

Maciej Garbacz*

Laboratoryjny robot mobilny Khepera II

1. Wprowadzenie

Zbudowane stanowisko ma na celu bezpieczne testowanie algorytmów poszukiwania ścieżki dla robotów mobilnych przy wykorzystaniu miniaturowego robota Khepera II. Sprawdzone w ten sposób algorytmy można później implementować w dużych robotach, pracujących w rzeczywistych warunkach. Istnieje pełna dowolność w umieszczaniu przeszkód w obszarze roboczym. Różne warianty ustawienia przeszkód pozwalają na sprawdzenie zaprojektowanych algorytmów poszukiwania ścieżki optymalnej, z uwzględnieniem omijania przeszkód [2]. Ponadto aplikacja umożliwia operatorowi sprawdzenie działania algorytmów przetwarzających informacje zbierane o otoczeniu poprzez czujniki. Kabel, którym robot połączony jest z komputerem, umieszczony jest na specjalnym wysięgniku. Daje to pełną swobodę ruchu robota oraz minimalizuje możliwość najeżdżania robota na kabel czy przesłaniania przez kabel czujników umieszczonych na robocie. Konieczność prowadzenia łącza przewodowego jest pewnym mankamentem, podważającym cechę autonomiczności takiego robota. Jeżeli jednak sterowanie może odbywać się drogą radiową, to zainstalowane na robocie akumulatory pozwalają na najwyżej jedną godzinę ciągłej pracy. Dlatego w większości przypadków (zajęcia kilku grup następujące po sobie) prowadzenie łącza zasilającego staje się konieczne.

2. Robot Khepera II

2.1. Fizyczna budowa robota

Budowa robota jest modułowa. Pozwala to na ewentualną rozbudowę i łatwe zainstalowanie dodatkowego wyposażenia. Khepera II porusza się na dwóch symetrycznie umieszczonych kołach wyposażonych w oponki. Dla utrzymania stabilności robot podparty jest dodatkowo w dwóch punktach. Każde z kół napędzane jest silniczkiem prądu stałego poprzez przekładnię redukcyjną 25:1. Pomiar położenia kół dokonywany jest poprzez

* Katedra Automatyki, Akademia Górniczo-Hutnicza, Kraków

przetworniki obrotowo-impulsowe. Rozdzielczość użytych enkoderów wynosi 24 impulsy na obrót silnika, co po uwzględnieniu przekładni daje 600 impulsów na obrót koła [6]. Po przeliczeniu na drogę odpowiada to 12 impulsom na 1 milimetr przejechanej przez robot drogi. Wygląd robota przedstawia rysunek 1.



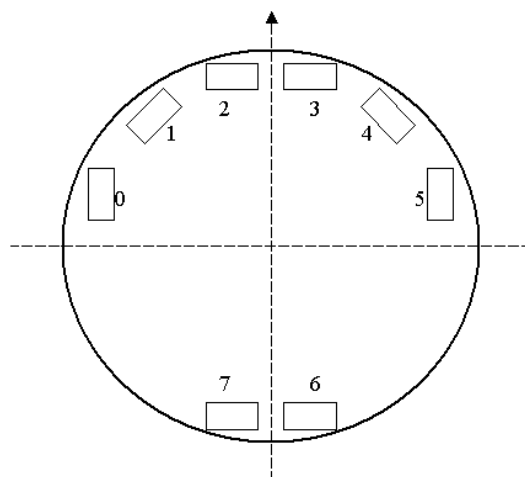
Rys. 1. Robot Khepera II

Robot wyposażony jest w cztery baterie o pojemności 250 mA·h każda. Teoretycznie w pełni naładowane baterie powinny umożliwić w pełni autonomiczną pracę robota przez około godzinę. Czas ten ulega jednak skróceniu wraz ze zużyciem baterii. Tak krótki czas potwierdza, że jednym z najpoważniejszych problemów w robotyce mobilnej jest trudność dostarczenia zasilania autonomicznemu robotowi. Podstawowe parametry robota zebrano w tabeli 1.

