

STEROWNIK ROZMYTY DO NAWIGACJI ROBOTA MOBILNEGO W NIEZNANYM OTOCZENIU

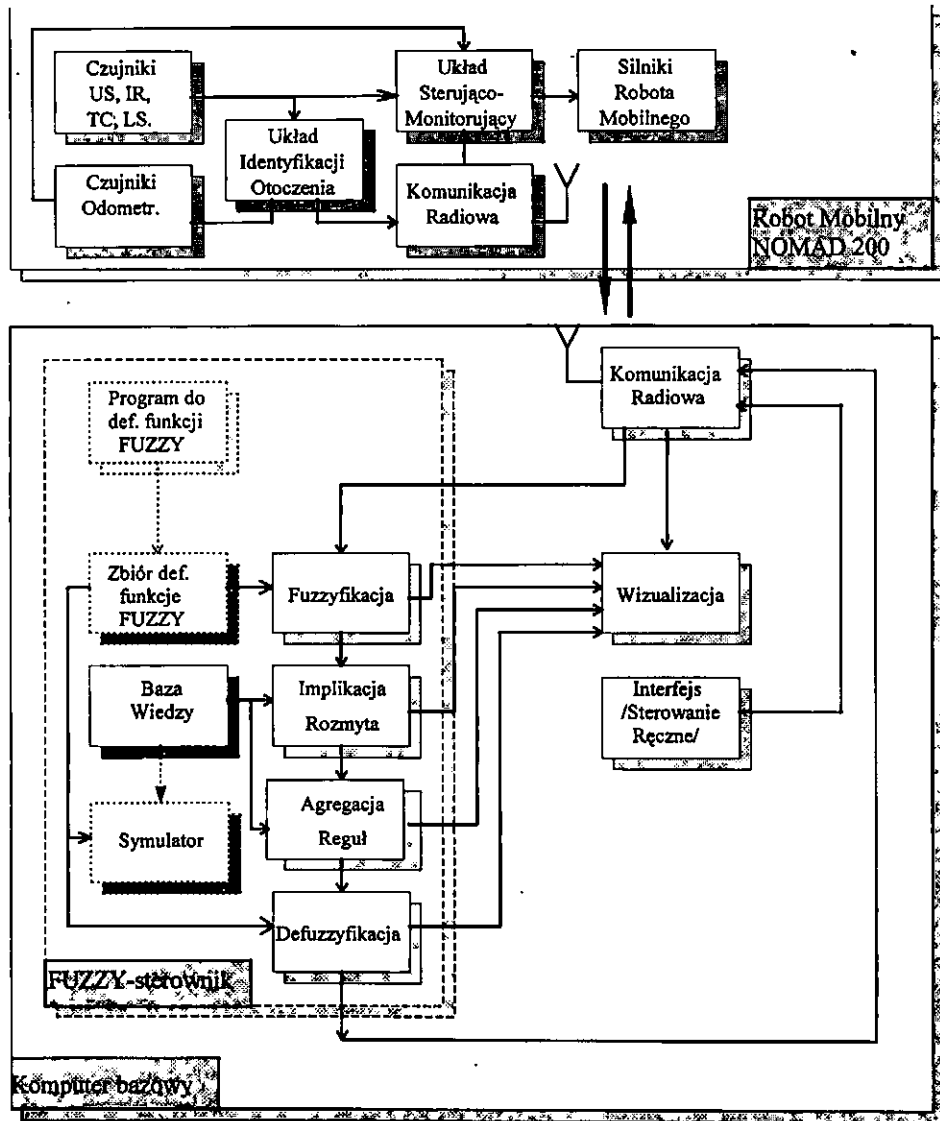
Streszczenie: W pracy prezentowana jest koncepcja i praktyczna realizacja systemu sterującego robotem mobilnym klasy Nomad 200 w nieznanym otoczeniu. System zrealizowany jest w oparciu o sterownik rozmyty, który na podstawie fuzji danych otrzymanych z czujników odległości dokonuje takiego doboru prędkości postępowej i obrotowej robota, aby mógł on w sposób bezkolizyjny dotrzeć do zadanego punktu. Referat przedstawia ogólną budowę całego systemu z dokładnym omówieniem modułu wnioskowania rozmytego oraz identyfikacji otoczenia wraz z wynikami badań dokumentującymi działanie systemu.

