

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY
INSTYTUT STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ

PRACA MAGISTERSKA
na kierunku ELEKTROTECHNIKA



Marcin Biernacki
Nr imm. 176709

Rok. akad. 2004/2005
Warszawa, 02.09.2005

*INTELIGENCJA GRUPOWA – WSPÓŁPRACA ZESPOŁOWA W ŚRODOWISKU
ROBOTÓW MOBILNYCH*

Zakres pracy:

1. *Wprowadzenie do inteligencji grupowej robotów*
2. *Przedstawienie systemów z rodziny robotów mobilnych*
3. *Zdefiniowanie funkcji i określenie działań współpracujących w grupie urządzeń*
4. *Metody lokalizacji robota w środowisku oraz tworzenia mapy terenu*
5. *Opracowanie modelu eksploracji w ujęciu inteligencji grupowej robotów – symulacja*
6. *Podsumowanie i wnioski*

Podpis i pieczęć

Kierownika Zakładu Dydaktycznego

Opiekun naukowy:
dr inż. Witold Czajewski

Termin wykonania: grudzień 2005
Praca wykonana i zaliczona pozostaje własnością
Instytutu i nie będzie zwrócona wykonawcy

1. Wprowadzenie do inteligencji grupowej robotów	4
1.1 Roboty funkcjonujące autonomicznie	5
1.2 Wizja grupowego współdziałania	6
1.3 Przyczyny zainteresowania inteligencją zbiorową.	6
2. Przedstawienie systemów opartych na współdziałaniu zespołowym	9
2.1 Zdefiniowanie pojęcia inteligencji grupowej	9
2.2 Przyroda źródłem inspiracji dla systemów złożonych	11
2.2.1 Bakterie	11
2.2.2 Mrówki	12
2.2.3 Wilki	13
2.2.5 Ludzie	14
2.3 Architektura złożonych systemów inteligencji grupowej	15
2.3.1 Konfiguracja grupowa współpracujących urządzeń mobilnych;	15
2.3.2 Cechy indywidualne i charakterystyka zespołu	18
2.3.3 Profilowanie akcji współdziałających jednostek	19
2.4 Rozproszona sztuczna inteligencja - DAI	19
2.5 Komunikacja, interakcja, współpraca	21
2.5.1 Właściwości komunikacji	21
2.5.2 Podstawowe kategorie komunikacji	22
3. Eksploracja i konstrukcja map środowiskowych	26
3.1 Samo-lokalizacja jednostki mobilnej	26
3.1.1 Pozycjonowanie względne i odometria – identyfikacja błędów.	27
3.1.1.1 Pomiary i eliminacja błędów	32
3.1.1.2 Oszacowanie błędów systematycznych	35
3.1.1.2 Oszacowanie błędów niesystematycznych	43
3.1.2 Pozycjonowanie bezwzględne	48
3.1.2.2 Znaczniki	48
3.1.2.3 Filtr Kalmana	49
3.1.3 Czujniki i rozwiązania aplikacyjne	49
3.2 Budowa i kompozycja mapy	50
3.2.1 Pozycjonowanie względem mapy	50

3.2.2 Metoda FastSLAM	53
4. Eksploracja nieznanego terenu w ujęciu inteligencji grupowej robotów	56
4.1 Opracowanie projektowe symulacji dla grupy współdziałających robotów	56
4.2 Założenia dla projektu eksploracji terenu przez grupę robotów mobilnych	57
4.3 Modelowanie jednostek i algorytmy - koncepcja	60
4.4 Postać aplikacyjna	61
4.4.1 Model mapy rastrowej	61
4.4.2 Model robota	64
4.4.3 Odczyt danych ze środowiska i zapis w pamięci	68
4.4.4 Schemat modelowania ruchu i błędów pozycjonowania	69
4.4.4.1 Wykonanie ruchu	70
4.4.4.2 Błędy pozycjonowania	71
4.4.6 Współdziałanie jednostek i algorytm ruchu	76
4.4.7 Schemat blokowy programu i interfejs graficzny	79
5. Uwagi końcowe i wnioski	92
Literatura i załączniki	94

