

**Tomasz BURATOWSKI, Tadeusz UHL**

AKADEMIA GÓRNICZO-HUTNICZA, KATEDRA ROBOTYKI I DYNAMIKI MASZYN

## Zastosowanie algorytmów planowania trajektorii dla mobilnych robotów kołowych

Dr inż. Tomasz BURATOWSKI

Jest pracownikiem w Katedrze Robotyki i Dynamiki Maszyn AGH. Jego zainteresowania skupiają się na mechatronice oraz wykorzystaniu sztucznej inteligencji. Jest autorem prac nad zastosowaniem powyższych technik do zadań związanych z modelowaniem, identyfikacją i sterowaniem mobilnymi robotami.

e-mail: [tburatow@agh.edu.pl](mailto:tburatow@agh.edu.pl)



Prof. dr hab. inż. Tadeusz UHL

Jest kierownikiem Katedry Robotyki i Dynamiki Maszyn, Akademii Górniczo-Hutniczej w Krakowie. W swoich pracach zajmuje się zagadnieniami dynamiki konstrukcji, a zwłaszcza ich analizy modalnej. Jego zainteresowania obejmują także układy aktywnej redukcji drgań, układy sterowania i szeroko pojętą mechatronikę. Jest autorem 15 książek i kilkuset artykułów dotyczących wspomnianych zagadnień.

e-mail: [tuhl@uci.agh.edu.pl](mailto:tuhl@uci.agh.edu.pl)



### Streszczenie

W artykule przedstawiono kilka algorytmów planowania trajektorii dla mobilnych robotów kołowych, które między innymi zostały wybrane dla grupy robotów. W trakcie konstruowania prototypu urządzeń wykorzystano układy sensoryczne oparte o czujniki ultradźwiękowe oraz czujniki podczerwieni.

### Abstract

This paper presents several path planning algorithms for mobile robots. In the process of mechatronic design, two sensors: ultrasonic and infrared have been applied.

## 1. Wstęp

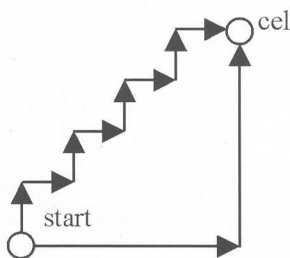
Analizując metody planowania trajektorii dla mobilnych robotów kołowych można stwierdzić iż istnieje wiele interesujących algorytmów umożliwiających zaprojektowanie toru ruchu robota [1,2,3,4]. Podstawową sprawą w zagadnieniu planowania trajektorii mobilnych robotów i nie tylko robotów jest określenie podstawowych definicji związanych z poruszającym się obiektem w przestrzeni. Podstawowe prawo związane z planowaniem trajektorii można zdefiniować następująco [1]:

Jeżeli  $W$  oznacza przestrzeń robota i  $C_i$  oznacza  $i$ -tą przeszkodę tak więc wolna przestrzeń robota, FS, jest zdefiniowana następująco:

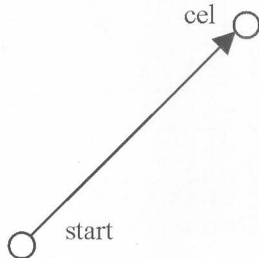
$$FS = W - (\cup C_i)$$

i trajektoria  $c \in C^0$  jest zdefiniowana jako:  $c : [0,1] \rightarrow FS$  gdzie  $c(0) \rightarrow p_{start}$  i  $c(1) \rightarrow p_{cel}$

Mając tak zdefiniowane zagadnienie można przejść do zdefiniowania sposobu określania odległości robota, między innymi, od przeszkód, punktu startowego ( $p_{start}$ ), celu ( $p_{cel}$ ). Przykładowo można określać odległość wykorzystując metrykę L1 ( $x,y$ ):  $|x| + |y| = const$  lub wykorzystując metrykę L2 ( $x,y$ ):  $x^2 + y^2 = const$ . Na rys.1 i rys.2 przedstawiono interpretacje graficzna obu metryk. W przypadku metryki L1 przedstawiono dwa z wielu możliwych rozwiązań.