Abstract: The concept and practical realization of a system controlling a class Nomad 200 mobile robot in an unknown environment are presented. The system is realized using a fuzzy controller which chooses the translation and rotation velocities for the robot, based on fusion of data obtained from distance sensors, enabling the robot to reach the set point without collision. This paper presents the overall construction of the entire system with a detailed description of the fuzzy logic module and environment identification along with research results documenting the operation of the system.

1. WSTĘP

W ostatnich latach odnotowuje się duże zainteresowanie przemysłu systemami sterującymi wykorzystującymi logikę rozmytą. Wielkie koncerny przemysłowe (głównie w Japonii) stosują sterowniki rozmyte w sprzęcie codziennego użytku (pralki, lodówki, odkurzacze itp.) Sterowniki te pozwalają bowiem na wykorzystanie transformacji strategii sterowania obiektem rzeczywistym na zbiór syntetycznych reguł, przez co łączą precyzję konwencjonalnej matematyki z wiedzą doświadczonego operatora. Zapis reguł w postaci jeżeli-to stanowi powód dla którego są one dla operatora łatwym i przyjaznym, bardzo wysokiego poziomu językiem programowania sterownika. Mogą one być stosowane wszędzie tam, gdzie sterowanie było dotychczas wykonywane przez człowieka czerpiącego informacje z różnego typu czujników. Przykładem takim mogą być prace prowadzone nad pojazdami autonomicznymi [5]. Inną płaszczyzną zastosowań sterowników rozmytych są urządzenia dotychczas sterowane przez algorytmy bazujące na wiedzy o obiekcie sterowania. W 1994 roku Mamdani udowodnił możliwość całkowitego zastąpienia regulatorów PID przez sterowniki rozmyte w których parametry sterowania są łatwe do modyfikacji w celu dostosowania ich do indywidualnych cech obiektu, bez znajomości jego modelu matematycznego. Powyższe względy skłoniły autora do budowy systemu sterującego dla robotów mobilnych. Praca zawiera opis pierwszej wersji systemu, która jest nadal rozbudowywana i doskonalona.

2. BUDOWA SYSTEMU STERUJĄCEGO



Rys. 2.1 Schemat budowy systemu sterującego robotem mobilnym w nieznanym otoczeniu

Rysunek 2.1 przedstawia schemat systemu sterującego robotem mobilnym w nieznanym otoczeniu. System składa się z szeregu modułów z których część stanowią programy podstawowe, biorące udział w pracy systemu *on-line*, natomiast część oprogramowania pełni jedynie funkcje pomocnicze lub narzędziowe (na rysunku 2.1 zaznaczone linią przerywaną). Ze względu na możliwość równoległej pracy układu identyfikacji otoczenia oraz sterownika, ograniczenia systemowe komputera robota, jak również w celu ułatwienia implementacji systemu na dowolnym robocie mobilnym, zadania zostały rozdzielone między komputer pokładowy robota mobilnego Nomad 200 i zewnętrzny komputer bazowy. Komputer bazowy

pracuje w systemie operacyjnym czasu rzeczywistego QNX 4.2, natomiast komputer pokładowy robota mobilnego pracuje w środowisku systemu DOS. Transmisja danych odbywa się przy pomocy modemów radiowych. Uruchomienie programu ROBOT na komputerze pokładowym robota powoduje jego aktywację i przejście do trybu czuwania. Na komputerze bazowym uruchamiany jest program interfejsu, pozwalający użytkownikowi zarówno na ręczne sterowanie, wizualizację procesu sterowania jak i zadawanie współrzędnych celu do którego baza mobilna ma dojechać w sposób autonomiczny.