Tabela 1
Podstawowe parametry robota Khepera II

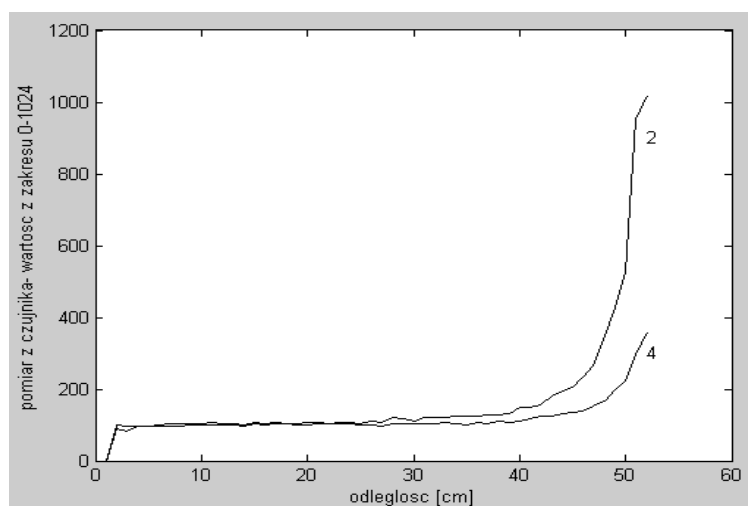
Procesor	Motorola 68331, 25 MHz
RAM	512 KB
Wysokość	30 mm
Średnica	70 mm
Waga	60 g
Ładowność	250 g
Prędkość	0,02÷1 m/s

Do pozyskiwania informacji o otoczeniu robot wyposażony jest standardowo w osiem czujników działających w paśmie podczerwieni. Rozmieszczenie czujników przedstawiono na rysunku 2.



Rys. 2. Rozmieszczenie czujników. 0–7 czujniki

Każdy czujnik składa się z nadajnika i odbiornika światła w paśmie podczerwieni. Czujniki dokonują dwóch rodzajów pomiaru. Pierwszy pomiar (bezwzględny) odnosi się do światła otaczającego. Pomiar taki dokonywany jest co 20 ms i uczestniczy w nim tylko odbiornik. Przy drugim rodzaju pomiaru (pomiar względny) bierze udział także nadajnik. Wysyła on wiązkę światła, a odbiornik mierzy różnicę w stosunku do pomiaru światła pochodzącego tylko z otoczenia. Na tej podstawie można wnioskować o odległości od otaczających przeszkód. Na rysunku 3 przedstawiono wartości z czujników 2 i 4 uzyskane eksperymentalnie przy dojeżdżaniu do prostopadle ustawionej przeszkody (białej kartki).

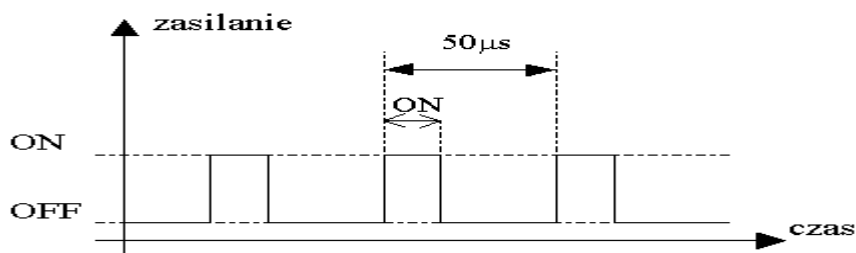


Rys. 3. Wartości z czujników 2 i 4 przy dojeżdżaniu do przeszkody

Okazuje się, że duże znaczenia ma rodzaj materiału, z którego zbudowane są przeszkody. Dla poprawy funkcjonowania czujników producent oferuje specjalny odblaskowy papier. Przeszkody po oklejeniu nim są wykrywane ze znacznie większej odległości, co pozwala na bezpieczną jazdę robota z dużymi prędkościami. Użytkownik dysponując pomiarami z 8 czujników, ma możliwość pozyskania informacji o otoczeniu robota. Pozwala to na dostosowanie aktualnej trajektorii do bieżącej sytuacji w obszarze roboczym. Czujniki odgrywają decydującą rolę w nawigacji robota. W podstawowym wyposażeniu są jedynym źródłem informacji o otoczeniu. Pozwalają na skorygowanie toru jazdy robota w przypadku zmian w obszarze roboczym lub w przypadku natrafienia na przeszkodę.

