

Maciej Garbacz*, Mieczysław Zaczek*

Robot mobilny Khepera III – oprogramowanie dla środowiska MATLAB**

1. Wprowadzenie

1.1. Opis stanowiska

W artykule zostanie przedstawione nowe stanowisko zbudowane w Laboratorium Automatyki, Robotyki i Systemów Fotofoltaicznych w Katedrze Automatyki AGH. Powstało ono w oparciu o najnowszy zakup Laboratorium – roboty mobilne Khepera III szwajcarskiej firmy k-Team. Roboty te są kolejną, ulepszoną wersją modelu Khepera II [1], znajdującego się również na wyposażeniu Laboratorium. Nowy model posiada znacznie większe możliwości od swojego poprzednika. Poprzez łatwo wymienialne akumulatory możliwa jest praktycznie ciągła praca z robotami, ponieważ zostały zakupione dodatkowe akumulatory wraz z ładowarką. Komunikacja odbywa się bezprzewodowo poprzez port szeregowy w oparciu o technologię Bluetooth, dzięki czemu unika się kłopotliwego połączenia kablowego. Oprócz czujników w paśmie podczerwieni, będących również na wyposażeniu Khepera II, nowy model posiada ultradźwiękowe czujniki odległości, co umożliwi zebranie większej ilości informacji o przestrzeni roboczej otaczającej robota.

1.2. Robot Khepera III

Robot Khepera III, podobnie jak poprzedni model, ma budowę modułową. Robot ma okrągły kształt, co minimalizuje skutki ewentualnych kolizji dwóch robotów ze sobą, na skutek błędnie działającego algorytmu. Robot napędzany jest poprzez dwa symetrycznie rozmieszczone kółka, które z kolei połączone są z silnikami prądu stałego poprzez przekładnie redukcyjne. Zastosowane napędy pozwalają na uzyskiwanie prędkości w zakresie od 14 mm/s do 298 mm/s. Każdy silnik wyposażony jest w enkoder zliczający, dzięki któremu użytkownik dysponuje informacją o pozycji robota w przestrzeni roboczej. Oprócz pomiarów z enkoderów na wale silników, dodatkowych informacji o przestrzeni roboczej dostarczają czujniki ultradźwiękowe i podczerwieni. Wygląd robota przedstawiono na rysunku 1.

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

** Artykuł zrealizowany w ramach umowy BS nr 11.11.120.231



Rys. 1. Robot Khepera III

Widoczne u góry większe czujniki to czujniki ultradźwiękowe, poniżej dookoła rozmieszczone są czujniki w paśmie podczerwieni. Robot zasilany jest z wymiennych akumulatorów litowo-polimerowych o pojemności 1400 mAh, dostarczających napięcie 7,4 V. Przy standardowym trybie pracy, bez dołączania dodatkowego wyposażenia robot może pracować na jednej baterii do ok. 8 h. Robot nie jest standardowo wyposażony w system zarządzania energią. Decyzję o wymianie akumulatora musi podjąć użytkownik, obserwując zachowanie robota. Użytkownik może uzyskać informację o aktualnym napięciu akumulatora, co jest pomocne przy określeniu stopnia jego rozładowania. Na wyposażeniu laboratorium znajdują się wymienne akumulatory wraz z zewnętrzną ładowarką. Umożliwia to ciągłe wykorzystywanie robota – w razie rozładowania akumulatora zastępuje się go uprzednio naładowanym egzemplarzem tego samego typu. Dzięki temu została usunięta niedogodność, która w przypadku poprzedniego modelu – Khepera II wymuszała ciągnięcie kabla za robotem. Robot przykryty jest z wierzchu pokrywą zabezpieczającą – po jej zdjęciu możliwe jest dołączenie wyposażenia dodatkowego. Podstawowe cechy robota zebrano w tabeli 1.

