

Dr inż. Barbara Siemiątkowska
Instytut Podstawowych Problemów Techniki PAN
Mgr inż. Rafał Chojecki
Prof. dr hab. inż. Mariusz Olszewski
Instytut Automatyki i Robotyki
Politechniki Warszawskiej

ZASTOSOWANIE WIELOWARSTWOWYCH SIECI KOMÓRKOWYCH DO PLANOWANIA TRASY DLA ROBOTA MOBILNEGO

W artykule przedstawione zostanie zastosowanie wielowarstwowej sieci komórkowej do planowania trasy dla robota mobilnego. W tej metodzie trasa generowana jest w postaci ciągu trójek (x,y,θ) określających punkty, przez które robot powinien przejechać i orientację w zadanym punkcie. Dla każdego punktu generowana jest optymalna prędkość liniowa i kątowa.

Abstract

This paper presents an application of Cellular Neural Network for path planning for mobile robot. The path is generated as a sequence of triples: (x,y,θ) , where (x,y) is the position of the robot and θ is the orientation of the vehicle. The method generates also optimal values of linear and angular velocities of the robot in each point.

1. WSTĘP

Umiejętność planowania bezkolizyjnej trasy jest jednym z podstawowych zadań każdego systemu nawigacyjnego. Zagadnienie to jest studiowane w robotyce od wielu lat. Do najczęściej stosowanych metod możemy zaliczyć: planowanie grafowe [11], metody gradientowe [1, 3, 8, 9, 10] oraz metody rastrowe [8, 10, 11]. W praktycznych zastosowaniach robotów mobilnych pojazd, oprócz planowania i realizowania trasy, powinien wykonać także szereg innych zadań, np. inspekcyjnych. Warto więc stworzyć system planowania trasy, który mógłby być realizowany sprzętowo i nie obciążał systemu komputerowego robota. Idealnym narzędziem wydają się być sieci komórkowe (CNN¹), które mogą być realizowane przy pomocy układów sprzętowych. Sieci te są opisywane w literaturze [5]. Przedstawiona w tym artykule metoda jest modyfikacją zaimplementowanej przy pomocy CNN metody dyfuzyjnej, która została opisana w pracy [11]. Schemat działania opisanego w [11] systemu jest następujący:

Otoczenie robota jest dzielone na podobszary - kwadraty. Każdej z komórek przypisywana jest jedna z etykiet: *wolna*, *zajęta*. Otoczenie jest reprezentowane w postaci neuronowej sieci komórkowej, w której komórki odpowiadają pewnym fragmentom (kwadratam) otoczenia. Dwie komórki sieci są połączone, jeśli odpowiadające im podobszary przylegają do siebie. Wagi połączeń między komórkami opowiadają odległości euklidesowej między opowiadającymi im podobszarami. Proces generowania trasy składa

¹ CNN- Cellular Neural Network

się z dwóch etapów: tworzenia mapy dyfuzyjnej oraz generowania na jej podstawie trasy.

Pierwszy krok składa się z następujących elementów:

a) Inicjalizacja sieci -

ustalane są wartości wag połączeń między poszczególnymi komórkami sieci:

$$a_{ij}^{kl} = \text{dist}(c_{ij}, c_{kl}) \quad (1)$$

gdzie: c_{ij} odpowiednia komórka mapy, a_{ij}^{kl} waga łącząca komórkę c_{ij} z c_{kl} .

b) Ustalenie wartości sygnałów zewnętrznych

$$u_{ij} = \begin{cases} 0 & , \text{ gdy komórka jest fragmentem przeszkody,} \\ 1 & , \text{ w przeciwnym przypadku.} \end{cases} \quad (2)$$

c) Ustalenie początkowej wartości komórek x_{ij}

$$x_{ij}(0) = \begin{cases} T \gg 1 & , \text{ gdy } c_{ij} \text{ reprezentuje cel,} \\ 0 & , \text{ w przeciwnym przypadku.} \end{cases} \quad (3)$$

gdzie: T jest początkową wartością komórki reprezentującej cel.

d) Proces dyfuzji -

stan komórki w chwili $t+1$ obliczany jest zgodnie z zależnością

$$x_{ij}(t+1) = u_{ij}(t) \cdot \max(x_{ij}(t), \max_{kl \in N_s} (x_{kl}(t) - a_{kl}^{ij})) \quad (4)$$

