

Realizacja zadania omijania przeszkód dla trójkołowego robota mobilnego

Opisano robot mobilny Barrel i jego oprogramowanie. Zaprezentowano prosty układ sterowania rozmytego realizujący zadanie omijania przeszkód.

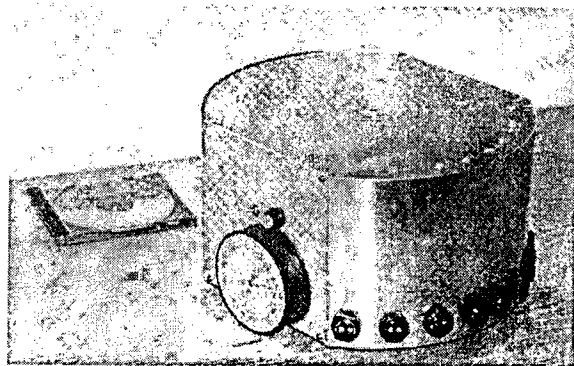
A Realisation of Obstacle Avoidance Behaviour for a Three-Wheeled Mobile Robot

A mobile robot called Barrel robot and its software are described. A simple fuzzy control system is presented. The system performs obstacle avoidance behaviour.

1. ROBOT MOBILNY BARREL

1.1. Charakterystyka ogólna

W ramach prac dotyczących robotów mobilnych i algorytmów ich sterowania prowadzonych w Katedrze Automatyki i Informatyki Politechniki Rzeszowskiej, skonstruowano prototyp robota Barrel. Jest on trójkołowym robotem mobilnym zbudowanym na planie koła o promieniu 30 cm. Po obu stronach robota umieszczono niezależne koła napędowe (rys. 1), z tyłu pojazdu zamocowano trzecie koło - nazywane dalej kołem wleczonym. Robot napędzany jest dwoma silnikami prądu stałego zamocowanymi poprzez przekładnie bezpośrednio na kołach napędowych. Koło wlezione tworzy punkt podparcia oraz pozwala na stwierdzenie zaistnienia poślizgu na kołach napędowych.