Rys. 1. Metryka L1

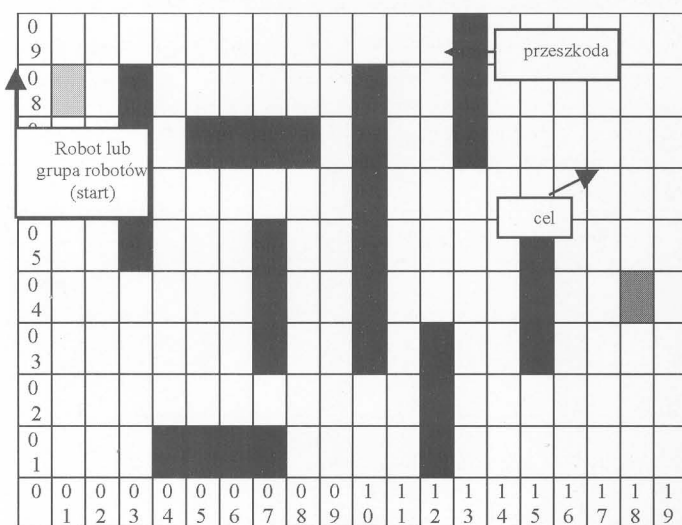


Rys. 2. Metryka L2

## 2. Zastosowanie algorytmów planowania trajektorii

Na podstawie tak zdefiniowanych pojęć w prosty sposób można przedstawić przykładowy obszar (rys.3) w którym ma się poruszać robot oraz dotrzeć do miejsca przeznaczenia.

Jednymi z najprostszych algorytmów jakie można zastosować w procesie planowania trajektorii dla robota lub grupy mobilnych robotów kołowych są algorytmy "Lumelsky Bug" [1], w których robot po wykryciu prze-



Rys. 3. Przykładowy obszar do eksploracji przez robota lub grupę robotów mobilnych.

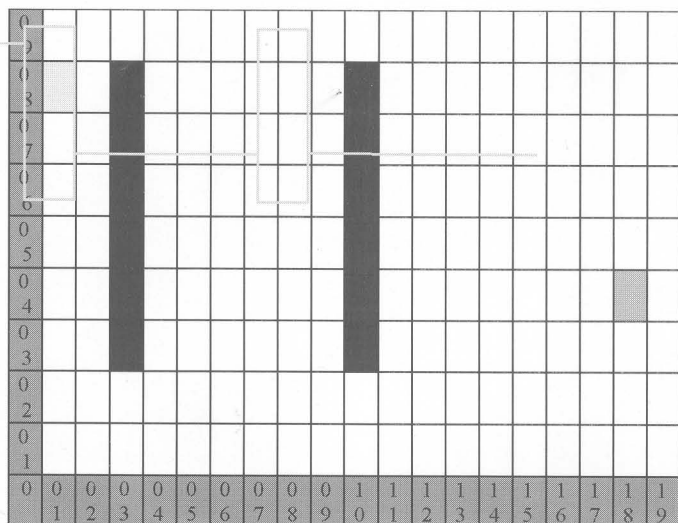
szkody podejmuje odpowiednią strategię działania. Algorytmy te zostały zainspirowane zachowaniem się insektów (np. mrówek). Tego typu algorytmy oparte są na założeniu iż znane jest położenie startowe robota (lub grupy) oraz znane jest położenie miejsca docelowego (cel), natomiast nieznane jest położenie przeszkód w danym obszarze. Podsumowując w tak zdefiniowanym zadaniu robota "wie" gdzie jest cel do którego ma podążać, ale nie wie jaką trajektorią. Zaletą algorytmów "Lumelsky Bug" jest ich prostota a także to iż mobilne roboty wystarczy wyposażyć w proste sensory taktyczne oraz enkodery. Do grupy algorytmów "Lumelsky Bug" zalicza się następujące sposoby planowania trajektorii:

"Bug 1" [1] Robot lub grupa robotów rozpoczyna ruch z punktu startowego. Każdy robot arbitralnie wybiera kierunek ruchu i eksploruje obszar. Po wykryciu przeszkody robot objeżdża ją dokoła zapamiętując punkt najbliższy do celu, następnie jeszcze raz objeżdża ją aby opuścić ją w zapamiętany punkcie. Należy podkreślić iż w algorytmie na początku należy zdefiniować w którą stronę robot ma skrócić po wykryciu przeszkody (skręt w prawo lub w lewo). Na rys.4 przedstawiono planowanie trajektorii z wykorzystaniem algorytmu "Bug 1" dla przykładowego obszaru z dwoma przeszkodami.

