

Maciej Garbacz*, Mieczysław Zaczek*

Metoda pól potencjałowych w nawigacji kołowego robota mobilnego

1. Wprowadzenie

W artykule zostanie przedstawiona implementacja algorytmu wykorzystującego metodę „sztucznych pól potencjałowych” [1, 2] do realizacji ruchu i nawigacji kołowego robota mobilnego Khepera III w nieznanym otoczeniu, z omijaniem występujących w nim przeszkód. Metoda pól potencjałowych należy do grupy metod lokalnych, pozwalających na realizację zadań „idź do celu” bez wcześniejszej znajomości mapy otoczenia. Aby robota można było traktować jako jednostkę autonomiczną, konieczne jest wyposażenie go w czujniki zbierające informacje o otoczeniu [3]. Do prawidłowego działania algorytmu realizującego określony cel, konieczna jest znajomość przestrzeni otaczającej robota. W opisanej w artykule aplikacji do rozpoznawania otoczenia zostały wykorzystane czujniki zbliżeniowe podczerwieni. Metoda została zmodyfikowana na potrzeby implementacji w rzeczywistym robocie mobilnym. W zastosowaniach praktycznych, robota nie można traktować jako punkt materialny. Dlatego wyznaczone siły odpychające nie odnoszą się bezpośrednio do przeszkód (jak to jest w założeniach teoretycznych metody), a przypisane są do czujników. Na potrzeby implementacji przyjęte zostało, że wokół robota znajduje się zawsze 9 teoretycznych przeszkód rozlokowanych zawsze pod takimi samymi kątami (wynikającymi z rozmieszczenia czujników) w stosunku do jego osi, działających z zerową lub większą od zera siłą. Przedstawiony algorytm ma za zadanie realizację ruchu robota do zadanego położenia, z omijaniem występujących w obszarze roboczym przeszkód. W zadaniach nawigacji możliwe jest również wykorzystywanie dodatkowych urządzeń zewnętrznych takich, jak np. kamera umieszczona ponad obszarem roboczym. Możliwe jest wówczas zbudowanie mapy otoczenia na podstawie informacji uzyskanych z obrazu z takiej kamery. Zaprezentowany w niniejszej pracy algorytm „sztucznych pól potencjałowych” bazuje jedynie na informacjach pochodzących z czujników zbliżeniowych. Robot na bieżąco śledzi sytuację w otaczającej go przestrzeni, dzięki czemu zdolny jest do reakcji w przypadku zagrożenia kolizji np. z drugim poruszającym się robotem.

* AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki, al. A. Mickiewicza 30, 30-059 Kraków

2. Robot Khepera III

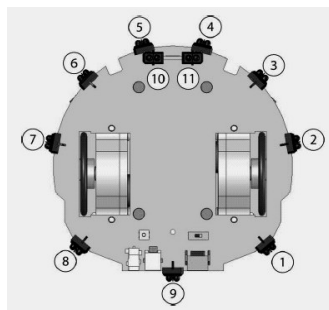
2.1. Budowa

Robot (rys. 1) porusza się na dwóch kółkach, umieszczonych w jednej osi, obleczone gumą dla uzyskania lepszej przyczepności. Dodatkowo Khepera III podparty jest w jednym punkcie, co zapewnia stabilne poruszanie się w przestrzeni roboczej. Dookoła obudowy robota umieszczonych jest 9 czujników podczerwieni oraz dodatkowo 2 pod spodem (czujniki zbliżeniowe), umożliwiających wykrywanie na niedaleką odległość przeszkód, bądź krawędzi stołu, po którym robot się porusza. Robot ma również możliwość zmierzenia odległości od przeszkody, za sprawą pięciu wbudowanych sensorów ultradźwiękowych (sonary). Do napędu wykorzystano dwa wysokiej klasy silniki DC (jeden dla każdego koła) zapewniające sprawne i dokładne sterowanie ruchem robota. Obydwa koła robota napędzane są silnikami DC sprzężonymi z przekładnią o przełożeniu 43,2:1. Silniki mają własne wbudowane enkodery przyrostowe, umiejscowione na osi silnika, dające 16 impulsów na obrót wału silnika. W sumie dają to 2764,8 impulsów na obrót koła, co odpowiada 21,47 impulsom na milimetr przejechanej drogi (średnica koła wynosi 41 mm, co daje 128,8 mm przejechanej drogi na pełny obrót tarczy koła). Maksymalna osiągalna prędkość robota wynosi 298 mm/s [11].