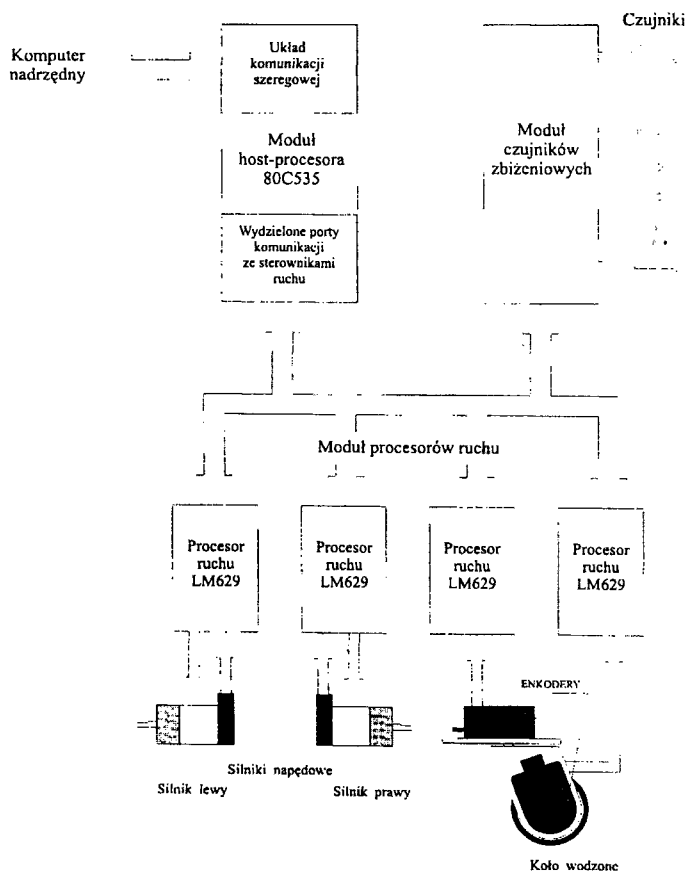


Rys. 1. Robot mobilny Barrel

W protokole komunikacyjnym robot-komputer oprócz rozkazów ściśle odnoszących się do specyficznego dla robota układu sterowania, zaimplementowano rozkazy z protokołu robota Khepera [2] dostosowując je do budowy sprzętowej robota trójkołowego. Pozwala to na szybkie przenoszenie algorytmów sterowania zbudowanych i zaimplementowanych w robocie Khepera. Stworzone jest więc środowisko pozwalające na weryfikowanie algorytmów sterowania w pewnej klasie robotów mobilnych.

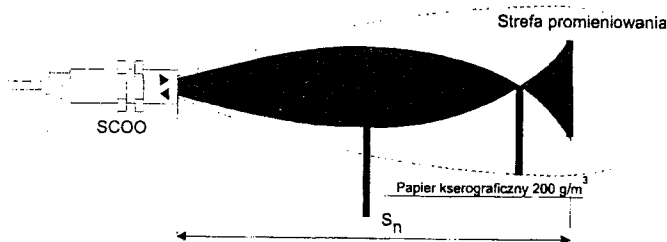
1.2. Układ sterowania

Układ sterowania (rys. 2) ma budowę modułową składającą się z modułu procesora nadrzędnego (80C535), modułu czujników zbliżeniowych i modułu czterech procesorów ruchu - specjalizowanych układów sterowania serwomechanizmami LM629. Dwa z nich spełniają rolę układów pomiarowych dla koła wodzonego. Pozostałe dwa kontrolują serwomechanizmy napędowe. Procesor 80C535 oprócz wypracowywania sterowań przesyłanych do układów LM629 realizuje także odometrię i komunikację z komputerem nadrzędnym.

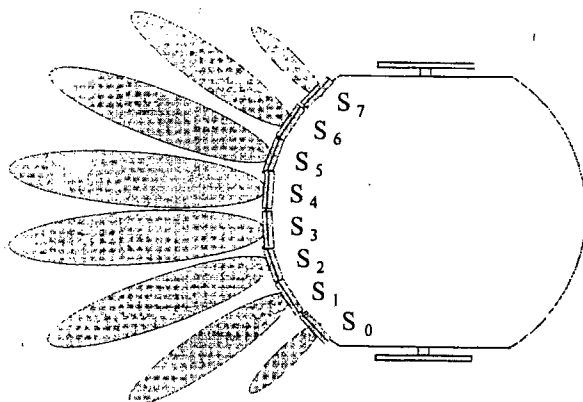


Rys. 2. Schemat układu sterowania

Jako czujniki zastosowano odbiciowe czujniki SCOO-200RP (rys. 3) zwracające wartości binarne. Charakteryzują się one dużym zasięgiem (dla białego matowego materiału do 20 cm) oraz możliwością zmiany wielkości strefy wykrywania przeszkody. Możliwość zmiany odległości granicznej, przy której wykrywana jest przeszkoda wykorzystano do zbudowania widocznej na rys. 4 strefy reakcji na przeszkodę. Konfiguracja taka pozwala na realizację m.in. zadania jazdy "przez tunel" oraz budowanie mapy terenu.



Rys. 3. Działanie odbiciowych czujników optoelektronicznych



Rys. 4. Rozmieszczenie czujników IR

Zastosowanie algorytmu rozmytego spowodowało konieczność rozwiązania problemu zwracania przez czujniki optyczne wartości z przedziału $[0,1]$. Zastosowano metodę "rozmywania w czasie" wartości zmiennych odpowiadających poszczególnym czujnikom. Pojawienie się przeszkody w polu widzenia czujnika powoduje w kolejnych iteracjach programu sterującego dodawanie pewnej stałej DP, zanik sygnału z czujnika (brak przeszkody) powoduje odejmowanie od stanu "analogowego" czujnika stałej DD. Stałe ponadto spełniają warunek $DD > DP$. Przyczynia się to do szybszego "zapominania" o istnieniu przeszkody i w efekcie bardziej płynnej jazdy.