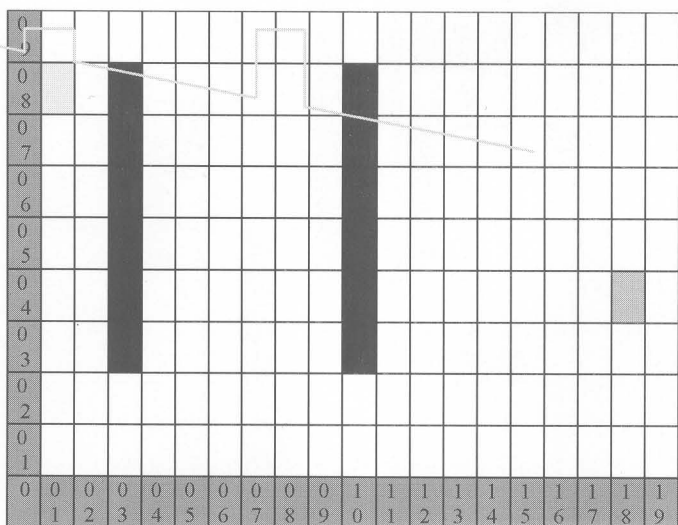
"Bug 2" [1] Robot lub grupa robotów rozpoczyna ruch z punktu startowego. Każdy robot arbitralnie wybiera kierunek ruchu i eksploruje obszar. Po wykryciu przeszkody robot objeżdża ją ale tylko do momentu gdy osiągnie punkt który umożliwia mu dalsze kontynuowanie ruchu na kierunku do celu.

Podobnie jak w poprzednim algorytmie na początku należy zdefiniować w którą stronę robot ma skrócić po wykryciu przeszkody (skręt w prawo lub w lewo). W przedstawionych przykładach z wykorzystaniem algorytmów "Lumelsky Bug" przyjęto iż robot po wykryciu przeszkody skręca zawsze w lewo. Na rys.5 przedstawiono planowanie trajektorii z wykorzystaniem algorytmu "Bug 2" dla przykładowego obszaru z dwoma przeszkodami.

Następnym ciekawym algorytmem związanym z planowaniem trajektorii dla robotów jest algorytm "Wavefront" [2]. Najważniejszym założeniem tego algorytmu jest poszukiwanie najkrótszej trajektorii pomiędzy punktem startowym a celem z uwzględnieniem, oczywiście, omięcia przeszkód występujących na tej trajektorii. W celu przedstawienia rozwiązania w sposób przejrzysty należy dokonać klasteryzacji obszaru w którym robot lub



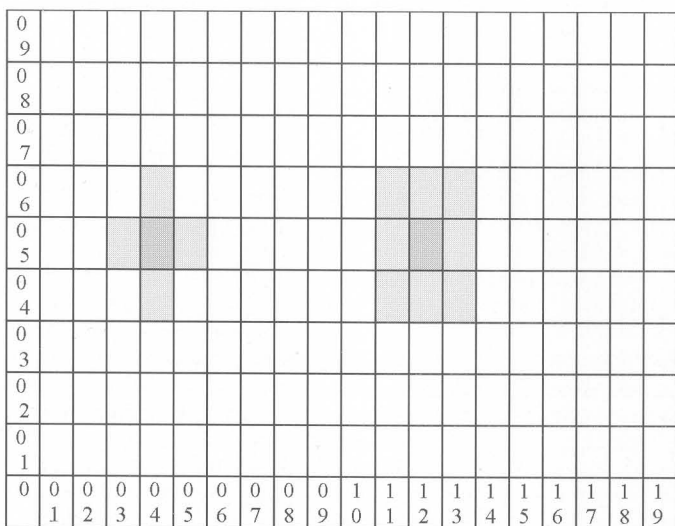
Rys. 4. Trajektoria ruchu dla mobilnego robota z wykorzystaniem algorytmu "Lumelsky Bug 1".