Rys. 1. Widok robota Khepera III

Rozmieszczenie czujników zbliżeniowych przedstawia rysunek 2. Czujniki zbliżeniowe nie dają informacji o odległości od przeszkody, jedynie o jej bliskości (odczyt rośnie w miarę zbliżania się do przeszkody). Czujniki podczerwieni mogą również pracować jako czujniki światła [11].



Rys. 2. Rozmieszczenie czujników zbliżeniowych (widok od spodu)

Podstawowym sposobem komunikacji pomiędzy komputerem a robotem mobilnym Khepera III jest protokół komunikacji szeregowej RS232. Podczas takiej komunikacji komputer pracuje jako ‘master’ a robot Khepera jako ‘slave’. Każde połączenie z robotem jest inicjowane przez komputer, a komunikacja realizowana jest przez przesyłanie komunikatów ASCII. Każde pojedyncze połączenie składa się z dwóch części:

- 1) rozkazu wysyłanego z komputera: rozpoczynającego się dużą literą, po której następują (jeśli są konieczne) numeryczne lub znakowe parametry oddzielone przecinkiem,
- 2) odpowiedzi, wysyłanej z robota do komputera: rozpoczynającej się małą literą (taką jak w rozkazie), po której następują (jeżeli rozkaz dotyczy odczytu stanu czujników) numeryczne parametry odpowiedzi oddzielone przecinkami.

Dostępne rozkazy można podzielić na dwie grupy:

- 1) rozkazy dotyczące konfiguracji robota (ustawienie parametrów protokołu szeregowego, ustawianie parametrów regulatorów położenia i prędkości, ustawianie parametrów sonarów),
- 2) rozkazy związane ze sterowaniem robota (zadawanie pozycji, zadawanie prędkości, odczyt czujników zbliżeniowych, odczyt czujników światła, odczyt odległości z sonarów).

Taki sposób komunikacji z robotem Khepera umożliwia programowanie przy użyciu dowolnego oprogramowania udostępniającego łączność poprzez port szeregowy COM. Robot wyposażony jest standardowo w urządzenie Bluetooth, zapewniające bezprzewodową komunikację poprzez port szeregowy pomiędzy nim a komputerem sterującym.

2.2. Reprezentacja położenia robota

Przyjęto, że współrzędne pozycji Khepery odpowiadają lokalizacji punktu przecięcia osi robota z osią jego kół. Do nawigowania robotem nie wystarcza znajomość jego położenia. Aby móc skierować go w odpowiednim kierunku, trzeba również dysponować wiedzą o jego aktualnej orientacji. Dlatego współrzędne robota są reprezentowane przez 3 zmienne (x, y, α). Aktualna orientacja jest mierzona względem startowej w stronę przeciwną do ruchu wskazówek zegara. Przy wykonywaniu obrotów wokół pionowej osi robota koła przesuwają się po okręgu. Ze względu na określoną odległość między kołami robota, na pełny obrót przypada 5934,45 impulsów z enkoderów. Kierunek, w którym porusza się robot, wyznaczany jest z różnicy wartości wskazywanych przez enkodery na obu kołach:

$$\alpha = \frac{enk_p - enk_l}{5934,45} \cdot 180^\circ \quad (1)$$

Dla określenia aktualnej pozycji robota potrzebna jest wartość enkoderów przed rozpoczęciem i po zakończeniu ruchu. Pokonany dystans jest wyznaczany poprzez porównanie średnich arytmetycznych wskazań enkoderów przed rozpoczęciem i po zakończeniu ruchu.

$$dist = \frac{enk_l_n + enk_p_n}{2} - \frac{enk_l_{n-1} + enk_p_{n-1}}{2} \quad (2)$$

Aktualne położenie robota wyznaczone jest z zależności:

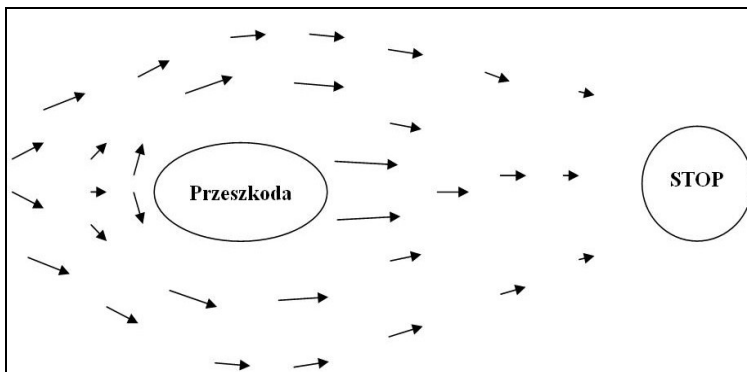
$$x_n = x_{n-1} + \frac{dist \cdot \cos(\alpha)}{21,47} \quad (3)$$

$$y_n = y_{n-1} + \frac{dist \cdot \sin(\alpha)}{21,47}$$

gdzie dzielenie przez 21,47 wynika z przeliczenia liczby impulsów z enkoderów na [mm] przebytej drogi.

3. Metoda pól potencjałowych

W literaturze metoda ta [1, 2] spotykana jest także pod nazwą „metoda sztucznych pól potencjalnych”. Podstawowym założeniem metody jest przyjęcie, iż robot, traktowany jako punkt materialny, porusza się w wyniku sił działających na niego. Siły pochodzą od przeszkód i od punktu docelowego. Siła od punktu docelowego przyciąga robota, natomiast siły od przeszkód odpychają go. Charakterystyka sił odpychających powinna być podana w taki sposób, aby wartość siły była mała, gdy robot znajduje się daleko od przeszkody i rosła praktycznie do nieskończoności na brzegach przeszkód. Siły, w formie wektorów, które działają na robota, są sumowane, tworząc wektor wypadkowy w danym punkcie przestrzeni ruchu. Jest to nowy kierunek, w którym powinien się poruszać robot (rys. 3).



Rys. 3. Rozkład wektorów sił wypadkowych działających na robota

Ze względu na to, że punkt porusza się po płaszczyźnie, a każde jego położenie opisane jest przez $q = (x, y)$, wypadkowe pole działające na cząstkę jest superpozycją pola przyciągającego i pól odpychających, zależnych od q :

$$U_w(q) = U_p(q) + \sum_i V_{oi}(q) \quad (4)$$

gdzie:

- U_w – potencjał wypadkowy,
- U_p – potencjał przyciągający,
- V_{oi} – potencjał odpychający od i -tej przeszkody.

Dla opisu potencjału przyciągającego, najczęściej używana jest funkcja kwadratowa postaci [1, 2]:

$$U_p(q) = \frac{1}{2} k_p \|q_r - q_k\|^2 \quad (5)$$

gdzie:

- q_r – współrzędne położenia robota,
- q_k – współrzędne położenia punktu końcowego,
- k_p – doświadczalnie dobierany współczynnik.

Dla tak zaproponowanego potencjału siła przyciągania wynosi:

$$F_p(q_r) = -\nabla U_p = k_p \|q_r - q_k\| \quad (6)$$

Ze względu na to, że siła ta będzie w każdym punkcie płaszczyzny proporcjonalna do odległości między położeniem aktualnym a docelowym cząstki, wykorzystuje się ją jako kryterium osiągnięcia celu. Przyjmuje się, że jeżeli wartość siły spadnie poniżej pewnej stałej, to punkt końcowy został osiągnięty.

Funkcją potencjału odpychającego ma być odpychanie ładunku od przeszkód, gdy znajduje się on blisko nich. W miarę zwiększania się odległości cząstki jego wpływ ma maleć.

Może być zatem opisany funkcją [2]:

$$V_{oi}(q) = \begin{cases} \frac{1}{2} k_{oi} \left(\frac{1}{\|q_r - q_{oi}\|} - \frac{1}{d_0} \right)^2, & \|q_r - q_{oi}\| < d_0 \\ 0, & \|q_r - q_{oi}\| > d_0 \end{cases} \quad (7)$$

gdzie:

- q_r – współrzędne położenia robota,
- q_{oi} – współrzędne położenia i -tej przeszkody,
- k_{oi} – doświadczalnie dobierane współczynniki,
- d_0 – odległość od przeszkody, poza którą potencjał przestaje wpływać na cząstkę.