SPIS RYSUNKÓW

- Rys 1. Przykłady architektury systemów składających się z wielu jednostek mobilnych.
- Rys 2. Przykład systemu przedstawiającego typ architektury rozproszonej.
- Rys 3. Przykład systemu przedstawiającego typ architektury hierarchicznej.
- Rys 4. Zobrazowanie postaci grup robotów i typu jednostek wchodzących w ich skład.
- Rys 5. Podział aplikacji wykorzystujących elementy rozproszonej sztucznej inteligencji.
- Rys 6. Fazy przetwarzania informacji w aplikacjach rozproszonej sztucznej inteligencji.
- Rys. 7 Przykład poglądowy robota mobilnego bazującego na różnicowym napędzie kół.
- Rys. 8 Elipsy błędów – niedokładności pozycjonowania pojazdu.
- Rys. 9 Nierównomierne obciążenie pojazdu i deformacja kół – zmiana średnicy obliczeniowej.
- Rys. 10 Pozycjonowanie pojazdu względem otaczającego środowiska – pozycja startowa.
- Rys. 12 Rzeczywista ścieżka ruchu robota.
- Rys. 13 Przykład wstępnej korekcji pozycjonowania robota.
- Rys. 14 Efekt dwóch dominujących błędów systematycznych Ed oraz Eb, w przypadku testu „kwadratu” dla ruchu robota w kierunku przeciwnym do ruchu wskazówek zegara.
- Rys. 15 Przykładowe wyniki dla dwukierunkowego testu „kwadratu”.
- Rys. 16 Przykładowe umiejscowienie przeszkody dla pojazdu w teście określającym błędy niesystematyczne.
- Rys. 17 Zmiany ścieżki pojazdu wskutek umieszczenia sztucznych przeszkód na drodze przejazdu.
- Rys. 18 Postać blokowa algorytmu wykrycia i oszacowania błędów ruchu.
- Rys. 19 Rysunek czujników wykorzystywanych w aplikacjach robotów mobilnych.
- Rys. 20 Postać blokowa algorytmu pozycjonowania pojazdów względem mapy.
- Rys. 21 Postać blokowa algorytmu pozycjonowania pojazdów względem mapy.
- Rys. 22 Przykład rzeczywistego robota mobilnego - PIONEER 2.
- Rys. 23 Konceptyjna postać robota umieszczonego na siatce rastrowej wraz z zasięgiem czujników odczytujących dane z bliskiego otoczenia.
- Rys. 24 Mapy w postaci siatki z odpowiednimi wartościami oznaczającymi jej zajętość.
- Rys. 25 Odwzorowanie małej przeszkody i jej zniekształconej geometrii na mapie rastrowej.
- Rys. 26 Konceptcja modelu robota i położenia czujników.
- Rys. 27 Umiejscowienia czujników – identyfikacja rastrów względem punktu centralnego.
- Rys. 28 Umiejscowienie czujników – ukierunkowanie pojazdu w 20-tu różnych pozycjach.
- Rys. 29 „Globalna” zasada działania symulacji.
- Rys. 30 Częściowa eksploracja mapy i powiększony fragment ze zróżnicowanym współczynnikiem pewności.
- Rys. 31 Szablon modelowania ruchu.
- Rys. 32 Schemat blokowy danych zewnętrznych, które wpływają na obliczenia i szacunki przeprowadzane przez robota.
- Rys. 33 Rysunek przedstawiający zależności pomiędzy czasem symulacji i ilością błędów dla zakłóceń systematycznych.
- Rys. 34 Rysunek przedstawiający zależności pomiędzy czasem symulacji i ilością błędów dla zakłóceń niesystematycznych.
- Rys. 36 Pokrywanie kilku obszarów przez grupę robotów.
- Rys. 37 Wykresy przedstawiające zależności procentowego wskaźnika odkrycia powierzchni w zależności od czasu badania oraz ilości pojazdów biorących udział w symulacji.
- Rys. 38 Wykresy przedstawiające zależności procentowego wskaźnika dokładności w zależności od czasu badania oraz błędów pozycjonowania pojazdów.