Tabela 1
Cechy charakterystyczne robota Khepera III

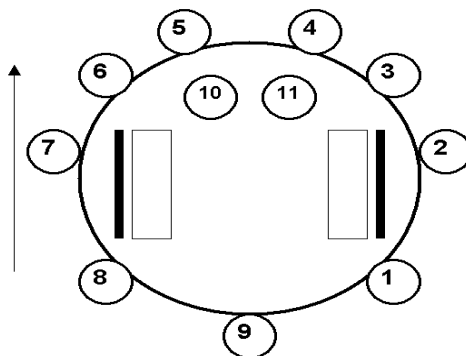
Procesor	DsPIC 30F5011, 60MHz
RAM	4 KB na DsPIC
Flash	66 KB na DsPIC
Napęd	2 silniki prądu stałego z enkoderami zliczającymi
Czujniki	9 czujników podczerwieni dookoła robota (odległości do 25 cm) 2 czujniki podczerwieni pod spodem 5 czujników ultradźwiękowych dookoła robota (zakres odległości 20 cm±4 m)
Zasilanie	Akumulator litowo-polimerowy (1400 mAh)
Autonomia	maks. 8h, bez dodatkowych akcesoriów
Komunikacja	Port szeregowy 115200 bps
Rozmiary	Średnica: 130 mm Wysokość: 70 mm
Masa	690 g
Zdalne sterowanie	LabVIEW, MATLAB, SysQuake lub inne dowolne oprogramowanie umożliwiające komunikację poprzez RS232

1.3. Czujniki ultradźwiękowe

Robot Khepera III został wyposażony w 5 czujników ultradźwiękowych typu *400ST100/400SR100* firmy *Midas* [2]. Fizycznie każdy czujnik składa się z nadajnika *400ST100* oraz odbiornika *400SR100*. Czujniki te zasilane są napięciem 20 Vdc, ich nominalna częstotliwość to $40 \text{ kHz} \pm 1 \text{ kHz}$. Użytkownik ma możliwość zaprogramowania liczby zwracanych ech oraz liczby aktywnych sonarów – zostanie to dokładnie opisane w rozdziale 2. Przeprowadzone doświadczenia pokazują, że czujniki ultradźwiękowe lepiej sprawdzają się przy wykrywaniu przeszkód znajdujących się w dalszej odległości (powyżej 40 cm). Zaimplementowany fabrycznie program demonstracyjny oparty na algorytmie Braitenberga w wersji wykorzystującej sonary nie działa poprawnie ze względu na zbyt bliskie odległości od przeszkód na stole znajdującym się na wyposażeniu laboratorium.

1.4. Czujniki podczerwieni

W przypadku poruszania się robota w bliskich odległościach od przeszkód, podstawowym źródłem informacji o otaczających przeszkodach są czujniki zbliżeniowe pracujące w paśmie podczerwieni. Dookoła robota rozmieszczone jest 9 czujników (rys. 2). Dodatkowo pod spodem robota umieszczono 2 czujniki, umożliwiające wykrywanie np. krawędzi blatu. W robocie wykorzystano czujniki typu *TCRT5000* firmy *Vishay Telefunken*. Czujniki te składają się z nadajnika i odbiornika światła. Pozwalają na pomiar wyłącznie oświetlenia (wykorzystując tylko odbiornik) bądź też na pomiar odległości od przeszkody (w tym przypadku wykorzystuje się zarówno nadajnik jak i odbiornik). Zwracany pomiar jest w postaci 12-bitowej liczby. Należy zwrócić uwagę, że wartość mierzona wzrasta wraz ze zmniejszaniem się odległości do przeszkody. Przykładowe charakterystyki czujników dla różnych materiałów, z których są wykonane przeszkody, zostaną przedstawione w rozdziale 2, dotyczącym programowania robota. Przy programowaniu należy wziąć pod uwagę fakt, że pomiar jest zależny od warunków oświetlenia panujących w laboratorium, od materiału z którego wykonane są przeszkody, a nawet od ich koloru. Program demonstracyjny z algorytmem Braitenberga w wersji wykorzystującej czujniki podczerwieni działa poprawnie w warunkach małych odległości od przeszkód.