Proces opisany równaniem (4) jest kontynuowany, aż do momentu osiągnięcia przez sieć stanu równowagi, tzn.

$$\forall ij \quad x_{ij}(t+1) = x_{ij}(t) \quad (5)$$

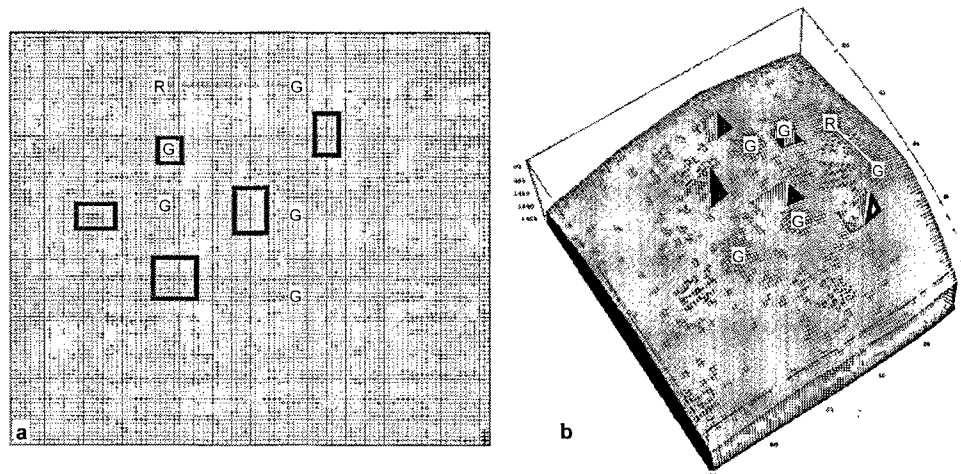
2. MODYFIKACJE ALGORYTMU DYFUZYJNEGO

Opisany we wstępie algorytm ma następujące cechy: dzięki zastosowaniu CNN możliwa jest implementacja wielorównoległa planowania trasy. Postać mapy dyfuzyjnej nie zależy od położenia robota, a jedynie od położenia celu i konfiguracji przeszkód. Czas generowania trasy jest porównywalny z metodami behawioralnymi. W metodzie nie występuje problem minimów lokalnych. Istnieją realizacje sprzętowe opisywanej sieci. Sytuacja, w której cel lub robot jest otoczony przez przeszkody jest łatwo rozpoznawalna - stan komórki zajętej przez robota jest w takiej sytuacji zawsze równy zero. Możliwe są następujące modyfikacje metody:

2.1. Generowanie trasy do najbliższego celu

Aby wygenerować trasę do najbliższego celu modyfikujemy równanie (3). Fala dyfuzyjna rozchodzi się nie od jednej komórki, ale od wszystkich komórek reprezentujących cel. Rys. 1a przedstawia mapę otoczenia, położenie robota (R) i punktów docelowych

(G). Rys. 1b przedstawia postać mapy dyfuzyjnej, na osi Z zaznaczono stopień pobudzenia poszczególnych komórek. Wygenerowaną trasę zaznaczono białą linią.



Rys. 1. Postać mapy dyfuzyjnej w przypadku wielu punktów docelowych: a) mapa otoczenia, b) wartości pobudzeń komórek – mapa dyfuzyjna

2.2. Planowanie trasy według kryterium przydatności

W wielu zagadnieniach praktycznych istnieje konieczność wyboru komórek według kryterium dostępności, np. chcemy wybrać trasę o najlepszej jakości podłoża, trasę o najmniejszym nasłonecznieniu lub chcemy, aby ścieżka była jak najbardziej bezpieczna. W tym celu dla każdej komórki mapy rastrowej definiujemy funkcję kosztu, wartość kosztu będzie wartością sygnału wejściowego u_{ij} . Wartość tej funkcji określa jakość każdego z podobszarów. Dla obszarów zajętych przez przeszkody $u_{ij} = 0$, dla pozostałych komórek $u_{ij} \in (0,1]$. Proces dyfuzji przebiega analogicznie jak w poprzednio opisanym przypadku.

