

`Q//Na razie brakuje odpowiedzi na zagadnienie 4 Zapraszam do zabawy,
ja się zagłębiam w slajdy z wykładu, może coś znajdę.//
//Nie jestem pewien 10. Sprawdźcie to proszę.

Zagadnienia:

- 1. koncepcja architektury Brooks'a (wyjaśnić pojęcie poziomu kompetencji);**
- 2. budowa modułu w architekturze Subsumption;**
- 3. pojęcie (def.) nawigacji reaktywnej + problemy związane z określaniem funkcji odwzorowującej;**
- 4. sterownik oparty na logice rozmytej (architektura sterownika);**
- 5. pojęcie (def.) "podstawowego" problemu planowania ruchu;**
- 6. pojęcie (def.) przestrzeni konfiguracyjnej;**
- 7. pojęcie (def.) siatki punktów decyzyjnych;**
- 8. metody planowania trasy (w tym: ideologia klasyfikacji);**
- 9. metoda Grafu Widzialności (koncepcja);**
- 10. metoda zmodyfikowanego Grafu Widzialności (koncepcja).**

1. koncepcja architektury Brooks'a (wyjaśnić POZIOM KOMPETENCJI);

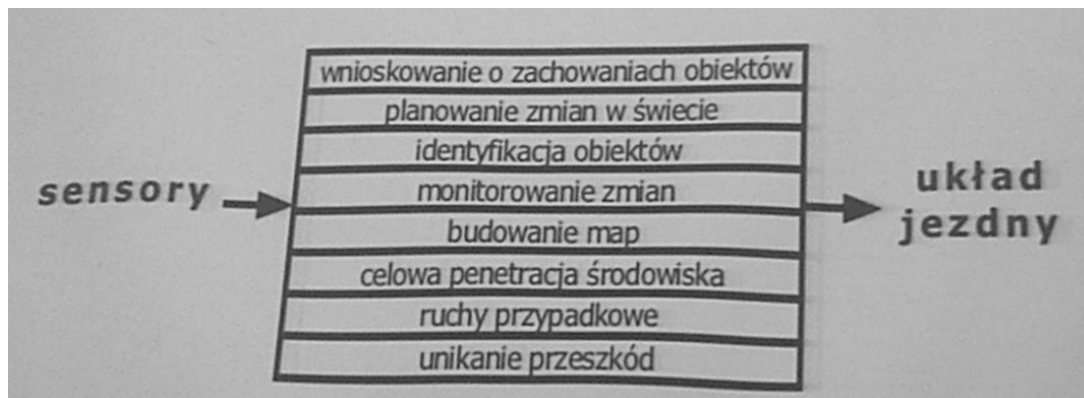
Zgodnie z koncepcją zaproponowaną przez Rodney'a Brooks'a, zadanie sterowania lokomocją dekomponujemy przez wyróżnienie nie samych zachowań, lecz tzw. "poziomów kompetencji" działania robota.

Poziom kompetencji - opis pożądanej klasy zachowań robota w oczekiwanych sytuacjach

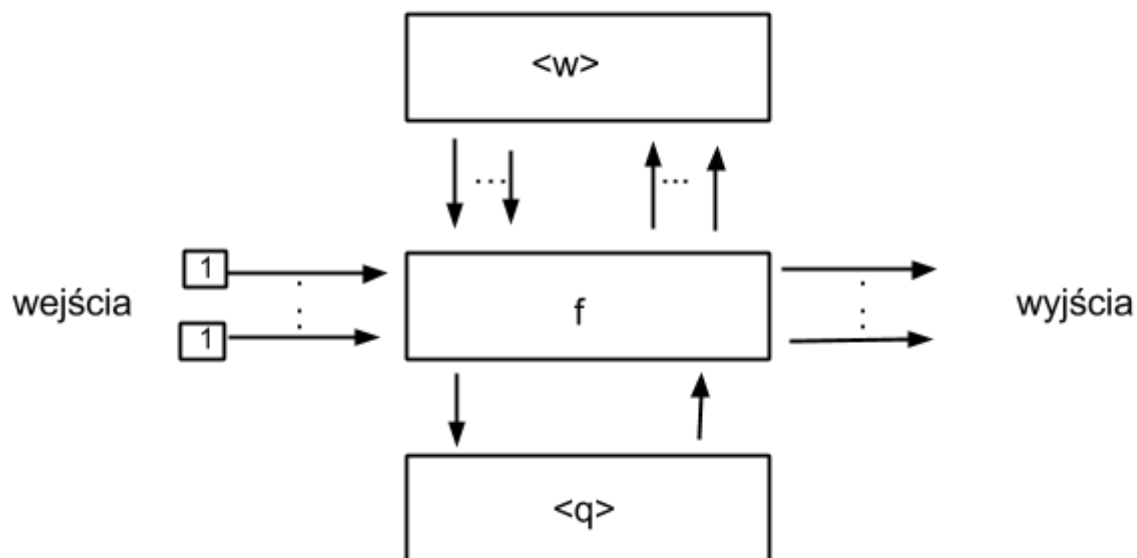
Poziomy kompetencji i struktura systemu:

- 0 - unikanie kontaktu z obiektami (ruchomymi lub stacjonarnymi);
- 1 - "przypadkowe" poruszanie się w otoczeniu bez kolizji z obiektami;
- 2 - "poznawanie" świata, poprzez rozpoznawanie i przemieszczanie się w kierunku najbardziej odległych miejsc (korytarze);
- 3 - budowanie mapy otoczenia i planowanie dróg pomiędzy poznаныmi miejscami
- 4 - wykrywanie zmian w otoczeniu statycznym;
- 5 - wnioskowanie o otoczeniu w kategoriach "iidentyfikowalności" obiektów i realizacja zadań związanych z pewnymi obiektami;
- 6 - tworzenie i wykonywanie planów prowadzących do pożądanej zmiany stanu otoczenia
- 7 - wnioskowanie o zachowaniu się obiektów i odpowiednie do tego modyfikowanie planów

Struktura systemu sterowania uwzględniająca opisaną koncepcję dekompozycji ma następującą postać:



2. BUDOWA MODUŁU w architekturze Subsumption.



<w> - zmienne wewnętrzne modułu

f - funkcja przetwarzania

<q> - stany modułu

Każdy stan automatu ma swój symbol. Istnieje wyróżnione wejście RESET. Po odebraniu sygnału RESET moduł znajduje się w stanie NIL.

3. Formalna definicja nawigacji reaktywnej

Def. **Przestrzeń percepcyjna S**, jest to przestrzeń, której każda współrzędna reprezentuje sygnał z sensora (wejście systemu nawigacji).

Def. **Przestrzeń komend Y**, jest to przestrzeń, której każda współrzędna reprezentuje współrzędną (parametr) komendy sterującej (wyjście systemu nawigacji)

Def. **Odwzorowanie**: przestrzeni percepcyjnej S na przestrzeń komend Y, określamy mianem nawigacji reaktywnej.

Budowanie systemu nawigacji reaktywnej polega na określeniu takiego odwzorowania. Może ono pochodzić z rodziny wstępnie zdefiniowanych funkcji (tak jak w metodzie pola potencjalnego) lub też może być wytworzone przy użyciu „uniwersalnych” aproksymatorów (takich jak sieci neuronowe lub logika rozmyta).

Problemy związane z określaniem odwzorowania M (funkcji odwzorowującej?)

Oznaczmy przez M^* odwzorowanie zdolne doprowadzić robota do celu. Przy określania odwzorowania M dla systemu nawigacji, pojawiają się dwa zasadnicze problemy:

- 1.
2. M musi być **lokalnie dobrą aproksymacją** odwzorowania AF . Lokalnie - oznacza tu, że ono obowiązuje w pewnym regionie przestrzeni percepcyjnej (czyli szczególnym przedziale wartości danych sensorycznych) - np.: robot musi zakręcić przed ścianą;
- 3.
4. M musi przybliżać M^* w **całej przestrzeni percepcyjnej S** (tzn. zapewnić dobrą reakcję w każdej sytuacji percepcyjnej - dla każdej kombinacji danych sensorycznych).