2.2. Sterowanie

Główny procesor robota Khepera II bezpośrednio steruje zasilaniem silników, dysponując odczytami impulsów z enkoderów zliczających. Sterowanie odbywa się poprzez zmianę okresu wypełnienia PWM (*Pulse width modulation*). Podstawowy okres wynosi $50 \mu\text{s}$. Jest on na tyle duży, że silnik nie reaguje na pojedyncze przełączenia. Istotne znaczenie ma średni czas trwania okresu „ON”. Sterowanie następuje poprzez zmianę stosunek pomiędzy okresem „ON” a okresem „OFF” (rys. 4).



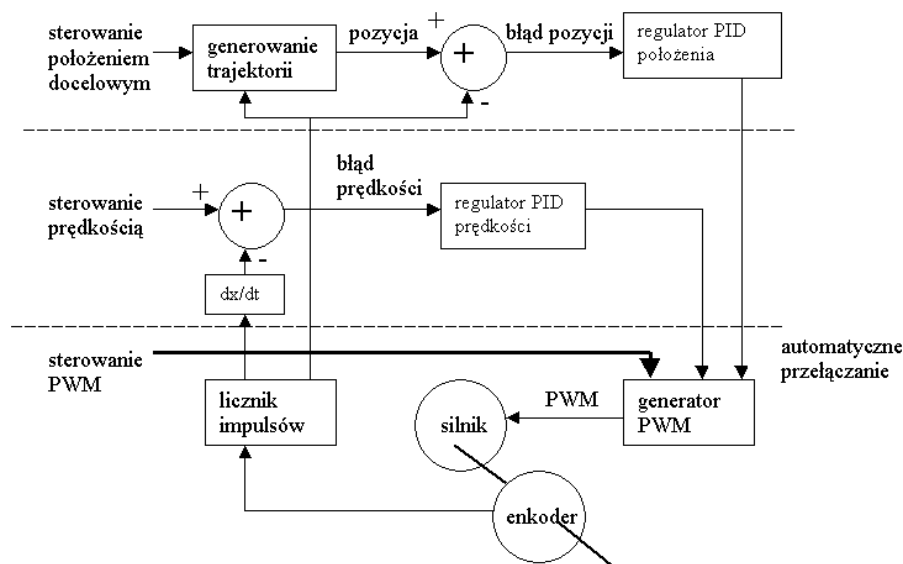
Rys. 4. Idea sterowania PWM

Struktura sterownika przedstawiona jest na rysunku 5. Jest ona trójwarstwowa.

Użytkownik może używać go w dwóch trybach:

- 1) trybie prędkości,
- 2) trybie położenia.

Odpowiedni tryb jest włączany automatycznie w zależności od instrukcji sterującej przesłanej do robota. Poszczególne parametry regulatorów (K_p , K_i , K_d) mogą być ustawiane przez użytkownika dla obydwu trybów. W trybie prędkościowym regulator stara się jak najszybciej osiągnąć zadaną prędkość; nie ma ograniczeń na przyśpieszenia. W trybie położeniowym natomiast użytkownik może zadeklarować przyśpieszenie, z jakim ma narastać prędkość oraz maksymalne prędkości dla obydwu kół. Wykres zależności prędkości od czasu ma kształt trapezu.