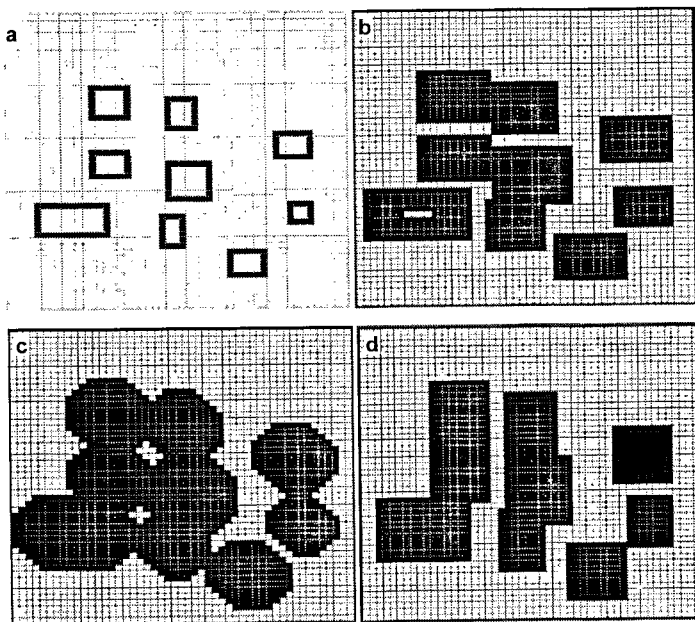
2.3. Planowanie trasy o określonej szerokości

Podstawową opisywaną w literaturze wadą metody dyfuzyjnej jest to, że zaplanowana trasa przebiega zbyt blisko przeszkód. Jeśli jednak wartości sygnałów wejściowych u_{ij} będą reprezentowały siłę odpychania robota przez przeszkody (analogicznie jak w przypadku metody potencjałowej), to zostanie wygenerowana trasa daleka od przeszkód, a jednocześnie unikamy problemów występowania minimów lokalnych.

3. WIELOWARSTWOWA SIĘĆ KOMÓRKOWA

Podstawową wadą opisywanej metody jest to, że nie uwzględnia kształtu robota, tzn. odległość od przeszkody określana jest od środka robota, niezależnie od jego kształtu. W takim przypadku zaplanowana ścieżka nie jest gładka, np. w wielu sytuacjach wymagana jest zmiana orientacji pojazdu w miejscu o 90° , co nie zawsze może być niewykonalne. Aby rozwiązać opisane powyżej problemy zastosowana została wielowar-

stwowa sieć komórkowa. Każda z warstw sieci odpowiada mapie dyfuzyjnej dla danej orientacji robota. Wartości sygnałów wejściowych u_{ij} przyjmują wartości z przedziału $[0,1]$, wartości te są wprowadzane do sieci w postaci sygnałów wejściowych u_{ij} i są generowane na podstawie rastrowej mapy przeszkód oraz orientacji robota. Sposób obliczenia wartości sygnałów u_{ij} zaczerpnięto z metody histogramów kierunkowych [2, 3], która jest metodą odruchowego planowania trasy. Komórki mapy rastrowej oznaczone są symbolem m_{ij} i określają stopień potwierdzenia hipotezy, że dana klatka jest zajęta przez przeszkody. Każda komórka sieci otrzymuje $2r \times 2r$ (r - wymiar sąsiedztwa) sygnałów wejściowych o wartościach m_{kl} , gdzie $kl \in N_r^{ij}$ (N_r^{ij} - r -sąsiedztwo komórki ij). Wagi połączeń między komórkami sieci kodują kształt i orientację robota. Wymiary r -sąsiedztwa określają rozmiar okna aktywnego.



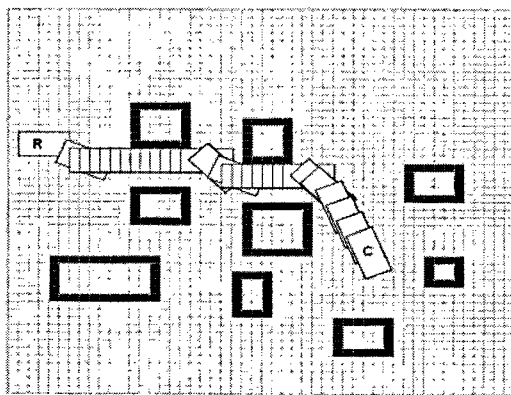
Rys. 3. Wartości sygnałów u_{ij} dla różnych orientacji robota: a) mapa otoczenia, b) $\theta = 0^\circ$, c) $\theta = 45^\circ$, d) $\theta = 90^\circ$