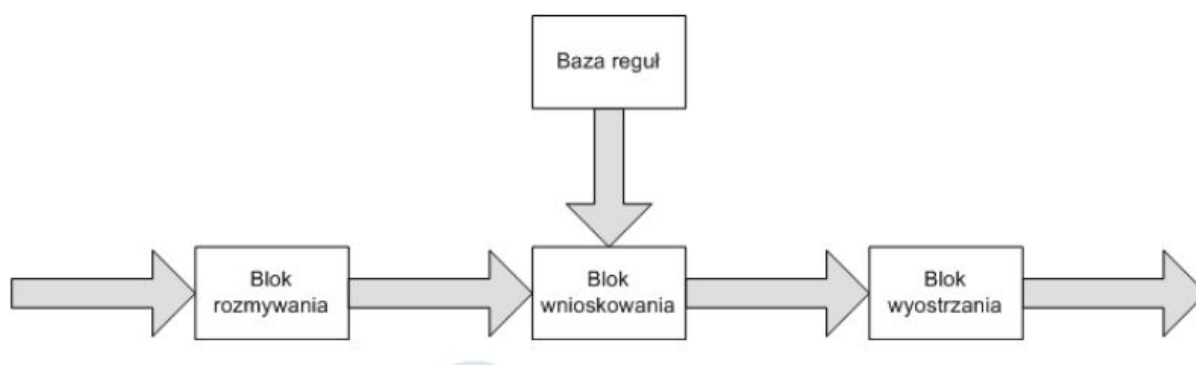
Drugi problem jest trudny gdyż robot napotyka dużą różnorodność sytuacji powiązanych z dużą liczbą różnych reakcji. Ponadto, trudno jest wyobrazić sobie rzeczywistą sytuację związaną z każdym regionem przestrzeni S - trudno więc jest określić reakcję związaną z każdym punktem tego regionu. Dlatego większość systemów nawigacji buduje się odwrotnie: rozważa się standardowe sytuacje rzeczywiste (np.: śledzenie ściany, przechodzenie przez drzwi, ruch w korytarzu). Jednak wtedy nie ma gwarancji, że regiony odpowiadające tym sytuacjom pokryją całą przestrzeń percepcyjną i nie będą na siebie zachodziły. Zachodzenie regionów wymaga łączenia lub arbitrażu komend.

Określona reakcja jest odpowiedzią na określoną sytuację.

Dla rozwiązania **pierwszego problemu** (lokalnie dobrej aproksymacji M^*) należy określić komendy, spowodują wymaganą reakcję. W większości przypadków określenie dokładnego modelu ruchu pojazdu jest prawie niemożliwe, np.: ze względu na nieliniowe równania różniczkowe (więzy nieholonomiczne), nieznane poślizgi kół itp. Trudno też określić precyzyjny model sensorów, ponieważ w układzie zbierania danych występuje dużo niekontrolowanych parametrów jak: kształt przeszkody, materiał i faktura jej powierzchni, temperatura otoczenia itp. Dlatego nie można uzyskać funkcji odwzorowującej opartej na matematycznym opisie sytuacji i trzeba zastosować metody przybliżone.

4. sterownik oparty na logice rozmytej (architektura sterownika);

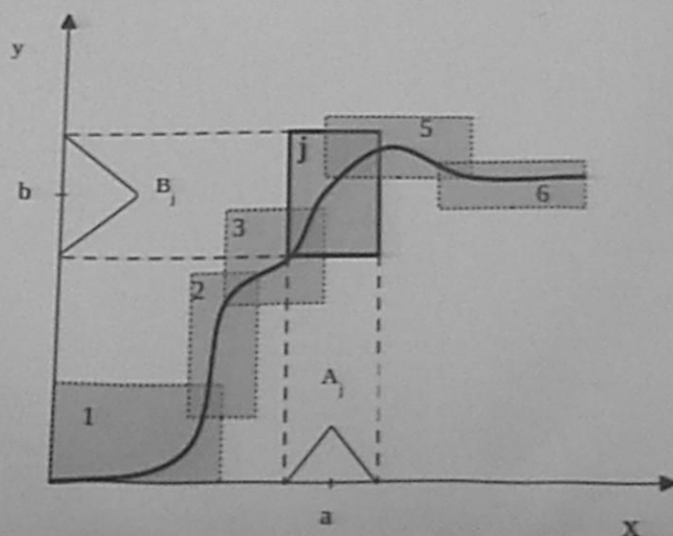
Dokument

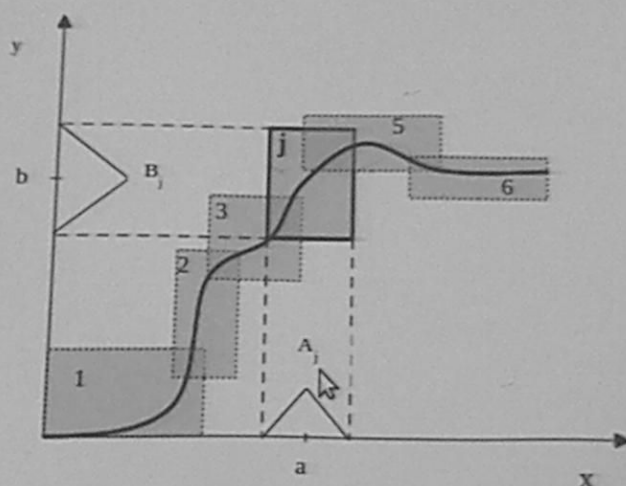


img. Sterownik rozmyty.