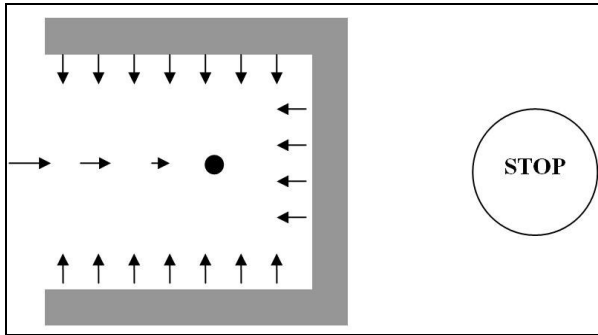
Dla tak zaproponowanego potencjału siła odpychania wynosi:

$$F_{oi}(q_r) = \begin{cases} -k_{oi} \left(\frac{1}{\|q_r - q_{oi}\|} - \frac{1}{d_0} \right) \frac{1}{\|q_r - q_{oi}\|^2}, & \|q_r - q_{oi}\| < d_0 \\ 0, & \|q_r - q_{oi}\| > d_0 \end{cases} \quad (8)$$

Jak widać, siła odpychająca będzie rosłać do nieskończoności, gdy ruchomy ładunek znajdzie się bardzo blisko przeszkody.

Mając wszystkie wektory sił odpychających i wektor siły przyciągającej, można utworzyć wypadkowy wektor, który nadaje chwilowy kierunek jazdy robota.

Istnieje prawdopodobieństwo, że robot utknie w minimum lokalnym, tzn. siła wypadkowa będzie równa 0, lecz robot nie znajduje się jeszcze u celu. Jest to główna wada tej metody. Klasyczny przykład takiej sytuacji ilustruje rysunek 4.



Rys. 4. Minimum lokalne metody pól potencjałowych

Niemożliwe jest wyjście robota z minimum lokalnego, jeśli używa się algorytmu pól potencjałowych. Można natomiast posłużyć się dodatkowymi metodami pozwalającymi kontynuować jazdę. Jedną z możliwości jest ruch robota wzdłuż ściany przeszkody, do czasu aż istnieje pewność ominięcia przeszkody. Kolejnym sposobem jest chwilowe wyłączenie siły przyciągającej [3]. Takie działanie nie jest działaniem idealnym, ponieważ trudno wyznaczyć obszar atrakcji minimum lokalnego, a co za tym idzie, robot może ponownie wrócić do minimum. Najczęstszym działaniem jest chwilowe wyłączenie algorytmu pól potencjałowych [3] i włączenie np. algorytmu Braitenberga [4], czyli omijania przeszkód i błędzenia losowego.

4. Implementacja metody pól potencjałowych

Podczas realizacji zadania sterowania robotem nie jest on traktowany jako omawiany w metodzie ładunek elementarny. Przeszkody, wykrywane przez czujniki podczerwieni, generują siły odpychające. Wskazania czujników podczerwieni nie podają bezpośrednio

odległości od przeszkody. Aby powiązać je z wartościami sił wynikającymi ze wzoru (8), należy je odpowiednio przeskalować.

Ze względu na podobny charakter przebiegu wykresu odczytów z czujników oraz wykresu wartości siły (przy dobranych k_{oi} oraz d_0) zastosowane zostały proste operacje arytmetyczne przekształcające odczyt z czujników do charakterystyki siły (rys. 5).

$$F_i = k_i \cdot \frac{\delta_i^2}{4095} - \eta \quad (9)$$

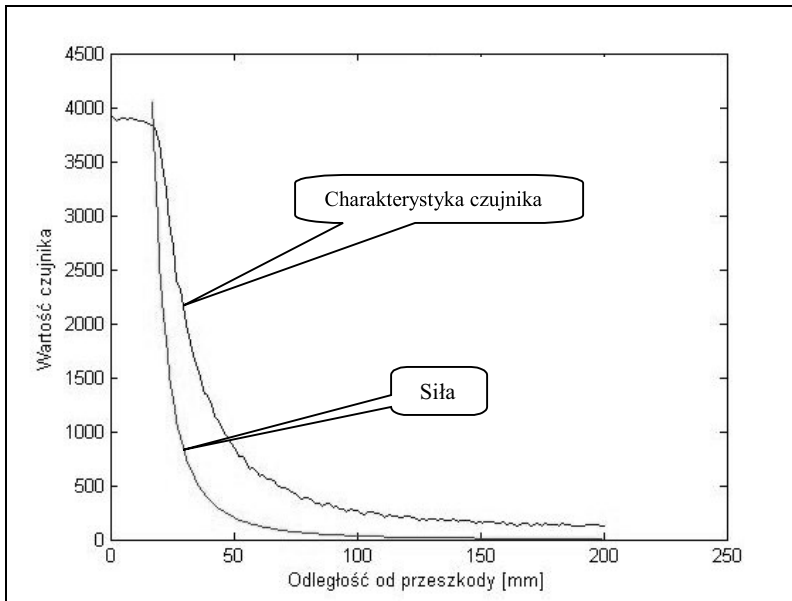
gdzie:

δ_i – pomiar i -tego czujnika,

η – stała dobrana doświadczalnie,

k_i – współczynnik wzmocnienia i -tej siły (dobry doświadczalnie).

Liczba 4095 wynika z maksymalnego odczytu z czujnika.



Rys. 5. Dopasowanie siły do charakterystyki czujnika

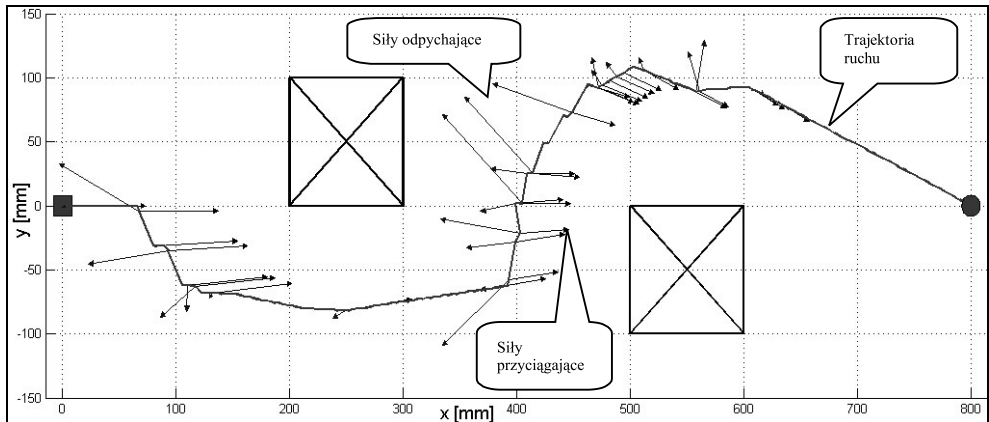
W zastosowaniach praktycznych robota nie można traktować jako punkt materialny. Dlatego dobrane doświadczalnie współczynniki k_i nie odnoszą się bezpośrednio do przeszkód (jak to jest w założeniach teoretycznych), a przypisane są do czujników.

Na potrzeby implementacji przyjęto, że wokół robota znajduje się zawsze 9 teoretycznych przeszkód rozlokowanych zawsze pod takimi samymi kątami (wynikającymi z rozmieszczenia czujników) w stosunku do jego osi, działających z zerową lub większą od zera siłą.

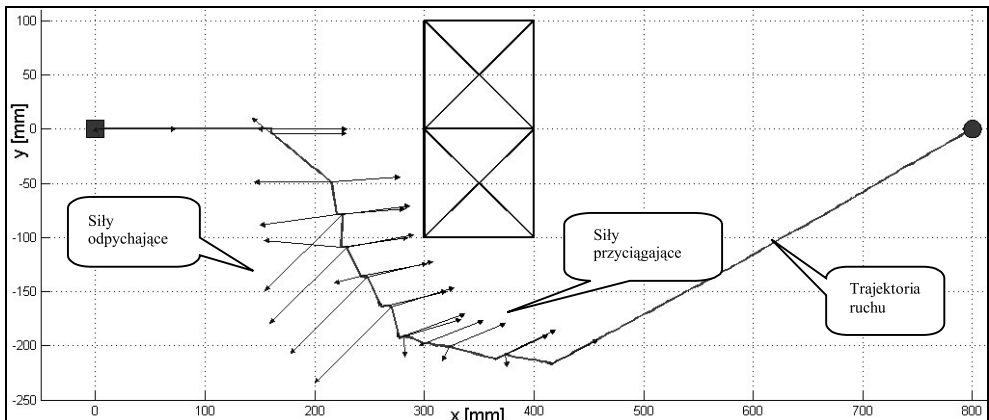
Na podstawie wskazania czujników wyznaczany jest wypadkowy wektor sił odpychających, z położenia robota na płaszczyźnie – wektor siły przyciągającej. Wektor wypadkowy siły przyciągającej i siły odpychającej wyznacza chwilowy kierunek ruchu robota (położenie i orientację).