Wartość sygnału u_{ij} obliczana jest zgodnie z zależnością

$$u_{ij}(t) = f\left(\sum_{kl \in N_r^{ij}} \epsilon^{kl} \cdot m_{kl}\right) \quad (6)$$

$$f(x) = \begin{cases} x & x < \Omega \\ \Omega & x \geq \Omega \end{cases} \quad (7)$$

Parametr Ω jest taką wartością funkcji kosztu, przy której uznajemy, że dane położenie jest niedopuszczalne. Wagi połączeń dla komórek, które dla danego położenia i orientacji są zajęte przez robota, są dużo większe niż pozostałe. Jeśli więc przy danym położeniu robota następuje kolizja, to wartość u_{ij} jest bardzo duża. Rys. 3b - 3d przedstawiają graficznie wartości sygnałów wejściowych dla następujących orientacji robota: $\theta = 0^\circ$, $\theta = 45^\circ$, $\theta = 90^\circ$. Pojazd ma wymiary 5 x 3 klatki. Mapę przeszkód przedstawia rys. 3a. Aby rysunek był bardziej czytelny, wartości u_{ij} są progowane. Kolorem czarnym zaznaczono komórki, w których robot nie może mieć zadanej orientacji (nastąpi kolizja z przeszkodami), kolorem białym - pozostałe.



Rys. 4. Zaplanowana trasa przy pomocy sieci komórkowej

Po obliczeniu wartości sygnałów wejściowych sieci przeprowadzany jest proces dyfuzji. Ustalane są połączenia wewnątrz każdej z warstw i między warstwami. Komórki x_{ij}^θ i x_{kl}^φ są ze sobą połączone, jeśli robot może przejechać z x_{ij}^θ i x_{kl}^φ zmieniając orientację z θ na φ oraz $|\theta - \varphi| < \Delta\varphi$, gdzie $\Delta\varphi$ jest ograniczeniem wynikającym z promienia skrętu robota. Inicjalizacja komórek sieci opisana jest zależnością.

$$x_{ij}^\theta = \begin{cases} G & , \text{gdy } c_{ij}^\theta \text{ reprezentuje cel} \\ 0 & , \text{w przeciwnym przypadku.} \end{cases} \quad (9)$$

Symbolami θ , φ oznaczono numery warstw. Indeksy ij , kl określają współrzędne komórek w danej warstwie.

Jeśli nie zakładamy końcowej orientacji robota, to wartość G będzie przypisywana komórkom odpowiadającym położeniu celu we wszystkich warstwach. Proces dyfuzji przeprowadzany jest zgodnie ze zależnością

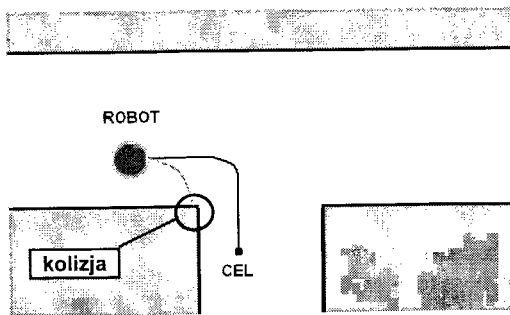
$$x_{ij}^{\theta} = \max(x_{ij}^{\theta}, \max_{kl \in N_{ij}^{\theta}} (x_{kl}^{\alpha} - a_{kl\alpha}^{ij\theta} - u_{kl}^{\alpha})) \quad (10)$$

gdzie: x_{kl}^{α} - wartość komórki kl w warstwie α , x_{kl}^{θ} wartość komórki ij w warstwie θ , $a_{kl\alpha}^{ij\theta}$ - waga połączenia między komórką kl warstwy θ i ij warstwy α .