Sterownik reaktywny oparty na logice rozmytej

Sterownik Fuzzy (lub rozmyta pamięć asocjacyjna) jest odwzorowaniem pomiędzy zmiennymi wejściowymi a wyjściowymi (te zmienne są oczywiście zbiorami rozmytymi a nie wartościami numerycznymi tak jak w przypadku klasycznej funkcji przejścia). Prościej można powiedzieć, że znane są pewne pary punktów (*input*, *output*) należące do funkcji, którą chcemy modelować. Przyjmuje się, że funkcja ta jest ciągła, co znaczy że jeśli znany jest punkt (*input*, *output*), to można powiedzieć: „wokół wartości *input* wartość funkcji jest bliska wartości *output*”. Taka funkcja jest modelowana przez zestaw relacji zbiorów rozmytych.





Rysunek pokazuje sterownik fuzzy w postaci „łat aproksymujących” funkcję M' . Relacja (A_j, B_j) między zbiorami rozmytymi jest wyprowadzona z relacji (a, b) pomiędzy punktami numerycznymi. Trójkąty reprezentują funkcje przynależności zbiorów rozmytych A i B .

Przykładowo, łata j odwzorowuje regułę opisaną relacją pomiędzy zbiorem „na prawo od linii” a „kątem skrętu” robota (lub „prędkością”).

5. Pojęcie „podstawowego” problemu planowania ruchu.

Aby ułatwić rozwiązanie problemu planowania należy z tego zbioru wyodrębnić pewne zagadnienia podstawowe. Można to osiągnąć przez przyjęcie określonych założeń uproszczonych.

Przyjmujemy że:

- robot jest jedynym poruszającym się obiektem w przestrzeni roboczej oraz ignorujemy jego własności dynamiczne (unikając problemów czasowych);
- ograniczamy taki ruch do bezkontaktowego, tak że problemy związane z interakcją mechaniczną pomiędzy dwoma obiektami można pominąć.

Wynikający z takich uproszczeń Podstawowy Problem Planowania Ruchu można postawić następująco:

Niech robot A będzie pojedynczym sztywnym obiektem poruszającym się w Przestrzeni Euklidesowej W , zwanej przestrzenią roboczą, reprezentowanej przez R^N , $N=2$ lub 3 .

Niech $B_1 \dots B_q$ będą sztywnymi obiektami umiejscowionymi w W , na stałe, stanowiącymi przeszkody dla ruchu robota. Nazywamy je przeszkodami.

Geometria obiektów $A, B_1 \dots B_q$ oraz ich lokalizacja w W (za wyjątkiem A) są dokładnie znane. Założmy ponadto, że żadne inne ograniczenia nie krępują ruchu A (A jest swobodnie poruszającym się obiektem).

Problem: Mając początkową i końcową pozycję (położenie i orientację) A w W , należy określić ścieżkę t opisującą ciągłą sekwencję pozycji A unikających kontaktu z przeszkodami

B_i , prowadzącą od pozycji początkowej do końcowej, lub zasygnalizować jej brak - gdy taka ścieżka nie istnieje.

6. Przestrzeń konfiguracyjna;

Koncepcja Przestrzeni Konfiguracyjnej polega na reprezentowaniu robota jako punktu w odpowiedniej przestrzeni - właśnie Przestrzeni Konfiguracyjnej Robota - oraz odwzorowaniu przeszkód do tej przestrzeni. To odwzorowanie transformuje Problem Planowania Ruchu Obiektu Wymiarowego w Problem Planowania Ruchu Punktu. To z kolei powoduje, że ograniczona ruchu robota stają się bardziej jawne.

Konfiguracja obiektu jest specyfikacją pozycji wszystkich punktów tego obiektu w ustalonym (zafiksowanym) układzie odniesienia. Stąd, konfiguracja q obiektu A jest specyfikacją pozycji (położenia τ i orientacji ϕ) układu F_A w F_W . Podzbiór W zajmowany przez A w konfiguracji q oznaczmy $A(q)$. Podobnie punkt a ciała A w konfiguracji q oznaczmy $a(q)$ w W .