Eksperymenty przeprowadzone na tak symulowanych czujnikach analogowych dowiodły przydatności metody.

3. ZADANIE OMIJANIA PRZESZKÓD

W niniejszym artykule zostanie zaprezentowany układ sterowania realizujący zadanie omijania przeszkód z wykorzystaniem logiki rozmytej. Logika rozmyta umożliwia w tym przypadku zarówno łatwy opis otoczenia robota, jak i intuicyjny sposób sterowania oparty na doświadczeniu projektanta. Podobne podejście zaprezentowano w pracy [4], gdzie przedstawiono realizację algorytmu omijania przeszkód dla miniaturowego robota mobilnego Khepera.

Zadanie omijania przeszkód jest pierwszym, które robot powinien wykonywać, bez względu na to, jakie zadanie globalne realizuje - transportuje obiekty, porusza się do celu, etc. Algorytm sterowania musi przeciwdziałać uszkodzeniu zarówno samego robota, jak i jego otoczenia.

3.1. Lokalizacja przeszkód

Do lokalizacji przeszkód zostaną wykorzystane czujniki IR, których rozmieszczenie i oznaczenia pokazano na rys. 4. Załóżmy, że do realizacji zadania omijania przeszkód chcemy wykorzystać całą dostępną informację. Zatem regulator powinien mieć 8 wejść i 2 wyjścia. Dla każdego wejścia możemy zdefiniować po dwa zbiory rozmyte. Wówczas otrzymujemy 256 (2^8) reguł sterowania. Ustalenie tak dużej liczby reguł przez eksperta jest zbyt trudne w praktyce, a zatem jest konieczna redukcja liczby wejść. W tym celu zastosowano trzy sygnały obliczane następująco: $front = \max(s_3, s_4)$, $left = \max(s_0, s_1, s_2)$ i $right = \max(s_5, s_6, s_7)$. Dzięki temu otrzymano tylko 8 (2^3) reguł sterowania.

Zakres zmian sygnałów z czujników wynika z rozdzielczości zastosowanych zmiennych symulujących analogowy stan czujników i zawiera się w przedziale 0..1023.

3.2. Regulator rozmyty

Do sterowania robotem wykorzystany został tzw. zmodyfikowany regulator rozmyty (*Modified Fuzzy Logic Controller*) [1], który jest rozwinięciem regulatora Takagiego-Sugeno [3]. Regulator posiada trzy wejścia (*front*, *left*, *right*) oraz dwa wyjścia v_l i v_r , będące prędkościami odpowiednio lewego i prawego koła. Pojedyncze wyjście regulatora jest obliczane następująco (przy założeniu, że system reguł jest zupełny i niesprzeczny):

$$v = \frac{\sum_{i=1}^n \beta_i u_i}{\sum_{i=1}^n \beta_i}, \quad (1)$$

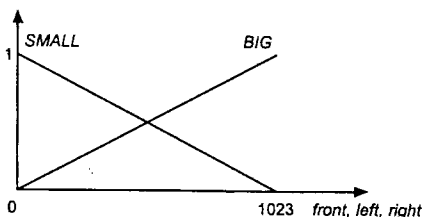
gdzie n jest liczbą reguł ($n=8$), u_i jest wartością sterowania z tablicy reguł (tab.) a współczynniki β_i wyrażają się wzorem:

$$\beta_i = (A(front) \cdot B(left) \cdot C(right))_i. \quad (2)$$

$A(front)$, $B(left)$ i $C(right)$ oznaczają odpowiednio stopnie przynależności wejść *front*, *left*

i *right* do zbiorów rozmytych *A*, *B* i *C* skojarzonych z tymi wejściami (rys. 5). Przyjęto następujące etykiety zbiorów: *SMALL* - mały, *BIG* - duży.