Jeśli nie uwzględniamy ograniczeń związanych z prędkością kątową i liniową robota, to generowanie bezkolizyjnej trasy przebiega podobnie, jak w przypadku poprzednio opisywanych metod. Jeśli robot znajduje się w komórce c_{ij}^{θ} , to poszukujemy wartości kl i α dla której wartość x_{kl}^{α} jest maksymalna. Przyjmujemy, że c_{kl}^{α} należy od r -sąsiedztwa c_{ij}^{θ} . Parametr α określa nową orientację robota. Wszystkie operacje opisane w algorytmie mogą być wykonane w sposób wielorównoległy. Na rys. 4 przedstawiono postać trasy zaplanowanej dla robota o kształcie prostokąta o wymiarach 5 x 3 klatek.

4. DOBÓR OPTYMALNYCH PARAMETRÓW

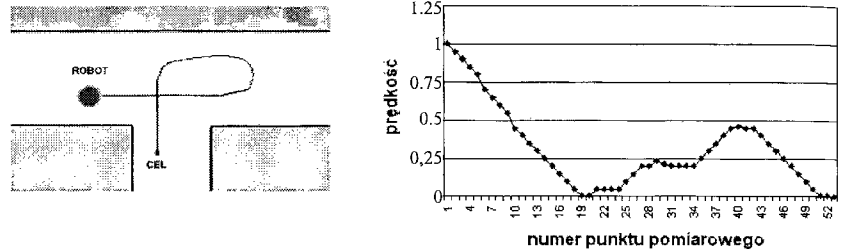
Większość metod planowania trasy nie uwzględnia parametrów robota, takich jak prędkość oraz przyspieszenie kątowe i liniowe. Parametry te są często istotne już na etapie planowania trasy. Rys. 5 przedstawia taką sytuację. Robot jedzie z dużą prędkością i otrzymuje polecenie dojechania do zaznaczonego celu. Ciągną czarną linią zaznaczono zaplanowaną trasę. Ze względu na ograniczone przyspieszenie kątowe i liniowe trasa nie może być zrealizowana i robot będzie przemieszczał się po trasie' zaznaczonej szarą linią. W efekcie nastąpi kolizja. W pracy [6] przedstawiono metodę generowania wartości prędkości liniowej i kątowej robota na podstawie rastrowej mapy otoczenia oraz ograniczeń przyspieszenia pojazdu. Metodę to można bezpośrednio zastosować w opisywanym powyżej algorytmie w następujący sposób: dla wszystkich komórek mapy wyznaczany jest zbiór parametrów $(v_{ij}^{\theta}, \omega_{ij}^{\theta})$, które są dopuszczalne dla danego położenia i orientacji pojazdu, v_{ij}^{θ} - jest prędkością liniową, ω_{ij}^{θ} - kątową.



Rys. 5: Planowanie trasy bez uwzględnienia parametrów robota

Generowanie bezkolizyjnej trasy jest następujące: jeśli robot znajduje się w komórce c_{ij}^{θ} oraz ma prędkość $(v_{ij}^{\theta}, \omega_{ij}^{\theta})$, to wyznaczany jest zbiór komórek $\{c_{kl}^{\alpha}\}$ należących do sąsiedztwa c_{ij}^{θ} . Ze zbioru wybieramy komórkę x_{ij} o maksymalnym pobudzeniu (wartości).

Biorąc pod uwagę aktualne wartości $(v_{ij}^0, \omega_{ij}^0)$ oraz ograniczenia przyspieszenia możliwa jest sytuacja, że w chwili dojazdu do wybranej komórki robot osiągnie wyznaczone wcześniej wartości $(v_{kl}^\alpha, \omega_{kl}^\alpha)$. W tym przypadku komórka ta wyznacza kolejne położenie pojazdu, a wartości przyspieszeń zmieniają się tak, aby osiągnąć wyznaczone wartości parametrów. Jeśli warunek ten nie jest spełniony, to kolejne położenie pojazdu jest wyznaczone przez taką komórkę, że wyznaczone wartości $(v_{kl}^\alpha, \omega_{kl}^\alpha)$ mogą być przez robota osiągnięte, następuje jednocześnie maksymalne zmniejszenie prędkości liniowej i kątowej. Rys. 6 przedstawia sytuację, w której robot jadący z prędkością 1 m/s otrzymał polecenie dojechania do zaznaczonego celu oraz rozkład prędkości w czasie wykonywania manewru. Wyniki otrzymano z symulatora robota.