Przestrzeń konfiguracyjna C robota A jest to przestrzeń $C = \{q\}$ wszystkich q obiektu A . (a nie przestrzeń W zajmowana przez robota w możliwych konfiguracjach q)

7. Siatka punktów decyzyjnych

Każda rozpoznawalną pozycję robota w W , reprezentowaną przez $q=(\tau, \phi)$ w C_{free} , można traktować jako stan (reprezentowany przez q). Spośród tych stanów wybieramy takie, dla których istnieje prosty sposób określenia działania prowadzącego do osiągnięcia sąsiedniego stanu:

gdzie: y - działanie, S - zbiór wyróżnionych stanów spójnego regionu C_{free} .

Wyróżnione stany q' określamy mianem punktów decyzyjnych. Dla potrzeb planowania trasy, ze zbioru wszystkich punktów decyzyjnych, wybieramy najmniejszą liczbę punktów, które wyznaczają dozwolony sposób poruszania się w otoczeniu. Przestrzenny układ tak wybranych stanów określamy mianem siatki punktów decyzyjnych i oznaczamy przez S^* . Wiedza reprezentowana siatką punktów decyzyjnych S^* , może być wyrażona zbiorem reguł postępowania (R) lub jawnie grafem (G), którego wierzchołki odpowiadają wyróżnionym stanom q^* epsilon S^* , a krawędzie działaniom y epsilon Y (prowadzącym do osiągnięcia sąsiedniego stanu).

8. Planowanie trasy

Planowanie trasy polega zasadniczo na:

a) wygenerowaniu reprezentacji siatki S^* punktów decyzyjnych;

b) poszukiwaniu w oparciu o tą reprezentację ciągu przemieszczeń (działań) przeprowadzających robota z punktu q_{init} do punktu q_{goal} , spełniającego przyjęte kryterium jakości ścieżki.

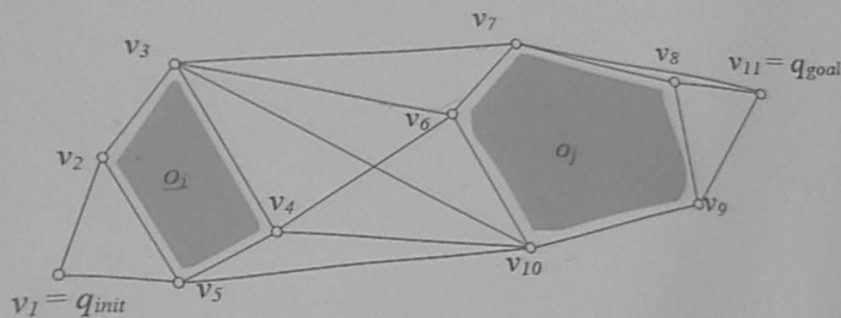
Ze względu na sposób reprezentacji wiedzy o otoczeniu można wyróżnić pięć grup metod planowania:

- metody mapy dróg;
- metody dekompozycji komórkowej;
- metody rastrowe;
- metody pól potencjałów;
- metody wektorowe.

9. Metoda Grafu Widzialności (visibility graph)

Metoda Grafu Widzialności opiera się na odwzorowaniu spójności części swobodnej (C_{free}) przestrzeni konfiguracyjnej (C) robota. Na siatkę 1-wymiarowych krzywych zwanych mapą dróg R . Graf Widzialności (VG) jest grafem nieskierowanym, którego wierzchołkami są konfiguracja początkowa q_{init} i docelowa q_{goal} robota oraz wszystkie wierzchołki C -przeszkód a krawędziami odcinki proste łączące wszystkie "widzące się" wierzchołki. Efektywne algorytmy konstruują graf VG oraz znajdują w nim najkrótszą ścieżkę w czasie $O(n^2)$, gdzie n to liczba wszystkich wierzchołków C -przeszkód. Ścieżka taka stanowi linię łamaną przechodzącą przez wierzchołki przeszkód.

Graf VG zbudowany na przeszkodach z rys. 2, ilustruje rys. 3.



Rys. 3. Graf Widzialności.

10. Metoda zmodyfikowanego Grafu Widzialności

Modyfikacja w tej metodzie polega na transformacji grafu VG na taki graf skierowany MVG, którego łuki leżące w C nie prowadzą do wierzchołków przeszkód lecz do nowych wierzchołków usytuowanych na ich krawędziach. Graf MVG ma taką właściwość, że co najmniej co drugi odcinek każdej poprowadzonej w nim ścieżki leży w domknięciu przestrzeni swobodnej (C_{free}), czyli przebiega wzdłuż krawędzi przeszkody. Podczas

realizacji działania robota, na tych odcinkach można zastosować sterowanie lokalne oparte na bezpośrednim sprzężeniu zwrotnym, zapewniające prowadzenie robota wzdłuż ściany.