Wzór (1) umożliwia realizację bardzo efektywnego obliczenia sterowania, co jest ważne w przypadku gdy robot porusza się z dużymi prędkościami. Przewagą regulatora zmodyfikowanego (podobnie jak regulatora Takagiego-Sugeno) w porównaniu z tradycyjnym regulatorem rozmytym Mamdaniego, jest uniknięcie czasochłonnego wyostrzania rozmytego zbioru wyjściowego metodą środka ciężkości.



Rys. 5. Zbiory rozmyte dla wejść regulatora

Reguły sterowania zostały dobrane intuicyjnie (tab.). Na przykład reguła o numerze 4 mówi:

(Jeżeli *front* = *SMALL* i *left* = *BIG* i *right* = *BIG*), to ($v_l = 10$, $v_r = 10$).

Oznacza to, że gdy przeszkody znajdują się tylko po lewej i po prawej stronie robota, to robot powinien poruszać się do przodu (v_l i v_r , dodatnie).

TABLICA REGUŁ STEROWANIA

<i>front</i>	<i>left</i>	<i>right</i>	v_l	v_r
<i>SMALL</i>	<i>SMALL</i>	<i>SMALL</i>	10	10
<i>SMALL</i>	<i>SMALL</i>	<i>BIG</i>	-10	10
<i>SMALL</i>	<i>BIG</i>	<i>SMALL</i>	10	-10
<i>SMALL</i>	<i>BIG</i>	<i>BIG</i>	10	10
<i>BIG</i>	<i>SMALL</i>	<i>SMALL</i>	-10	10
<i>BIG</i>	<i>SMALL</i>	<i>BIG</i>	-10	10
<i>BIG</i>	<i>BIG</i>	<i>SMALL</i>	10	-10
<i>BIG</i>	<i>BIG</i>	<i>BIG</i>	-10	10

W przypadku stosowania rozkazów robota Khepera w protokole komunikacyjnym jednostką prędkości w rozkazach sterujących robotem jest impuls/10 ms co odpowiada rzeczywistej prędkości 8 mm/s. W naszym przypadku maksymalna prędkość po linii prostej wynosi 80 mm/s. Została ona dobrana eksperymentalnie. Wartości te są przeliczane na wewnętrzne wartości sterowań sewomechanizmów zgodnie z procedurą nakładki na protokół komunikacyjny.

3.3. Wyniki eksperymentów

Algorytm sterowania został zrealizowany w postaci programu napisanego w języku C. Do sterowania wykorzystano komputer PC486; prędkość transmisji wynosiła 9600 bodów.

Zaprezentowany algorytm nie jest zbyt złożony, lecz takie było zamierzenie projektowe. Przyjęto, że układ sterowania ma być na tyle prosty, aby było możliwe jego ręczne strojenie. Dzięki redukcji wejść regulatora i liczby zbiorów rozmytych uzyskano 8 reguł sterowania. Taką liczbę reguł można łatwo dobrać intuicyjnie. W celu zwiększenia skuteczności układu sterowania należałoby korzystać z większej ilości informacji z czujników, co jednak spowoduje, że układ będzie trudniejszy do strojenia a czas obliczeń wydłuży się.

Praca finansowana przez Komitet Badań Naukowych w ramach grantu nr 8 T11A 028 11.

LITERATURA

- [1] Kluska J.: *A modified fuzzy controller*, Systems Science, 1995, Vol. 21, No. 1, 37-50.
- [2] K-Team: *Khepera user manual*, Lausanne 1995.
- [3] Takagi T., Sugeno M.: *Fuzzy Identification of Systems and Its Applications to Modeling and Control*, IEEE Trans. SMC, 1985, Vol. 15, No. 1, 116-132.
- [4] Wiktorowicz K., Zajdel R.: *Pomiary, Automatyka, Kontrola*, 1997, nr 4, 112-115.