Rys. 6. Dojechanie do celu z dużą prędkością

5. EKSPERYMENT

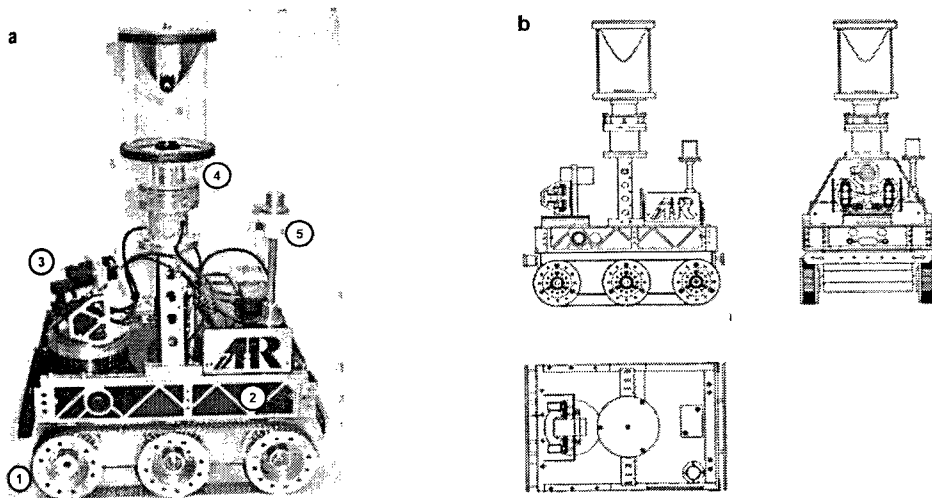
Opisana w pracy metoda jest w fazie wstępnych eksperymentów prowadzonych symulacyjnie oraz przy użyciu dwóch typów robotów mobilnych. Metoda może bazować na znanej mapie otoczenia lub mapa ta może być tworzona podczas wykonywania zadania. Drugi wariant jest mniej korzystny, ponieważ planowana trasa może nie być optymalna.

W badaniach eksperymentalnych wykorzystano dwa typy robotów: *Pioneer 1* oraz *Navigator I*. *Pioneer 1* wykorzystany był w eksperymentach, w których wymagana jest duża prędkość pojazdu. Układ napędowy robota *Pioneer 1* pozwala na przemieszczanie się z prędkością 1 m/s. Niestety duża prędkość, duże gumowe koła oraz stosunkowo mała rozdzielczość przetworników obrotowo impulsowych powodują powstawanie dużych błędów odometrii, co uniemożliwia wykonywanie zadania przy złożonej ścieżce, bez dodatkowych systemów lokalizacji.

Navigator I (rys. 7) jest miniaturowym robotem mobilnym specjalnie zaprojektowanym do prowadzenia prac badawczych z dziedziny nawigacji robotów. Jego niewielkie rozmiary oraz bogate wyposażenie sensoryczne pozwalają na precyzyjne pomiary przemieszczeń [4].

Robot oprócz sensorów odometrycznych został wyposażony jest w kompas elektroniczny oraz żyroskop, dzięki czemu możliwe jest precyzyjne określenie orientacji pojazdu. Dodatkowo posiada obrotową głowicę wyposażoną w dalmierze optyczne i sensor wizyjny. W korpusie umieszczono zestaw optycznych sensorów bliskiego zasięgu, których zadaniem jest wykrywanie przeszkód przed i po bokach robota.

Głowica umożliwia wykrywanie przeszkód w odległości do 1,2 m od robota w zakresie kątowym 180°. Aby zwiększyć wiarygodność pomiaru układ wyposażono w trzy naprzemiennie działające sensory. Dwa sensory dalekiego zasięgu zorientowane są pionowo i ustawione do siebie równoległe. Trzeci jest sensorem krótkiego zasięgu. W głowicy zorientowany jest poziomo. Zastosowanie takiej budowy głowicy wynika z faktu, że sensory dalekiego zasięgu posiadają strefę martwą, a początkowa ich charakterystyka jest nieliniowa i niejednoznaczna. Zastosowanie dalmierzy o różnym zasięgu pozwala na określenie położenia przeszkody w całym zakresie. Wbudowana kamera pozwala na określenie innych cech obiektów, co nie zostało wykorzystane w opisywanym eksperymencie.