Rys. 2. Rozmieszczenie czujników podczerwieni

2. Oprogramowanie w środowisku MATLAB

2.1. Środowisko MATLAB

Żeby ułatwić komunikację z robotem, została napisana biblioteka zawierająca gotowe instrukcje do użycia w środowisku MATLAB-a. Komunikacja możliwa jest w każdym środowisku obsługującym port szeregowy, jednak ze względu na popularność MATLAB-a w Katedrze Automatyki zdecydowano się stworzyć bibliotekę pod to środowisko. W przeciwieństwie do poprzedniej wersji – Khepera II, biblioteka taka nie została dostarczona przez producenta wraz z robotem.

2.2. Instrukcje środowiska MATLAB

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 poprzez 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. Na bazie udostępnionych dla robota rozkazów zrealizowany został zestaw funkcji pozwalających na programowanie w środowisku MATLAB-a:

- `kopen('COMx')` – gdzie ‘x’ – numer portu: otwiera port szeregowy COMx dla komunikacji z robotem oraz ustawia parametry protokołu (115200bps, 8 bitów danych, 1 bit stopu, brak parzystości, brak kontroli przepływu).

Użycie funkcji $ref = kopen('COM5')$ powoduje otwarcie połączenia z robotem poprzez port COM5 i przypisanie pod zmienną 'ref'. Dalsza komunikacja z robotem odbywa się poprzez zmienną 'ref'.

- $kclose(ref)$ – funkcja zamykająca połączenie z robotem.

Funkcje służące do ustawiania konfiguracji robota:

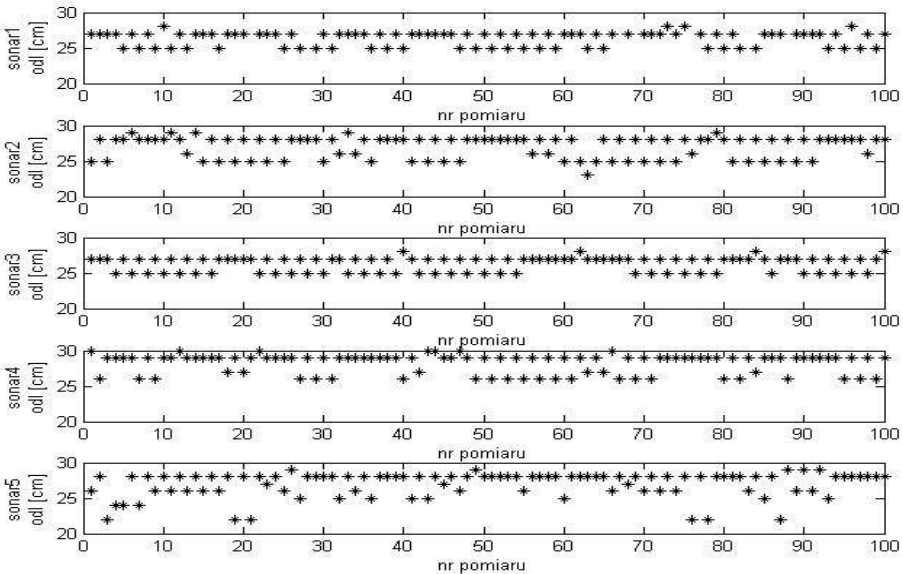
- $kConfSensor(ref, n_sens)$ – ustawianie ilości aktywnych sonarów (standardowo ustawiony jest tylko środkowy sonar);
- $kConfSensEcho(ref, n_echo)$ – ustawianie ilości ech zwracanych przez każdy z aktywnych sonarów (standardowo zwracane są 3 echa);
- $kInitMotors(ref)$ – inicjalizacja i reset silników;
- $kBatteryState(ref, index)$ – odczyt stanu akumulatora; index – wybiera: napięcie, prąd, średni pobór prądu, pojemność lub temperaturę.