2.1 Układ identyfikacji otoczenia

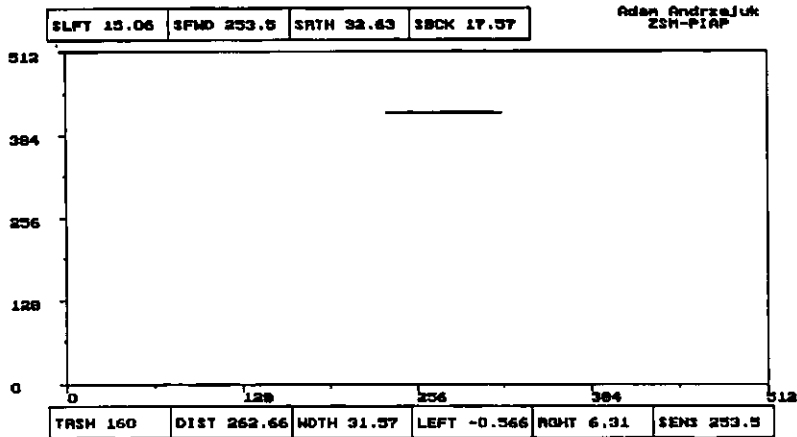
Układ identyfikacji otoczenia dokonuje fuzji odczytów z dalmierzy podczerwieni (IR), ultradźwiękowych (US) oraz dalmierza laserowego (LS), a także w sposób ciągły analizuje wskazania czujników wewnętrznych robota (odometria). Czujniki IR oraz US mają uzupełniający się zakres pomiarowy dzięki czemu monitorowana jest cała przestrzeń w odległości od 0 do 6,5 m wokół robota. Rozdzielczość kątowa czujników wynosi 22,5 stopnia, więc wnioskowanie o szerokości przeszkody jest obciążone dużym błędem. Z tego powodu wykorzystywany jest dalmierz laserowy. Jego zadaniem jest określenie szerokości przeszkody. Autor opracował metodę automatycznej, dynamicznej kalibracji dalmierza laserowego [7], dzięki czemu zminimalizowany został wpływ zakłóceń na dokładność pomiaru wykonywanego przy zastosowaniu tego typu czujnika. W rezultacie cyfrowej obróbki sygnału pochodzącego z kamery, otrzymywana jest przybliżona wartość opisująca szerokość najbliższej przeszkody znajdującej się z przodu robota. Na rysunku 2.2 zmienna WIDTH opisuje szerokość wykrytej przeszkody, natomiast zmienne LEFT i RIGHT, położenie kątowe krańców przeszkody względem osi robota mobilnego. Zmienna Sens opisuje odległość od przeszkody z przodu robota zmierzoną przy użyciu czujnika US lub IR. Na górze rysunku 2.2 znajdują się zmienne opisujące odległości od przeszkód rozmieszczonych wokół robota (z lewej strony, z przodu, z prawej i z tyłu).

W prezentowanym sterowniku zastosowano takie rozwiązanie, że wieża robota mobilnego zawsze jest obrócona w kierunku punktu docelowego, natomiast manewry związane z omijaniem przeszkody lub zmianą kierunku wykonywane są jedynie przez koła robota. W związku z tym powstają dwa obszary, których monitorowanie jest szczególnie ważne z uwagi na bezpieczeństwo i efektywność pracy systemu. Pierwszy obszar znajduje się z przodu robota w rozumieniu aktualnego kierunku przemieszczania robota, natomiast drugi obszar znajduje się z przodu robota w rozumieniu aktualnej konfiguracji wieży robota (rysunek 2.3).