Rys. 5. Trajektoria ruchu dla mobilnego robota z wykorzystaniem algorytmu "Lumelsky Bug 2".

grupa robotów ma się poruszać, tak aby jednoznacznie można było zdefiniować każdy z klastrów. Algorytm "Wavefront" jest prosty i bardzo efektywny w procesie poszukiwania najkrótszej trajektorii pod warunkiem iż dysponuje się pełną wiedzą na temat obszaru w którym porusza się robot oraz znane jest położenie celu w tym zdefiniowanym obszarze. Dużą zaletą tego algorytmu jest możliwość wykorzystania go do poszukiwania trajektorii w przestrzeniach dwu lub trójwymiarowych z dużą ilością przeszkód.

Przykładowy obszar eksploracji przez roboty mobilne przygotowany do zastosowania powyższego algorytmu przedstawiono na rys.3. Następnym



Rys. 6. Sposoby poruszania się robota na płaszczyźnie.

krokiem w procesie poszukiwania najkrótszej trajektorii jest przyjęcie sposobu poruszania się robota, wynika on z przyjętej metryki przedstawionej na rys.1 i rys.2. Tak więc można wybrać dwa sposoby poruszania się robota po obszarze dwuwymiarowym, pierwszy ruch w 4 kierunkach (aprosymacja metryki L1) lub drugi, ruch w 8 kierunkach. Oba sposoby poruszania się robota po dwuwymiarowej płaszczyźnie przedstawiono na rys.6.

W celu zilustrowania procesu planowania trajektorii dla mobilnego robota lub grupy robotów wykorzystano przykładowy obszar przedstawiony na rys.3. i przygotowano go do wyznaczenia najkrótszej drogi pomiędzy punktem startowym a punktem do osiągnięcia. Prawy dolny wiersz i lewa kolumna (zaznaczone na niebiesko) zostały zaopatrzone w numery umożliwiający jednoznaczne określenie klastra na płaszczyźnie i nie należące do obszaru eksploracji. Przez wartość 0 należy rozumieć klaster do eksploracji, natomiast wartość jeden należy przypisać w miejscach występowania przeszkody. Tak więc mapa obszaru w początkowej fazie planowania trajektorii będzie miała postać, przedstawioną na rys.7.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
4	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	2	0
3	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	

Rys. 7. Początkowa faza planowania trajektorii dla obszaru z rys.3.

Wartość równą 2 na rys.7 przypisano miejscu docelowemu, czyli miejscu gdzie roboty mają dojechać. Następnie założono ośmio-kierunkowy sposób poruszania się robota i rozpoczęto poszukiwanie trajektorii poprzez modyfikację wartości w klastrach sąsiadujących z celem

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
8	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	3	3	3
4	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	3	2	3
3	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	3	3	3
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	

Rys. 7. Modyfikacja wartości w klastrach sąsiadujących z celem

Następnie należy wykonywać modyfikację wartości klastrów dla możliwej do osiągnięcia ścieżki robota do czasu gdy wszystkie klastry z wartością 0 zostaną zmodyfikowane, a ich wartość będzie  $\geq 2$ . Poniżej na rys.8 przedstawiono mapę obszaru eksplorowanego dla trzeciego kroku robota

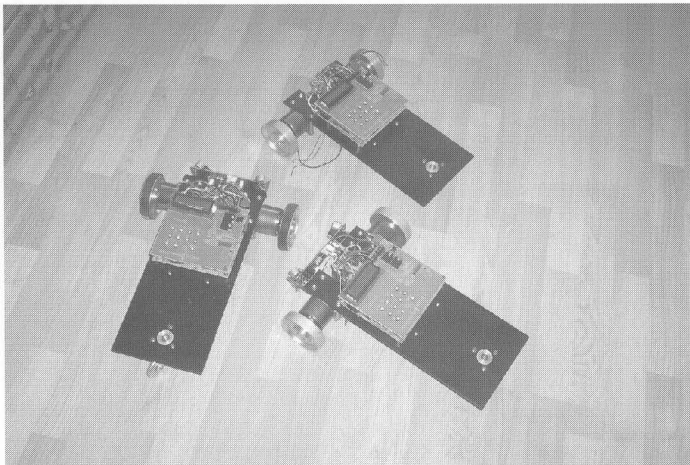
Przedstawiony algorytm można oczywiście zaimplementować na dowolnym sprzęcie komputerowym, co umożliwi błyskawiczne uzyskanie informacji o najkrótszej trajektorii robota.