Funkcje służące do sterowania robotem:

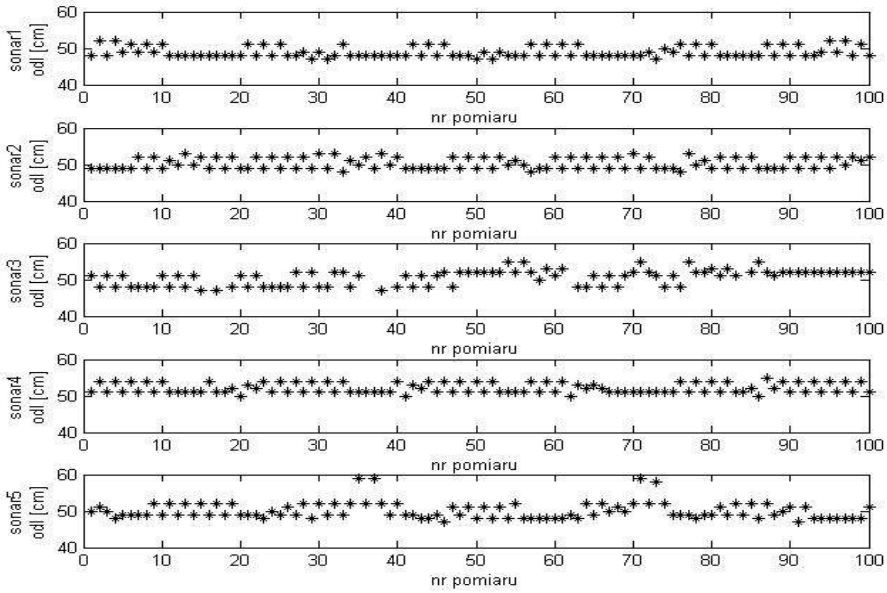
- $kAmbient(ref)$ – funkcja zwraca wektor 11 wartości odczytanych z czujników światła;
- $kProximity(ref)$ – zwraca wektor 11 wartości odczytanych z czujników zbliżeniowych (wartość największa blisko przeszkody);
- $kGetMeasure(ref, us_numb)$ – odczyt odległości w [cm] z wybranego czujnika ultradźwiękowego (us_numb);
- $kReadPos(ref)$ – odczyt położenia z liczników enkoderów lewego i prawego koła; wartość zwracana jest w postaci liczby impulsów (pojedynczy impuls odpowiada odległości 0,047 mm);
- $kReadSpeed(ref)$ – odczyt prędkości silnika lewego i prawego koła;
- $kSetPos(ref, left, right)$ – ustawienie zawartości liczników enkoderów pozycji;
- $kSetSpeed(ref, left, right)$ – zadawanie prędkości dla lewego i prawego koła (utrzymanie zadanej prędkości przez regulator PID);
- $kSetSpeedProfile(ref, max_speed, acceleration)$ – ustawianie parametrów trapezoidalnego profilu zmian prędkości (przyspieszenie i maksymalna prędkość);
- $kSetTargetPos(ref, left, right)$ – zadawanie pozycji, do jakiej ma dojechać robot (pozycja dla lewego i prawego koła); pozycja zadawana jest w impulsach (pojedynczy impuls odpowiada odległości 0,047 mm); ruch robota jest realizowany bez kształtowania przyspieszenia;
- $kSetTargetProfile(ref, left, right)$ – zadawanie pozycji do jakiej ma dojechać robot; pozycja zadawana w impulsach; ruch jest realizowany z zadaniem przyspieszeniem i prędkością maksymalną (ustalaną rozkazem $kSetSpeedProfile$);
- $kSetPWM(ref, left, right)$ – zadawanie prędkości dla silników robota, bez działania regulatora prędkości (zadawanie współczynnika wypełnienia PWM);

- $kStop(ref)$ – zatrzymanie robota;
- $kSetPosPID(ref, kp, ki, kd)$ – ustawianie parametrów (kp , ki , kd) dla regulatora położenia;
- $kSetSpeedPID(ref, kp, ki, kd)$ – ustawianie parametrów (kp , ki , kd) dla regulatora prędkości.

