki wynosiły od 0,09 do 0,4. Najdłuższy z tych czasów odpowiadał korekcie ścieżki po stwierdzeniu braku przeszkody, nie spowodował jednak zatrzymania robota. Położenie docelowe zostało osiągnięte z dokładnością 0,05 m. i 0,09 rad.



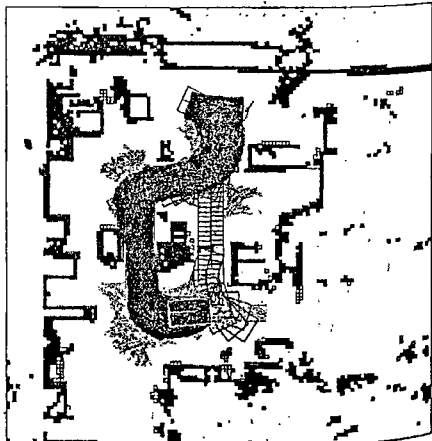
Rys. 5. Trasa zaplanowana przed lokalną korektą trasy.



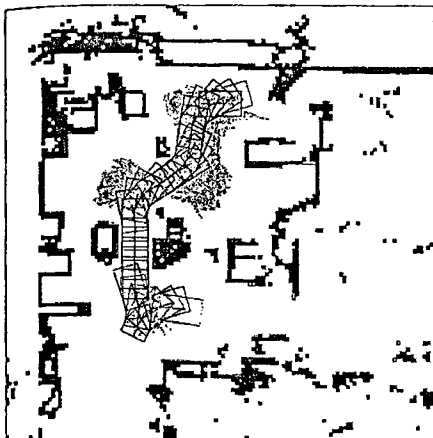
Rys. 6. Trasa po lokalnych korektach.



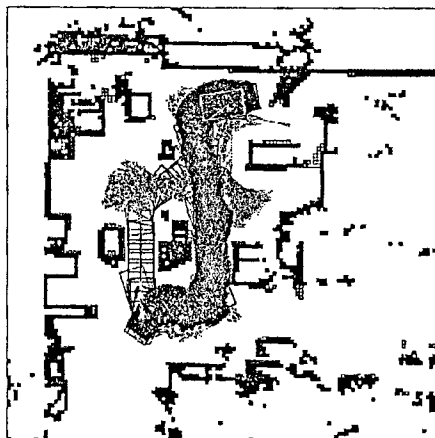
Rys. 7. Trasa zaplanowana przed wykryciem zastawienia korytarza.



Rys. 8. Trasa zrealizowana po wykryciu zamknięcia korytarza.



Rys.9. Trasa zaplanowana przed wykryciem otwarcia korytarza.



Rys.10. Trasa zrealizowana, po wykryciu otwarcia korytarza

Na podkreślenie zasługuje fakt, że choć przy każdym odczycie danych ze skanera występowały jakieś zmiany informacji o przestrzeni roboczej to w wymienionych przypadkach robot tylko raz musiał zatrzymać się by mieć czas na zaplanowanie nowej ścieżki. Podobnie rzadko robot zatrzymywał się w pozostałych nie opisanych przypadkach.

Badania potwierdziły poprawność działania *inteligentnego sterownika ruchu*. Znaleziona 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 pokierował by 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.

- [1] BARRAQUAND J., LATOMBE J-C , *On Non-Holonomic Mobile Robots and Optimal manoeuvring*, Revue d'intelligence artificielle. Vol.3, no.2, 1989, str. 77-103.
- [2] FRAICHARD T., LAUGIER C., *On line reactive planning for a non-holonomic mobile in a dynamic world*, IEEE International Conference on Robotics and Automation, Sacramento, 1991, str. 432-437.
- [3] LATOMBE J.C., *Robot Motion Planning*, Kluwer Academic Publishers 1991.
- [4] PODSĘDKOWSKI L. *Sterownik ruchu dla częściowo znanej przestrzeni roboczej VI* Krajowa Konferencja Robotyki Świeradów-Zdrój 1998, str. 187-194.
- [5] PODSĘDKOWSKI L. „*Very well informed A\* searching algorithm and its application for nonholonomic mobile robot motion planning*” „International Journal of Science and Technology” USA 1999.
- [6] PODSĘDKOWSKI L. „*Dynamiczne planowanie trajektorii robotów mobilnych w zmiennej przestrzeni roboczej*”, Zeszyty Naukowe PŁ, Nr 809, Rozprawy Naukowe, z. 254, 1999 r.