Rys. 7. Navigator I: a) podstawowe moduły: 1 - moduł jezdny, 2 - moduł centralny, 3 - głowica skanująca, 4 - układ sensora wizji dookólnej, 5 - kompas elektroniczny, b) widok ogólny

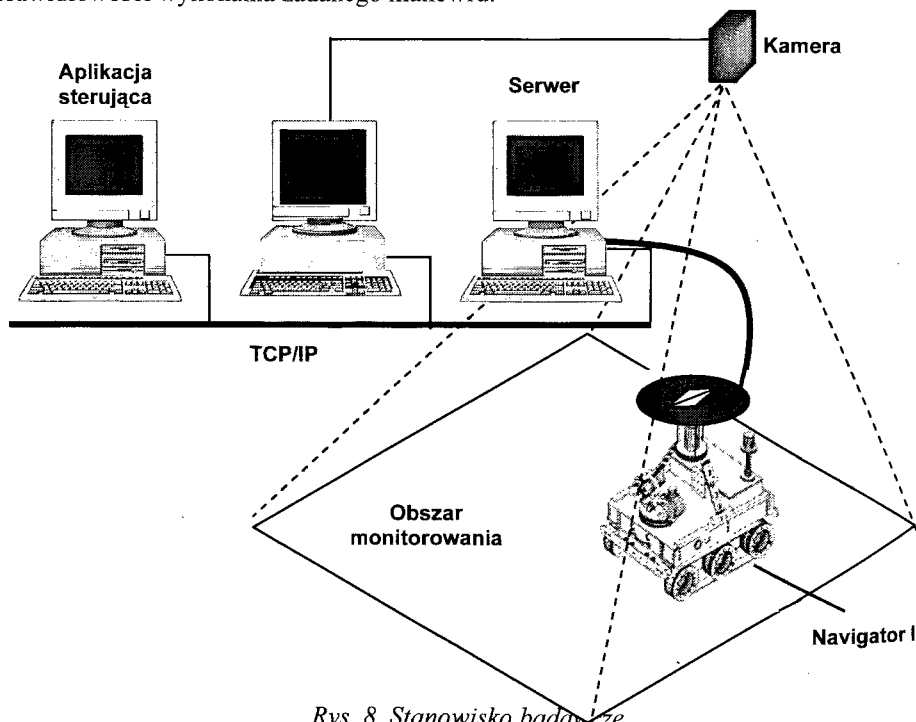
Robot mobilny *Navigator I* nie jest jednostką autonomiczną. Ze względu na niewielkie rozmiary pojazdu nie było możliwe zainstalowanie na jego pokładzie wydajnego komputera, zdolnego obsłużyć wszystkie urządzenia znajdujące się w robocie oraz dokonać złożonych obliczeń związanych z wykonywanym zadaniem. Robot został wyposażony w uproszczony mikrokomputer pokładowy typu *Slave* połączony przez łącze szeregowe RS232 z jednostką nadrzędną. Jego zadaniem jest obsługa układów pokładowych i komunikacja z *Masterem*.

Układ sterowania w warstwie funkcjonalnej spełnia następujące zadania: komunikacyjne, trakcyjne, sensoryczne, pomiarowe, sterowania oraz diagnostyczne.

Aplikacja sterująca jest rozproszona. Robot połączony jest z linuksowym serwerem nadzorującym komunikację z komputerem pokładowym. Nadrzędna aplikacja sterująca łączy się z serwerem robota przez połączenie sieciowe wykorzystując protokół TCP IP. Zaletą tego rozwiązania jest to, że aplikacja może być napisana w dowolnym środowisku programistycznym oraz uruchamiana na oddzielnej jednostce komputerowej.

W przypadku potrzeby dużej mocy obliczeniowej możliwe jest wykorzystanie kilku jednostek komputerowych pracujących w jednej sieci. Schemat stanowiska badawczego przedstawiono na rys. 8.