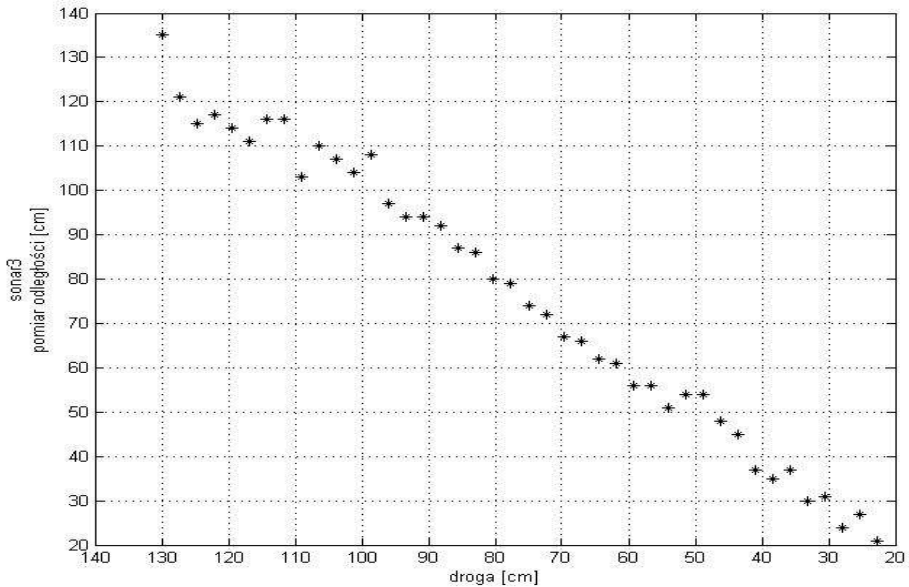
Zrealizowane oprogramowanie wykorzystane zostało do przetestowania powtarzalności pomiarów z czujników położenia i czujników zbliżeniowych robota. Na rysunku 3 przedstawiono pomiar odległości z pięciu sonarów unieruchomionego robota, gdy przeszkody ustawione były w odległości 25 cm, a na rysunku 4 dla przeszkód w odległości 50 cm. Wyniki pomiarów wskazują na dobrą powtarzalność uzyskiwanych pomiarów odległości. Pomiar odległości realizowany podczas jazdy robota ze stałą prędkością przedstawiono na rysunku 5. Wykres pokazuje pomiar uzyskany ze środkowego sonaru, gdy stała przeszkoda umieszczona była w odległości 130 cm, a warunkiem zatrzymania jazdy robota była odległość mniejsza niż 20 cm. Pomiar wartości uzyskiwanych z wybranych czujników zbliżeniowych podczerwieni (emisja światła i pomiar po odbiciu) w trakcie zbliżania się do przeszkody przedstawione zostały na rysunku 6. Czujniki zwracają pomiar w postaci liczby 12-bitowej. Największa wartość występuje blisko przeszkody. Te same czujniki mogą być wykorzystane do pomiaru światła docierającego do nich (brak emisji, wykorzystanie tylko odbiorników). Pomiar z czujników zbliżeniowych zależy od rodzaju przeszkody oraz warunków oświetlenia. Przykładowe wyniki dla jednego z czujników dla różnych przeszkód (drewniane klocki, biała kartka i ciemna kartka) dla tych samych warunków oświetlenia przedstawiono na rysunku 7.



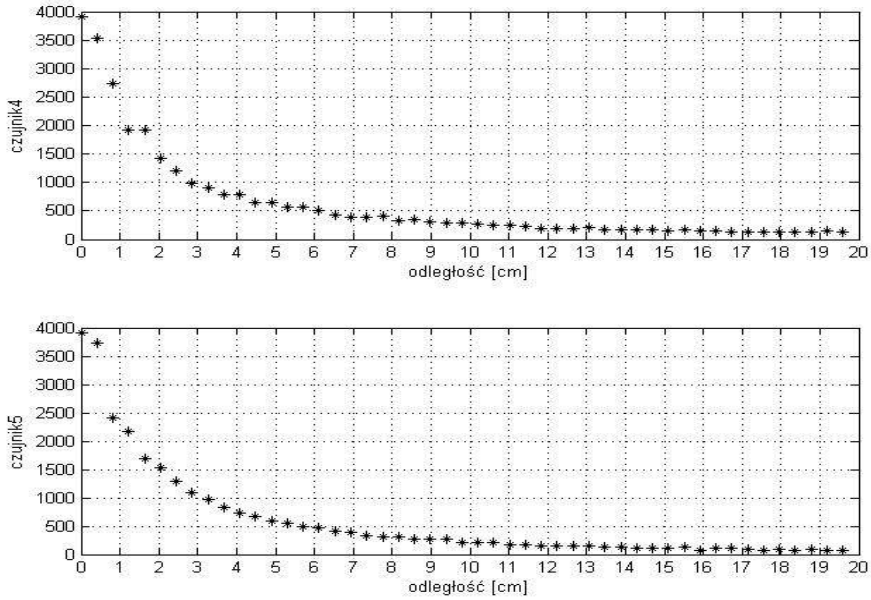
Rys. 3. Wyniki pomiarów poszczególnych sonarów; przeszkody ustawione w odległości 25 cm od robota