1. Wprowadzenie do inteligencji grupowej robotów

Świat, który otacza nas wszystkich, może dostarczać wielu wzorców, które z powodzeniem mogą być wprowadzone w przestrzeń „sztucznej inteligencji” kreowanej przez człowieka. To właśnie przyroda jest naturalnym środowiskiem, którego złożoność jest niemożliwa do pełnego i wyczerpującego opisanie przy pomocy matematycznych wzorów. Jednak naśladowując niektóre z tych „idealnych” rozwiązań, ludzie mogą stworzyć sztuczny świat, wzorujący się na żywych istotach i ich działaniu w naturalnym środowisku. Przy pomocy nowo powstałych systemów i algorytmów, wzorowanych na zachowaniu żywych osobników można próbować rozwiązać różne problemy, które przy wykorzystaniu „konwencjonalnej” automatyki i robotyki są bardzo trudne do rozstrzygnięcia.

Wraz ze wzrastającymi potrzebami w dziedzinach przemysłu czy też różnorodnych nowoczesnych technik należy rozszerzać pole działań na inne, innowacyjne rozwiązania, by lepiej wykorzystać wydajność istniejących systemów. Efektywność poczynań ku lepszemu zintegrowaniu sztucznej inteligencji razem z otaczającym nas światem bardzo obiecująco przedstawia wizja „inteligencji grupowej”. Scharakteryzować ją można jako zespół urządzeń - robotów współdziałających ze sobą w dążeniu do osiągnięcia określonego celu. Podstawą takiego podejścia do problemu rozwiązywania postawionych przed sztucznymi systemami zadań, są stosunkowo proste roboty, wykonujące mało skomplikowane prace. Jeśli za jakiś czas zostaną stworzone w pełni autonomiczne i spełniające określone, wszechstronne funkcje roboty, kolejnym krokiem będzie stworzenie systemów bazujących na takich jednostkach, które będą się ze sobą porozumiewały i współdziałały. Jednak już teraz należy podjąć prace opiewające o elementy z zakresu „inteligencji grupowej”, gdzie pozornie trudne problemy mogą być rozwiązane przez niezbyt złożone jednostki. Bazując na obserwacjach świata przyrody – zachowaniu zwierząt i owadów – można wykreować stosunkowo mało skomplikowane roboty, gdzie głównym założeniem będzie ich współpraca i wykonywanie ściśle określonych działań. Pojawia się więc potrzeba zdefiniowania i opisanie zadań jakie mogłyby spełniać takie systemy złożone z wielu jednostek, wchodzących w skład grup robotów.

1.1 Roboty funkcjonujące autonomicznie

Stworzone w laboratoriach i wykorzystywane coraz częściej urządzenia są coraz bardziej skomplikowane, jak i sterujące nimi algorytmy. Wiele jednak pozostało pracy przed zespołami starającymi się skonstruować wszechstronnego i autonomicznie działającego robota, który będzie odwzorowywał w każdym elemencie istotę ludzką. Coraz więcej pojawia się w powszechnym użyciu maszyn stworzonych by pomagać ludziom w życiu codziennym - począwszy od prostych robotów służących do mało skomplikowanych prac, aż po bardzo rozbudowane systemy ogarniające swoimi możliwościami najnowsze zdobycze techniki. Wyróżnić tu można wiele różnych idei



przyświecających konstruktorom takich urządzeń: stworzenie roboto-zabawek o wyglądzie zwierząt domowych (psów) – projekt AIBO firmy Sony, opracowanie humanoidalnych robotów i „wszczepienie” w nie stylów ruchu zaczerpniętych od gatunku ludzkiego (projekt ASIMO) czy też wykreowanie robota symulującego do złudzenia

działanie człowieka, który pomógłby rozwikłać zagadki funkcjonowania mózgu - DB-KAWATO. Humanoidalne kształty u robotów sprawiają, że łatwiej jest z nimi nawiązać kontakt zwykłym użytkownikom, choć są to przede wszystkim produkty komercyjne, których użyteczność jest znacznie ograniczona w zaawansowanych projektach. Jednak większość prototypów wciąż przypomina człowieka i dość wyraźnie widać, iż dążenie do „upodobania” maszyny do człowieka nie ustaje. Ostatnimi czasy pojawiają się projekty z zakresu robotyki i neurochirurgii, gdzie naukowcy i grupy badawcze pragną w jak największym stopniu odwzorować organizm ludzki i w ten sposób poznać jego „zasady funkcjonowania”. Wprawdzie pojawia się coraz więcej ma-



szyn mających naśladować człowieka, jednak w niewielkim stopniu go przypominających. Nasz mózg składa się z miliardów neuronów tworzących zawiłą sieć, której żaden komputer na razie nie potrafi symulować.