Na rys.9 i rys.10 przedstawiono mapy tereny po odpowiednio kroku 8 i 17 robota



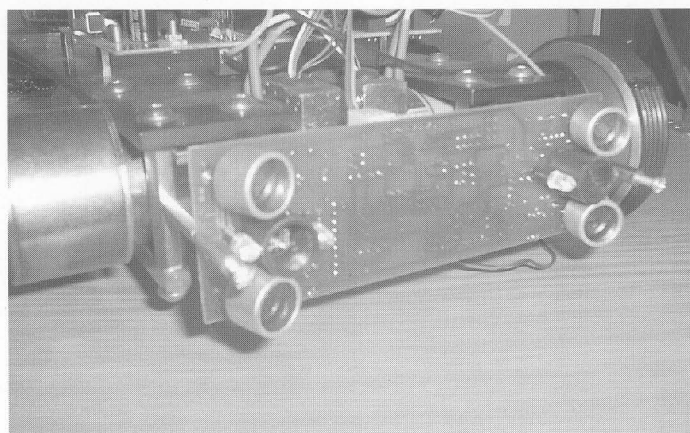
### 3. Projekt warstwy sensorycznej związanej z algorytmami planowania trajektorii

We wspomnianej grupie robotów (rys.14) specjalny nacisk w procesie projektowania mechatronicznego położono na warstwę sensoryczną, która bezpośrednio wpływa na możliwości związane ze zbieraniem informacji o otoczeniu robota.



Rys. 14. Grupa mobilnych robotów kołowych

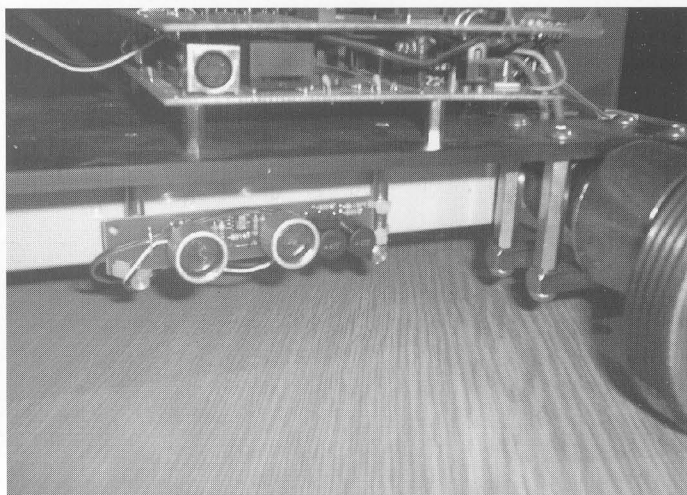
Podstawowym założeniem związanym ze zbieraniem informacji o otoczeniu przez roboty jest bezdotykowy pomiar odległości od przeszkód oraz samo ich wykrywanie. W tym celu zaprojektowano warstwę sensoryczną składającą się z układów elektronicznych wykrywających przeszkody za pomocą czujników ultradźwiękowych i czujników na podczerwień. Zastosowanie kombinacji tych dwóch rodzajów czujników umożliwia efektywniejsze wykrywanie przeszkód. Jak wiadomo przetworniki ultradźwiękowe są elementami piezoceramicznymi zbudowanymi z tworzywa piezoelektrycznych które pod wpływem przyłożonego napięcia zmieniają swoje rozmiary fizyczne. Przetworniki ultradźwiękowe są elementami składającymi się z połączonych ze sobą odwrotnie spolaryzowanych dwóch płytek. Po przyłożeniu do nich napięcia płytki wyginają się powodując zmianę ciśnienia akustycznego. Jeżeli napięcie zmienia się z częstotliwością równą częstotliwości rezonansu mechanicznego takiego piezoelektryka, wówczas sprawność emitowania ultradźwięków jest największa i wynosi około 5-10%.