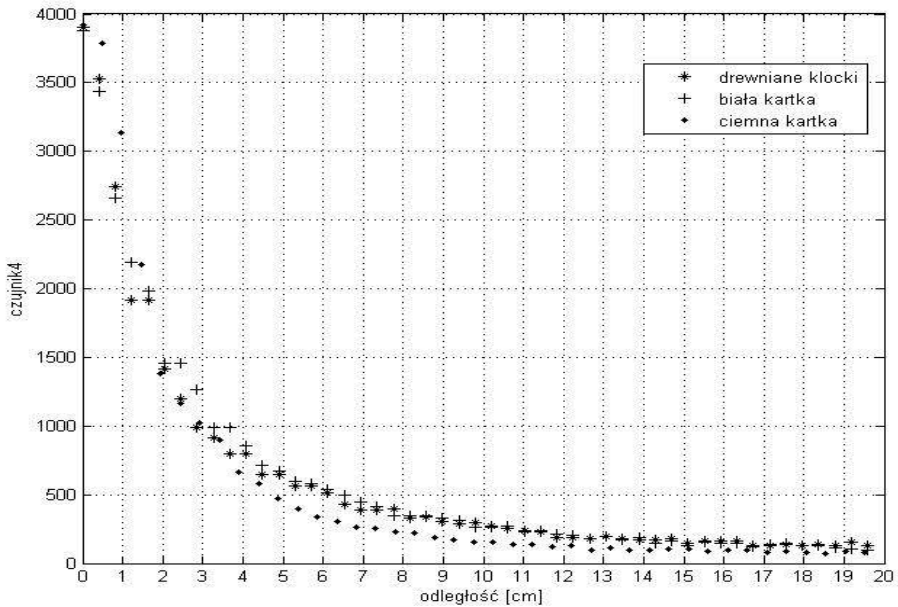
Rys. 4. Wyniki pomiarów poszczególnych sonarów; przeszkody ustawione w odległości 50 cm od robota



Rys. 5. Pomiar odległości ze środkowego sonara podczas jazdy robota ze stałą prędkością



Rys. 6. Pomiary z czujników podczerwieni, gdy robot zbliżył się do przeszkody



Rys. 7. Pomiary z czujnika podczerwieni w zależności od rodzaju przeszkody

3. Podsumowanie i wnioski

Stworzona biblioteka funkcji pozwala, poprzez wykorzystanie bogatych możliwości jakie daje środowisko MATLAB, na łatwość implementacji i testowania algorytmów ruchu robota w nieznanym otoczeniu. Jest to szczególnie cenne z uwagi na dużą popularność środowiska MATLAB/Simulink wśród studentów odbywających zajęcia laboratoryjne. Biblioteka ta podnosi funkcjonalność robota Khepera III i pozwala na prowadzenie zajęć laboratoryjnych z wykorzystaniem tego robota nawet ze studentami niższych lat.

Literatura

- [1] Garbacz M., *Laboratoryjny robot mobilny Khepera II*. Automatyka (półrocznik AGH) t. 9, z. 3, 2005, 393–400.
- [2] Midas Components Ltd. – Air Ultrasonic Ceramic Transducers 400ST/R100.
- [3] Vishay – Reflective Optical Sensor with Transistor Output TCRT5000(L).
- [4] *Khepera III*, User Manual. K-Team S.A, 2007.
- [5] Matlab 7.0 User Guide- The Mathworks Inc., 2007.