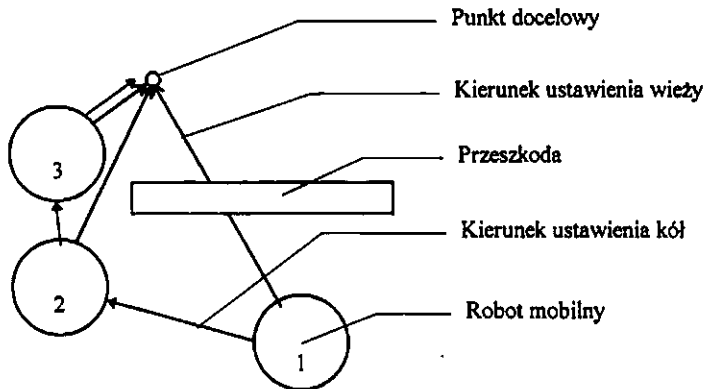
Układ identyfikacji odczytuje również dane z czujników wewnętrznych robota mobilnego. Na bieżąco są więc analizowane prędkości wszystkich stopni swobody robota, jego współrzędne oraz konfiguracja. Na podstawie tych danych obliczany jest błąd kierunku robota, czyli kąt o jaki należy obrócić robota aby wykonywał on ruch dokładnie w kierunku celu oraz błąd położenia robota, czyli odległość między robotem a zadanym punktem.

W rezultacie działania układu identyfikacji otoczenia otrzymywana jest informacja o:

- a) odległości od przeszkody znajdującej się między robotem, a punktem docelowym,
- b) szerokości przeszkody i optymalnym kierunku jej ominięcia,
- c) odległości od przeszkody znajdującej się na drodze poruszającego się robota,
- d) wartości błędu położenia robota,
- e) wartości błędu kierunku robota,
- f) prędkości wszystkich stopni swobody robota,
- g) konfiguracji wszystkich stopni swobody.



Rys. 2.2 Interpretacja graficzna fuzji sensorów po wyodrębnieniu przeszkody najbliższej



Rys. 2.3 Zmiana konfiguracji wieży i kół w kolejnych fazach omijania przeszkody

2.2 Układ monitorujący

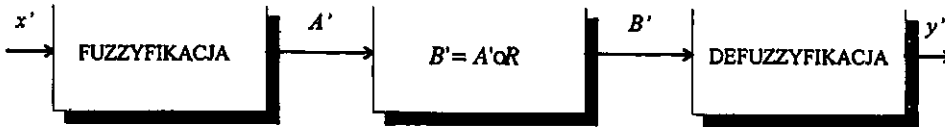
W przypadku gdy sterowanie robota mobilnego odbywa się zdalnie zachodzi konieczność stosowania układu, który w sposób nadrzędny będzie monitorował proces sterowania robota. Robot mobilny ma strukturę wieloprocesorową. Wraz z możliwością wykonywania wielu zadań równocześnie pojawia się niebezpieczeństwo utraty sterowania nad bazą mobilną. Gdy np. wydana jest komenda jazdy robota z określoną prędkością, a następnie połączenie radiowe zostaje zakłócone lub przerwane, wówczas nie ma możliwości zdalnego zatrzymania robota. Aby zapobiec powstawaniu takich sytuacji został opracowany i zaimplementowany na robocie układ monitorujący. Składa się on z bardzo prostej procedury uzależniającej prędkość robota od odległości od przeszkody. Gdy natomiast następuje fizyczne zetknięcie z jakąkolwiek przeszkodą (co jest wykryte przez czujnik dotyku (TS)) robot automatycznie się zatrzymuje. Działanie tego modułu jest nadrzędne nad sterownikiem rozmytym, co skutecznie przeciwdziała powstawaniu kolizji będących wynikiem ewentualnych zakłóceń transmisji radiowej.