Różnorakie podejścia do sposobu projektowania i funkcjonowania urządzeń mobilnych znalazły swoje zastosowanie w praktyce. Dotyczą one przede wszystkim właśnie robotów działających „w pojedynkę”, które nie potrafią lub też nie są w ogóle przystosowane do komunikowania się z innymi tego typu urządzeniami¹. Niemożliwe jest więc wspólne, bezkolizyjne i bezproblemowe wykonywanie postawionych przed nimi zadań.

1.2 Wizja grupowego współdziałania

Obserwacje przyrody i istniejących w niej naturalnych systemów współdziałania grupowego mogą dostarczyć wielu spostrzeżeń, które można wykorzystać do budowy modelu służącego do rozwiązania wielu różnorodnych problemów, z jakimi borykają się ludzie. Jest dosyć powszechnym faktem to, iż rozwiązywanie złożonych zadań nie wymaga skomplikowanych i wyszukanych zachowań jednostkowych. Na przykład kolonia mrówek czy też rój pszczół stanowi złożony i dobrze zorganizowany system społeczny spełniający wiele funkcji od zapewnienia wyżywienia kolonii po jej obronę. Natomiast obserwując pojedyncze osobniki nie zauważamy by wykonywały jakieś złożone czynności. Spostrzeżenie to potwierdzają badania, z których wynika, iż podstawą funkcjonowania tak złożonych społeczności jest dzielenie się swoimi doświadczeniami (wiedzą) z innymi członkami społeczności. Proste zachowania z mechanizmem dzielenia się wiedzą, przejawiające się w funkcjonowaniu złożonych grup nazywane są najczęściej **inteligencją zbiorową**.

1.3 Przyczyny zainteresowania inteligencją zbiorową.

Istnieje wiele przyczyn, które sprawiają, że zainteresowanie systemami złożonymi jest dosyć duże i ma przed sobą świetlaną przyszłość. Można tu wymienić przede wszystkim argumenty, które choć wydają się dość oczywiste, to z drugiej

strony przemawiają za tym, iż wizja współpracujących ze sobą jednostek jest bardzo obiecująca:

- Z uwagi na to iż nie ma żadnego scentralizowanego organu, który pełniłby funkcję nadrzędną grupy współdziałających urządzeń, cały system nadal będzie wypełniał postawione przed nim zadania nawet, gdy część jednostek ulegnie zniszczeniu;
- Zadania mogą być z natury bardzo trudne, bądź też niemożliwe do wykonania przez samodzielnie funkcjonującego robota – dla grupy urządzeń problem ten będzie znacznie umniejszony;
- Budowa kilku mało skomplikowanych w swej strukturze jednostek zamiast jednej może być o wiele łatwiejsza niż skonstruowanie jednego wszechstronnego urządzenia;
- Postawione zadania do wykonania, tudzież cele do osiągnięcia będą wykonane o wiele szybciej przez grupę niż przy wykorzystaniu samodzielnie działającego robota;
- Dobrze wykorzystanie przestrzeni przez zespół robotów, który może pokryć obszar o wiele większy niż jedno urządzenie;
- Paralelizm. Wykonywanie kilku czynności w tym samym czasie przez jednego robota nie jest często możliwe, z kolei w grupie urządzeń każda jednostka może być dedykowana dla jednej tylko czynności;

Patrząc od strony ekonomicznej, mamy również korzyści, które wydają się być bardzo istotne, bowiem budowany system grup robotów może być tańszy i bardziej elastyczny w użytkowaniu.

W szerszej perspektywie można spoglądać na system złożony z wielu jednostek jako układ rozpowszechniony w każdej dziedzinie życia. Ma on niewątpliwie ko-

rzyści i pewną przewagę nad innymi systemami. Od wielu tysięcy lat ludzie i niektóre gatunki zwierząt organizują się w grupy, aby mogli łatwiej zdobywać pożywienie i współdziałać w wielu czynnościach w ogromnej części swojego życia. Jednak nawet w naszych czasach można odnaleźć elementy, które ukazują, że samoorganizacja jednostek nie wymaga scentralizowanej kontroli i nadzoru. Począwszy od wcześniej wspomnianych kolonii mrówek, tworzących bardzo złożone w swej budowie skupiska, aż po teorie naukowe wielkich ekonomistów i filozofów jak np. Adam Smith, który nakreślił model doskonałego społeczeństwa, gdzie gospodarka nie potrzebuje interwencji centralnego rządu, ponieważ jest samoregulująca i samoorganizująca.