Opisany algorytm został zaimplementowany dla robota Khepera III (z dodatkową płytą sterującą KoreBot LE), jako program napisany w języku C, rozwijany w środowisku KoreBot IDE [8, 9, 10]. Aplikacja jest ładowana do pamięci robota zapewniając autonomię jego działania.

Przykładowe przebiegi ilustrujące działanie metody przedstawiono na rysunkach 6 i 7, gdzie pokazana jest zarejestrowana trasa przejazdu robota oraz wektory sił odpychających i przyciągających.



Rys. 6. Ilustracja działania metody przy dwóch przeszkodach na trasie (mapa)



Rys. 7. Ilustracja działania metody dla przeszkody typu ściana (mapa)

5. Podsumowanie

Przedstawiona w pracy metoda „pól potencjałowych” należy do grupy metod lokalnych, pozwalających na realizację zadań „iść do celu”, bez wcześniejszej znajomości mapy otoczenia. Zaprezentowana w niniejszej pracy implementacja metody bazuje jedynie na informacjach pochodzących z czujników zbliżeniowych. Zapewnienie dużej dokładności osiągnięcia zadanej pozycji końcowej nie było przedmiotem przedstawionych badań. Przy wyznaczaniu aktualnej pozycji robota metodami względnymi (pomiar odometryczny) mogą pojawiać się błędy lokalizacji wynikające z poślizgów kół. Podczas przeprowadzonych testów na stole roboczym o wymiarach 1400 mm × 900 mm błąd w pozycji końcowej nie przekraczał 25 mm (długość trasy oraz rodzaj powierzchni będą wpływać na wielkość błędu).

Otrzymana trajektoria ruchu robota nie jest optymalna w sensie długości, co wynika z braku informacji o położeniu przeszkód przed rozpoczęciem ruchu. Metoda ta może być wykorzystywana także jako globalna, gdy znana jest mapa otoczenia.

Przeprowadzone testy potwierdziły (mimo wad związanych z minimami lokalnymi) przydatność metody do realizacji zadania nawigacji robotów mobilnych w nieznanym otoczeniu.

Literatura

- [1] Khatib O., *Real-time obstacle avoidance for manipulators and mobile robots*. International Journal of Robotics Research, vol. 5, No. 1, 1995, 90–98.
- [2] Miguel A., Castaneda P., Savage J., Hernandez A., Cosío F.A., *Local Autonomous Robot Navigation using Potential Fields*. 2008, http://sciyo.com/articles/show/title/local_autonomous_robot_navigation_using_potential_fields.
- [3] Dulęba I., *Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych*. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.
- [4] Braitenberg V., *Vehicles: Experiments in synthetic psychology*. MIT Press., Cambridge 1984.
- [5] Garbacz M., Zaczyk M., *Robot mobilny Khepera III – oprogramowanie dla środowiska MATLAB*. Automatyka (półrocznik AGH), t. 12, z. 3, 2008, 759–767.
- [6] Garbacz M., Zaczyk M., *Algorytmy rozmyte w nawigacji kołowego robota mobilnego*. Automatyka (półrocznik AGH), t. 14, z. 3/1, 2010, 429–438.
- [7] Giergiel J., Hendzel Z., Żylski W., *Kinematyka, dynamika i sterowanie mobilnych robotów kołowych*. PWN, Warszawa 2002.
- [8] Carron T., *Developing Korebot applications on Windows*. <http://ftp.k-team.com/korebot/toolchain-win32/KB-Tutorial.pdf>; 2008.
- [9] *KoreBot. User Manual*. K-Team S.A., Switzerland, 2008.
- [10] Jania K., *Oprogramowanie sterujące dla robota kołowego Khepera III*. AGH, Kraków 2009 (praca dyplomowa).
- [11] *Khepera III User Manual*. ver. 2.1, K-Team S.A., Switzerland, 2008.