2.3 Sterownik rozmyty

Schemat reprezentujący sterownik rozmyty przedstawia rysunek 2.4. Sterowanie rozmyte jest aplikacją wielu reguł wnioskowania. Mając relację A' , reprezentującą stan wejścia sterownika, relację R reprezentującą stan sterownika, wówczas rozmyte wyjście B' otrzymuje się przez złożenie A' i R :

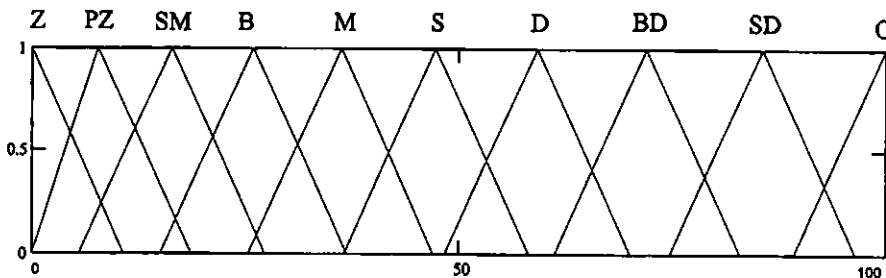
$$B' = A' \circ R \quad (1)$$

Wejścia sterownika są jednak normalnymi wartościami liczbowymi, dlatego niezbędna jest translacja tych wartości na wartości rozmyte (fuzyfikacja). Wartość sterująca otrzymywana na wyjściu sterownika ma natomiast postać rozmytą, więc konieczna jest translacja wartości rozmytej do wartości numerycznej (defuzyfikacja).



Rys. 2.4 Schemat sterownika rozmytego

Dotychczas nie istnieje teoria mówiąca o optymalnej ilości przedziałów rozmytych danego zbioru rozmytego, ani też o optymalnych kształtach funkcji przynależności. Zbiory rozmyte stworzone więc zostały w oparciu o dostępną literaturę [1,6] jak również intuicję i doświadczenie autora programu. Przyjęta została zasada, że minimalna i maksymalna wartość centrów podzbiorów rozmytych odpowiada minimalnej i maksymalnej wartości zmiennej. Zgodnie z tą zasadą przyjęto, że minimalna prędkość translacji jest równa 0, a maksymalna 10 cali/sek (jednostki wynikają z formatu narzuconego przez producenta robota, a przeliczanie ich na standard europejski powoduje stratę czasu obliczeniowego). Maksymalna prędkość obrotowa jest równa 20 stopni/sek, a minimalna -20 stopni/sek. Odległość minimalna jest równa 0 cm, a maksymalna 650, itd. Zbiory rozmyte opisane są metodą funkcyjną z wykorzystaniem funkcji trójkątnej. Liczba przedziałów wynika z przeprowadzonych doświadczeń i jest na tyle duża aby proces sterowania spełniał postawione mu zadania, ale na tyle mała aby nie powodować zbędnej komplikacji zbiorów rozmytych co zasadniczo wpływa na prędkość obliczeń. Aby umożliwić łatwość zmian w definicji zbiorów rozmytych napisany został program, który na podstawie danych wejściowych zapisanych w pliku konfiguracyjnym, dokonuje automatycznej generacji zbioru rozmytego, przypisuje etykiety lingwistyczne każdemu przedziałowi rozmytemu, zapisuje go na dysku w postaci pliku tekstowego oraz wyświetla na ekranie jego graficzną interpretację (rysunki 2.5 i 2.6).

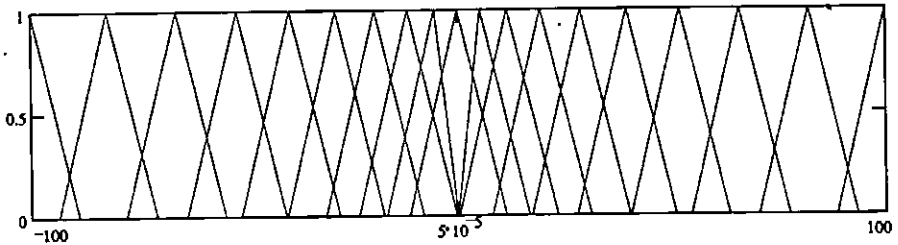


Rys. 2.5 Funkcje przynależności zbioru „prędkość translacji”

Zgodnie z ww. zasadami, stworzone zostały trzy podstawowe zbiory rozmyte opisujące wielkości wejściowe do sterownika:

- 1) Odległość od przeszkody.