2. Przedstawienie systemów opartych na współdziałaniu zespołowym

Pojęcie **zachowania zbiorowego** (ang. collective behavior) odnosi się przede wszystkim do zachowania i sposobu postępowania agentów (jednostek wchodzących w skład zespołu) w systemach składających się z więcej niż jednego osobnika. Jako główny element w takiej interpretacji i jednocześnie budowie odpowiadającej temu systemowi należy nakreślić pojęcie **inteligencji grupowej**. Termin ten można scharakteryzować przez określenie sposobu współpracy pomiędzy jednostkami i sposobu w jaki na siebie oddziałują.

2.1 Zdefiniowanie pojęcia inteligencji grupowej

Aby opisać systemy zbudowane na fundamencie współpracy należy wcześniej zdefiniować lub co najwyżej spróbować określić pojęcia związane z tym obszarem badań. W literaturze odnaleźć można wiele definicji i opisów dotyczących tematu kooperacji, interakcji i sposobu współdziałania systemów złożonych z wielu osobników.

INTELIGENCJA to dosyć szerokie pojęcie, mogące kojarzyć się całkowicie odmiennie w różnych aspektach życia. Ograniczając się jednak do jej ogólnej postaci można ją zdefiniować za powszechnym słownikiem języka polskiego:

„**inteligencja** - zdolność posługiwania się rozumem; umiejętność zdobywania informacji i wykorzystywania ich”²

Jednak w technicznym znaczeniu, inteligencja objawiać się może i być rozumiana trochę inaczej niż w odniesieniu do człowieka - istnieją pojęcia dotyczące sztucznej inteligencji (ang. Artificial Intelligence), która wciąż znajduje się w sferze badań i jest wciąż rozwijana. Zawężając jednak zainteresowanie tylko do współdziałania gru-

powego możemy określić definicję **WSPÓŁPRACY** jednostek. W literaturze fachowej dotyczącej robotyki i szeroko rozumianej automatyki odnaleźć można kilka interesujących podejść do powyższego tematu **współpracy**:

- a) „wspólne działanie, ukierunkowane bezpośrednio na osiągnięcie wzajemnego, zamierzonego celu”³;
- b) „forma interakcji, bazująca zwykle na komunikacji”⁴;
- c) „wspólne działanie dotyczące osiągnięcia rezultatu w postaci zwiększenia wydajności lub oszczędności czasu”⁵;
- d) „dla postawionego zadania (problemu), grupa składająca się z wielu robotów przedstawia sobą współpracę zespołową jeśli, uwagi na leżący u jej podstaw mechanizm współdziałania, istnieje powiększenie użyteczności systemu”⁶;

Definicje te pokazują szeroki zakres użyteczności jak i zarówno możliwości, które przedstawia sobą system współdziałających ze sobą urządzeń „inteligentnych”. W różnoraki sposób można bowiem ukierunkować działania grupy urządzeń w zależności od postawionego przed nimi zadania. Ograniczając się do wąskiej, lecz dosyć zwężonej interpretacji można przedstawić przykłady dotyczące powyższych definicji.

Ad. a) Poruszany aspekt dotyczy przede wszystkim rozdziału zadań pomiędzy agentów wchodzących w skład współpracującej grupy, realizującej postawione zadanie (ang. Distributed Artificial Intelligence)

Ad. b) Występuje tu odniesienie do współdziałania opartego przede wszystkim na komunikacji i wymianie informacji;

Ad. c) Część ta dotyczy mierzalnych wielkości dotyczących współpracy, czego przykładem może być oszczędność czasu;