Aby robot mógł osiągnąć wyznaczony cel według opisanej metody konieczne jest określanie jego pozycji i orientacji w każdym punkcie. Pomiar odometryczny jest wystarczający jedynie do określania przemieszczeń na odcinkach prostych. Ponieważ robot ma podwozie gąsienicowe, podczas zmiany orientacji powstają duże błędy pomiarowe, wynikające z poślizgów. Dzięki żyroskopowi oraz elektronicznemu kompasowi możliwe jest precyzyjne określenie orientacji ze stałym błędem. W przypadku zastosowanego żyroskopu piezoelektrycznego pomiar jest przyrostowy, czego skutkiem jest kumulacja błędów przy wielu manewrach. Sensor generuje sygnał proporcjonalny do prędkości obrotowej. Z tych powodów w systemie wykorzystywany jest jedynie do sprawdzania prawidłowości wykonania zadanego manewru.



Rys. 8. Stanowisko badawcze

Zaletą sensora jest krótki czas pomiaru, co jest szczególnie istotne przy wykonywaniu szybkiej sekwencji zwrotów. Zaletą kompasu elektronicznego jest stały poziom błędów w danych warunkach środowiskowych (lokalne pole magnetyczne związane np. z metalową konstrukcją nośną budynku). Niestety sensor ten podatny jest na zakłócenia od silników oraz innych urządzeń wytwarzających pole magnetyczne, co może powodować chwilowe zakłócenia wpływające na dokładność pomiaru. Innym problemem jest długi czas wykonywania pomiaru po zmianie orientacji, dochodzący do 3 s. Wywołuje to konieczność zatrzymania pojazdu lub przemieszczanie się w jednym ustalonym kierunku.

Połączenie obu sensorów żyroskopu i kompasu korzystnie wpłynęło na niezawodność systemu oraz szybkość jego działania. W przypadku wystąpienia chwilowych zakłóceń pola magnetycznego kompas generuje fałszywą informację o zmianie orientacji. Jeśli taka sytuacja nastąpi, system samo-lokalizacji robota sprawdza, czy nadrzędny układ sterowania wygenerował polecenie zmiany orientacji oraz czy żyroskop zarejestrował ją i czy była ona zgodna z rozkazem co do wartości i kierunku. W przypadku rozbieżności informacji z sensorów robot zatrzymuje się, po czym następuje następny pomiar orientacji za pomocą kompasu.

Eksperymenty przeprowadzono na specjalnie przygotowanym polu testowym (rys. 8). Dla sprawdzenia poprawności pozycjonowania robota wykorzystano zewnętrzny system lokalizacji wykorzystujący podwieszoną pod sufitem kamerę. Robot został wyposażony w specjalny znacznik. System kontrolny wyposażony we własną jednostkę komputerową w czasie rzeczywistym przekazuje do programu nadrzędnego informacje o położeniu i orientacji robota na polu testowym.

LITERATURA

- 1 *Barraquand B., Langois J.C., Latombe J. C.*: Numerical potential field techniques for robot path planning. IEEE Transactions on Robotics and Automation, vol. 22(2), 224 - 241, 1992
- 2 *Borenstein J., Koren Y.*: Real-time Obstacle Avoidance for Fast Mobile Robots. IEEE Transactions on Systems, Man, and Cybernetics, vol. 7, 1179-1187, 1989
- 3 *Borenstein J., Koren Y.*: Real-time Map-building for Fast Mobile Robot Obstacle Avoidance. SPIE Symposium on Advances in Intelligent Systems, 1990
- 4 *Chojceki R.*: Minirobot mobilny NAVIGATOR 1. PAR, 7 (2003) 11, 28-32, 2003
- 5 *Chua L., Young L.*: Cellular Neural Network. IEEE Transaction on Circuit System, 500-505, 1990
- 6 *Fox D., Burgard W., Thrun S.*: The Dynamic Window Approach to Collision Avoidance. IEEE Robot. Autom. Mag., vol.4, 23-33, 1997
- 7 *Kant K., Zucker S.*: Towards Efficient Trajectory Planning, The Path-Velocity Decomposition. Journal of Robotics Research, vol. 5(3), 72-89, 1986
- 8 *Konolige K.A.*: Gradient Method for Realtime Robot Control. IROS, 639--646, 2000.
- 9 *Khatib O.*: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. International Journal of Robotics Research, vol. 1, 90-98, 1992
- 10 *Latombe J.C.*: Robot Motion Planning. Kluwer Academic Publishers. MA Boston, 1992.
- 11 *Siemiątkowska B.*: Cellular Neural Network for Path Planning. SIRS, 125 - 130, 1994