- 2) Odległość od celu (błąd pozycji).
- 3) Błąd kierunku.



Rys. 2.6 Zbiór rozmyty „prędkość rotacji”

W oparciu o powyższe zbiory rozmyte stworzone zostały zbiory opisujące wartości wyjściowe ze sterownika:

- 4) Prędkość rotacji w zależności od pożądanego kąta obrotu (dla kątów dużych prędkość może być duża, lecz musi maleć wraz z maleniem kąta).
- 5) Kąt obrotu w zależności od odległości od przeszkody (robot ma średnicę ok. 60 cm, w związku z tym minimalny kąt o jaki należy obrócić koła robota aby nie zawadził on przeszkody zdeterminowany jest jego odległością od przeszkody).
- 6) Prędkość translacji (musi być tak dobrana, aby robot był w stanie zahamować przed przeszkodą lub przed punktem docelowym).

Wnioskowanie rozmyte realizowane jest w sterowniku na podstawie reguł *if-then*. Budowa bazy wiedzy jest procesem wymagającym dobrej znajomości obiektu sterowania, natomiast nie jest wymagana znajomość jego modelu matematycznego.

W oparciu o zasady [3] ciągłości bazy reguł, spójności i kompletności, stworzone zostały trzy zbiory tekstowe zawierające reguły wnioskowania rozmytego:

- 1) *If błąd pozycji is x and odległość od przeszkody is y then prędkość translacji is z*
- 2) *If błąd kierunku is a then prędkość rotacji is b*
- 3) *If odległość od przeszkody is e then prędkość rotacji is f*

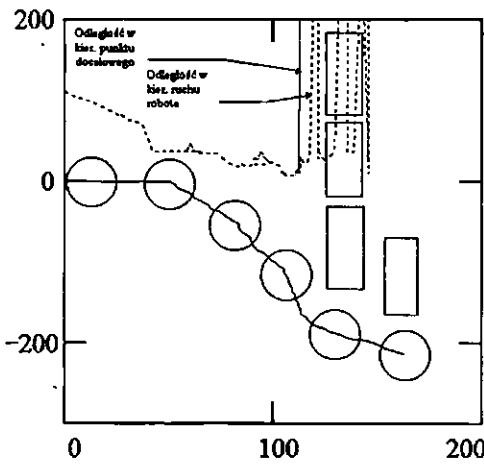
Proces wnioskowania rozmytego składa się z pięciu etapów [2,3,4]: fuzyfikacji, określenia stopnia oddziaływania reguły (zapłonu reguły), implikacji rozmytej (relacji rozmytej dla każdej reguły), agregacji reguł i defuzyfikacji.

W omawianym sterowniku stopień zapłonu każdej reguły określany jest jako minimum stopni przynależności zmiennych lingwistycznych odpowiadających wszystkim zmiennym wejściowym. Nowa postać funkcji przynależności zbioru rozmytego zmiennej lingwistycznej wyjścia otrzymywana jest metodą Mamdaniego. Agregacja reguł realizowana jest przez przeprowadzenie sumy logicznej wszystkich otrzymanych funkcji przynależności zmiennej wyjściowej, natomiast defuzyfikacja realizowana jest w metodą środka masy obszaru.