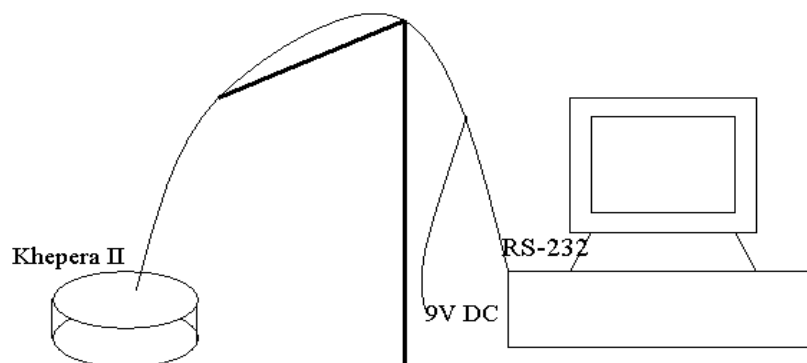
Rys. 5. Schemat sterownika

2.3. Możliwości rozbudowy

Budowa robota umożliwia łatwe umiejscowienie na nim dodatkowego wyposażenia. Producent oferuje szereg akcesoriów ułatwiających pracę z robotem bądź rozszerzających jego możliwości. Jednym z akcesoriów jest chwytak (o niewielkim udźwigu, rzędu 50 g) pozwalający na przenoszenie drobnych przedmiotów z miejsca na miejsce. Istnieje możliwość zainstalowania kamery wideo dostarczającej obraz o rozdzielczości 500×582 piksele (w systemie PAL) lub 510×492 piksele (w systemie NTSC). Kamera taka pozwala na pozyskanie większej ilości informacji o otaczającej robot przestrzeni. Ponadto istnieje możliwość zamontowania modułu do komunikacji radiowej. Moduł taki współpracuje ze stacjonarną bazą. Jedna baza może obsłużyć do 32 robotów. Pozwala to na pełną autonomiczność robota, bez konieczności prowadzenia łącza przewodowego. Ze względu jednak na krótki czas pracy przy czerpaniu energii z akumulatorów, korzyści z komunikacji radiowej są niewielkie, gdyż prowadzenie kabla i tak jest konieczne w celu dostarczenia energii. Istnieje również możliwość ładowania swojej aplikacji do RAM (w postaci pliku wykonywalnego dla procesora Motorola 68000 lub napisanego w języku C i odpowiednio skompilowanego). Wówczas robot może poruszać się bez konieczności komunikowania się z komputerem.

3. Stanowisko

Wygląd stanowiska przedstawia rysunek 6. Specjalny wyciąg z obrotowym ramieniem prowadzi kabel do robota dając mu swobodę ruchu oraz nie przesłaniając czujników. Poprzez specjalny moduł dostarczony przez producenta, zasilanie i sterowanie prowadzone jest wspólną wiązką cienkich przewodów.



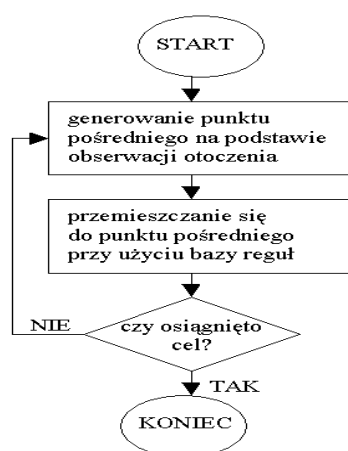
Rys. 6. Schemat poglądowy stanowiska

Obszar roboczy jest ograniczony bandami. Ograniczenie takie podyktowane jest przede wszystkim długością kabla. Użytkownik wykorzystując np. klocki drewniane, może dowolnie zaplanować przestrzeń roboczą, ustawiając przeszkody. Robot ma możliwość pracy w trybie demonstracyjnym, poruszając się według algorytmu Braitenberga. Komunikacja komputera z robotem odbywa się poprzez port szeregowy z wykorzystaniem protokołu RS232. Komputer pełni rolę mastera, a robot slave'a. Komputer nadrzędny oraz robot porozumiewają się poprzez przesyłanie sobie wiadomości w formacie ASCII. Komunikacja może się odbywać nawet poprzez *terminal emulation*, jednak jest to sposób dość uciążliwy. Instrukcja wysyłana do robota musi kończyć się znakiem końca linii. Wszelka komunikacja jest inicjowana przez mastera. Zasadniczo istnieją dwa rodzaje instrukcji. Pierwszy rodzaj dotyczy zmiany konfiguracji robota (np. zmiana nastaw regulatorów, parametrów transmisji). Drugi rodzaj instrukcji służy do wydawania komend do działania dla robota (np. uruchomienie silników, odczyt wartości z czujników). Instrukcja wychodząca składa się z jednej lub dwóch wielkich liter z podaniem ewentualnych parametrów oddzielonych przecinkami. W odpowiedzi zwracana jest ta sama komenda, pisana małymi literami z ewentualnymi parametrami. I tak, np. „D, 500, 500” powoduje ustawienie prędkości odpowiednio lewego i prawego silnika. Jednostką jest impuls/10 ms, co odpowiada 8 mm/s. Jako potwierdzenie do komputera przesyłane jest „d”. Z kolei wysłanie „N” skutkuje żądaniem odczytu pomiarów z czujników. W odpowiedzi dostaje się „n, w₀, ..., w₇”), gdzie w₀, ..., w₇ to wartości poszczególnych pomiarów. Najbardziej rozbudowane jest oprogramowanie do współpracy z pakietem LabView. Ze względu jednak na popularność pakietu Matlab w laboratoriach Katedry Automatyki, do komunikacji z robotem wykorzystuje się właśnie Matlab. Producent dostarcza specjalną bibliotekę z instrukcjami pozwalającymi na pełną komunikację z urządzeniem. Wszechstronność pakietu Matlab pozwala na łatwe implementowanie własnych algorytmów, bez konieczności zagłębiania się w szczegóły komunikacji poprzez port szeregowy. Ponadto gotowe standardowe funkcje łatwo pozwalają na operowanie na wielkościach matematycznych związanych z poruszającym się robotem. Dodatkowo istnieje możliwość graficznego obserwowania zarejestrowanych wielkości.