Rys. 15. Widok mobilnego robota z przodu

W projekcie zdecydowano się wykorzystać metodę, w której nadajnik i odbiornik znajdują się obok siebie (rys.15 i rys.16), a więc w tej samej odległości od przeszkody odbijającej wysłany sygnał. Do odbiornika dociera sygnał z nadajnika po odbiciu od przeszkody. Wysłany sygnał przebywa mierzoną drogę dwukrotnie: do przeszkody i po odbiciu od niej.

Duże znaczenie ma tutaj kształt i materiał z którego jest wykonana przeszkoda, która skierowana pod kątem ostrym do nadajnika może spowodować odbicie fali ultradźwiękowej w innym kierunku niż odbiornik. Przeszkody wykonane z materiałów pochłaniających dźwięk mogą również zakłócić pomiar, tłumiąc sygnał z nadajnika. "Sercem" układu sterującego zestawem czujników ultradźwiękowych jest mikroprocesor DS89C420 produkcji DALLAS-MAXIM. Jest to mikroprocesor kompatybilny z rodziną mikroprocesorów '51 posiadający ten sam zestaw in-



Rys. 16. Widok mobilnego robota z prawej strony

strukcji lecz z całkowicie przeprojektowanym rdzeniem dzięki czemu jeden rozkaz wykonuje do 12 razy szybciej niż standardowa '51 z tym samym rezonatorem kwarcowym. Maksymalna prędkość pracy tego mikroprocesora wynosi 33 MIPS. Mikroprocesor zaopatrzony jest w 64 kB pamięci programu FLASH dzięki czemu możliwe jest jego wielokrotne programowanie (gwarantowane do 1000 razy). Wbudowany w mikroprocesor bootloader umożliwia programowanie mikroprocesora w systemie (ISP).

W przypadku czujników na podczerwień wiązka podczerwieni emitowana jest przez diody IR. Każdorazowe zbliżenie się do przeszkody powoduje padanie wiązki podczerwieni wysłanej za pomocą diod a odbitej od przeszkody na układ odbiornika OPT101 i zwiększenie napięcia wyjściowego odbiornika. Napięcie to jest podawane na wejście analogowe komparatora MAX516 i porównywane z napięciem progowym podanym przez mikroprocesor. Układ OPT101 zawiera w swojej strukturze fotodiody i wzmacniacz transimpedancyjny (przetwornik prądu na napięcie) zamieniający prąd płynący przez diodę podczas oświetlenia jej promieniami IR na napięcie. Układ posiada wewnętrzny rezystor sprzężenia zwrotnego decydujący o wzmacnieniu wzmacniacza.

### 4. Podsumowanie

Zastosowanie bezdotykowych sensorów zbierających informację o otoczeniu robota w połączeniu z prostymi algorytmami generowania trajektorii umożliwia skuteczne sterowanie behawioralne grupą mobilnych robotów kołowych. Istotnym zagadnieniem w procesie projektowania tego typu urządzeń jest mechatroniczne podejście do projektu.

### 5. Literatura

- [1] LUMELSKY V. J., STEPANOV A.A.: Path-planning strategies for a point mobile automatio moving amidst unknown obstacles of arbitrary shape, Algorithmica, 1987.
- [2] ALEXANDER R. S., ROWE, N. C.: Path planning by optimal-path-map construction for homogeneous-cost two-dimensional regions. Proceedings of IEEE International Conference on Robotics and Automation, 1990, USA.
- [3] CHEN D. Z., KLENK K. S., TU, H. T.: Shortest path queries among weighted obstacles in the rectilinear plane. Proceedings of the 11th Annual ACM Symposium on Computational Geometry, 1995, Vancouver, Canada.
- [4] HERSHBERGER J., SURI S.: On computing euclidean shortest paths in the plane. Proceedings of the 34th Annual IEEE Symposium on the Foundation of Computer Science, 1993.

**Title:** The Use of path planning algorithms for wheeled mobile robots.