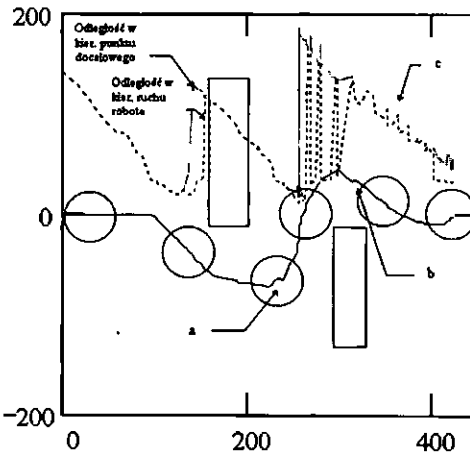
3. EKSPERYMENTY Z WYKORZYSTANIEM ROBOTA MOBILNEGO

Pierwszym omawianym testem, jest ominięcie szerokiej przeszkody znajdującej się na drodze robota. Tor ruchu robota, oraz odległości od znajdujących się przed nim przeszkód (zarówno w kierunku ruchu robota jak i kierunku ustawienia wieży) przedstawione są na rysunku 3.1. Robot od momentu wykrycia przeszkody jedzie prosto w kierunku punktu docelowego. Z prawilą wykrycia przeszkody, układ identyfikacji otoczenia sugeruje ominięcie przeszkody z prawej strony mimo, że położenie robota (znajduje się on dokładnie w osi przeszkody) nie wskazuje na to w sposób jednoznaczny. Sterownik rozmyty wraz ze zmniejszaniem się odległości od przeszkody zwiększa kąt skrętu kół robota, dzięki czemu przeszkoda zostaje

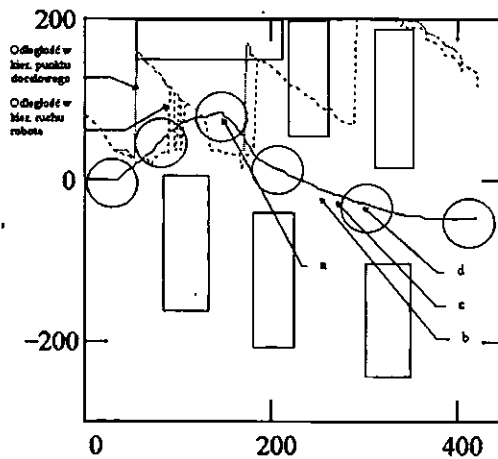
ominięta w bezpieczny sposób. W celu utrudnienia zadania, za pierwszą przeszkodą została umieszczona kolejna, niewidoczna dla czujników robota do chwili ominięcia pierwszej przeszkody. Mimo tego jest ona również prawidłowo ominięta.



Rys. 3.1 Omijanie szerokiej przeszkody



Rys. 3.2 Omijanie dwóch przeszkód



Rys. 3.3 Jazda w korytarzu utworzonym z przeszkód

Kolejny test, którego graficzna interpretacja przedstawiona jest na rysunku 3.2 polega na dojechaniu robota mobilnego do punktu docelowego oddalonego o 4,2 m. Na drodze robota umieszczone są dwie przeszkody ustawione w ten sposób, że możliwa jest tylko jedna bezkolizyjna ścieżka ruchu. Ominięcie pierwszej przeszkody musi odbyć się z jej prawej strony, natomiast druga przeszkoda musi być ominięta z lewej strony (ze względu na odległość od ścian laboratorium). Ponieważ odległość między przeszkodami wynosiła zaledwie 1 m zachodziła obawa, że sterownik robota nie będzie w stanie w sposób bezkolizyjny dotrzeć do celu. Jak pokazano na rysunku 3.2, robot mobilny pierwszą przeszkodę ominął bez wahania (tzn. na skutek wnioskowania rozmytego została podjęta jednoznaczna decyzja odnośnie wykonywanego manewru), natomiast przy drugiej przeszkodzie, w miarę zdobywania informacji o rozmieszczeniu przeszkód, zmieniało się