4. Algorytmy planowania trajektorii

Tak skonstruowane stanowisko pozwala na przetestowanie tworzonych algorytmów planowania trajektorii z uwzględnieniem omijania przeszkód. Jedno z bardziej rozpowszechnionych podejść zakłada podział trajektorii w kilku punktach pośrednich. Bardziej zaawansowane algorytmy sprawdzają się nawet w dynamicznie zmieniającym się środowisku [5].

Schemat blokowy takiego algorytmu przedstawia rysunek 7.



Rys. 7. Schemat blokowy algorytmu poszukiwania ścieżki

Na każdym etapie ruchu robota badane są następujące parametry:

- odległość od najbliższego punktu pośredniego (d),
- kąt pomiędzy kierunkiem jazdy robota a prostą łączącą robota i punkt pośredni (α),
- kąt pomiędzy kierunkiem jazdy robota a prostą łączącą punkt docelowy (β),
- prędkości obydwu kół (v_l, v_r).

Tworzy się specjalną bazę reguł, która na podstawie piątki ($d, \alpha, \beta, v_l, v_r$) generuje sterowanie na obydwie koła robota [4]. Przy tworzeniu bazy reguł można wykorzystać logikę rozmytą. Z powodzeniem stosuje się również algorytmy genetyczne [3]. Funkcję celu konstruuje się w oparciu o wymienione wyżej parametry, dobierając odpowiednie wagi. Również wykorzystanie sieci neuronowych przynosi dobre efekty. Stworzono również algorytmy hybrydowe, łączące elementy logiki rozmytej i sieci neuronowych [5].

5. Podsumowanie

Skonstruowane stanowisko pozwala w bezpieczny sposób sprawdzić w warunkach laboratoryjnych działanie algorytmów wyznaczania trajektorii robota mobilnego. Algorytmy, które pomyślnie przejdą testy, mogą być wykorzystane do zastosowania dla dużych

robotów do pracy w warunkach przemysłowych czy nawet niebezpiecznych [1]. Robotyka mobilna oprócz zastosowań czysto dydaktycznych i naukowych wykorzystywana jest także w przemyśle oraz do zadań inspekcyjnych (praca w kanałach, szybach wentylacyjnych). Specjalne roboty pirotechniczne wykorzystywane są przez wojsko lub straż pożarną do penetrowania terenów niebezpiecznych – tam, gdzie praca człowieka niesie duże ryzyko. Marginalną sprawą może być również wykorzystanie do celów komercyjnych (promocje firm, pokazy na festiwalach).

Literatura

- [1] Chojecki R.: *Miniaturowy robot mobilny VIRTUS*. Pomiary Automatyka Robotyka, nr 9, 2001, 5–7
- [2] Garbacz M.: *Algorytmy wyznaczania trajektorii dla robotów mobilnych*. V Ogólnopolskie Warsztaty Doktoranckie OWD'2003, Istebna-Zaolzie, z. 1, 2003, 57–62
- [3] Kubota N.: *Learning of mobile robots using perception-based genetic algorithm*. Measurement, 29, 2001, 2000, 237–248
- [4] Watabe H. i in.: *Intelligent Motion Generator for Mobile Robot by Automatic Constructed Action Knowledge-Base Using GA*. KES 2003, LNAI 2773, 2003, 271–278
- [5] Xiaowei Ma i in.: *Fuzzy neural network-based real-time self-reaction of mobile robot in unknown environments*. Mechatronics, 11, 2001, 1039–1052
- [6] *Khepera 2-user manual*. 2002. K-Team S.A., Ch. De Vuasset, CP 111, 1028 Preverenges, Switzerland