sterowanie powodując gwałtowne zmiany kierunku ruchu robota (rysunek 3.2 punkt a - ostry skręt w lewą stronę, następnie w prawą stronę i ponownie w lewą stronę). Po dokonaniu kompletnej identyfikacji otoczenia system sterujący bez problemów przeprowadził robota między przeszkodami. W punkcie b rysunku 3.2, pokazana jest korekcja kierunku ruchu robota zapobiegająca zahaczeniu przeszkody. Istotny jest fakt, że w trakcie testu minimalna odległość między robotem, a najbliższą przeszkodą nie była mniejsza niż 5 cm. Zakłócenia wskazań czujników odległości (rysunek 3-2 c) spowodowane są tym, że robot zbliżał się do ściany na której znajdują się grzejniki o nieregularnym kształcie (żeberka). Jest to powodem oscylacji wskazań o +/- 5 cm.

Kolejne doświadczenie polega na dojechaniu robota do punktu docelowego ścieżką wyznaczoną przez przeszkody. Jest to zadanie podobne do jazdy w korytarzu z tym jednak utrudnieniem, że droga wyznaczana przez przeszkody nie ma regularnego ani ciągłego charakteru. Jak widać na rysunku 3.3 robot wykonał zadanie w sposób pewny, pokonując całą odległość środkiem wyznaczonej ścieżki i jedynie w czterech miejscach (rysunek 3.3 punkty a,b,c,d) dokonywał drobnej korekcji kierunku ruchu.

4. PODSUMOWANIE

Efektem realizacji projektu jest sterownik rozmyty zaimplementowany na robocie mobilnym Nomad 200 realizujący bezkolizyjną jazdę w nieznanym otoczeniu. Przeprowadzone doświadczenia pozwalają stwierdzić, że system działa poprawnie i z dużą dozą niezawodności, a układ monitorujący w sposób pewny zabezpiecza bazę mobilną przed kolizją. System sterujący ma zdolności predykcji, dzięki czemu możliwe jest sterowanie robotą w nieznanym i dynamicznie zmieniającym się otoczeniu. Ograniczeniem jest tu jednak mała prędkość transmisji danych w związku z powstawaniem błędów w sieci radiowej, co w przypadku szybkozmiennego stanu otoczenia powoduje dezaktualizację danych sterujących. Mała prędkość transmisji danych spowodowała również brak możliwości jednoczesnego sterowania prędkością obrotową i postępową robota, więc w chwili zmiany kierunku ruchu robot musi się zatrzymać. Modyfikacja systemu polegająca na likwidacji tej niedogodności będzie kolejnym etapem rozwoju sterownika. Dalsze badania autora będą miały na celu wykorzystanie adaptacyjnych i samoorganizujących się zbiorów rozmytych do budowy zaawansowanego, inteligentnego sterownika rozmytego.

5. LITERATURA

- [1] Yager R.R, Filev,D.P: *Essentials of fuzzy modeling and contro*, Cloth ISBN:0-471-01761, 1994, USA.
- [2] Jager R: *Fuzzy Logic in Control*, Thesis Technische Universiteit Delft. ISBN 90-9008318-9, 1995, Nederland.
- [3] Driankov D, Hellendoorn H, Reinfrank M: *An Introduction to Fuzzy Control*, ISBN 3-540-60691-2, ed. Springer, 1996, USA.
- [4] Lista dyskusyjna fuzzy-mail@vexpert.dbai.tuwien.at
- [5] Masłowski A, Andrzejuk A, Czerniewska-Majewska A, Szumilas L, Szynkarczyk P: *Hybrid Simulation System for Analysis of Autonomous Vehicles*, Preprints of the 13th Triennial World Congress of IFAC, San Francisco, USA, 1996, vol Q, pp. 471-476.
- [6] Zadeh L.A: *Fuzzy sets*, Information and Control 8, pp 338-353.
- [7] Andrzejuk A: *Automatyczna kalibracja dalmierza laserowego Sensus 500 oraz fuzja odczytów z dalmierzy podczerwieni, ultradźwiękowych i dalmierza laserowego*, Opracowanie naukowo-badawcze PIAP, Nr 7325.