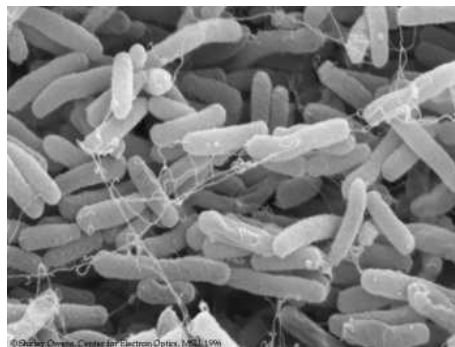
Jakkolwiek pozostają przedstawione definicje dotyczące robotów mobilnych, to jednak można tu zauważyć dosyć wyraziste trzy elementy, które charakteryzują inteligencję grupową i ideę kooperacji wśród urządzeń działających w jednym zespole. Są to: **zadanie**, **mechanizm współpracy** (działania), **wydajność**. Ostatnia definicja [d]) zawiera w sobie te trzy elementy i jest niejako „esencją” wcześniejszych sformułowań. Właśnie ją można przyjąć za główny opis i jednocześnie przedstawienie charakteru systemów opartych na inteligencji grupowej i wzajemnej współpracy urządzeń.

2.2 Przyroda źródłem inspiracji dla systemów złożonych

Źródłem narodzin systemów, których charakterystyczną cechą jest współdziałanie zespołowe, jest środowisko naturalne. Można tu wymienić kilka podstawowych i dość wyrazistych przykładów (zaczepniętych ze świata natury) na to, iż organizmy potrafią łączyć się w grupy i w nich sprawnie współpracować.

2.2.1 Bakterie

Współdziałanie pomiędzy (prostymi) organizmami sięga początków istnienia życia na Ziemi. Choć minęły miliony lat to jednak bakterie, które tworzyły nowe życie na naszej planecie są bardzo podobne do współczesnych nam mikroorganizmów. Już od początku ich istnienia tworzyły one swego rodzaju kolonie, w których się



rozwijały. Dopiero w ostatnich dziesięcioleciach stał się jasny fakt, iż większość bakterii komunikuje się poprzez szereg wyszukanych sygnałów.⁸ Jako przykład może posłużyć „*mykobakteria*”, która organizuje się w wielokomórkową strukturę grzybiącą. W tym przypadku, poszczególne jednostki produkują są pewne związki chemiczne, które służą do formowania się całej kolonii.

Biorąc pod uwagę główne elementy składające się na komunikację pomiędzy jednostkami w grupie, można w przypadku bakterii określić (choć z niezbyt dużą dokładnością) poszczególne ich części.

Dystans interakcji pomiędzy organizmami jest niezbyt duży zważywszy na wielkość samej bakterii. Z kolei stopień równoczesności interakcji jest uwarunkowany samą prędkością propagacji substancji chemicznych emitowanych przez te organizmy. Intencyjność dla przekazywania informacji (tj. składników służących celowi zbudowania złożonej struktury) jest nieobecna. Bakterie te wydzielają substancje chemiczne bezwiednie bez ich jakiegokolwiek interpretacji – proste zachowania prostych organizmów.⁸

2.2.2 Mrówki

Spośród żyjących gatunków owadów, najbardziej wnikliwie zostały zbadane przez człowieka mrówki, pszczoły i osy. Mrówki wykazują szereg elementów związanych ze współdziałaniem w grupie. Jako przykład posłużyć może sposób w jaki organizmy te zdobywają posiłek. Gdy jedna z mrówek „pracownic” odnajduje źródło nowych



zasobów, wtedy wracając do gniazda pozostawia na ziemi substancję zwaną feromonem. Z kolei nowe osobniki z gniazda będą podążały tą samą trasą co mrówka, która odnalazła żywność. Trasa jaką przemierzają pracownice może zostać „ulepszona” tj. może zostać skrócona przez maszerujące zastępy tych owadów, które odnajdą drogę na skróty. Jeśli źródło pożywienia zostanie całkowicie wyeksploatowane, wtedy na trasie nie będą pozostawiane feromony i z upływem czasu ich intensywność będzie zanikała.⁷

W powyższym przypadku dystans interakcji jest ograniczony do lokalnego obszaru w którym pozostawiony został swego rodzaju przekaznik – feromon. Czas pomiędzy nadaniem sygnału a jego odbiorem jest stosunkowo długi z uwagi na opóźnienie odnalezienia pozostawionego feromonu na ziemi przez jednego osobnika z kolonii (stopień równoczesności interakcji). Jest to komunikacja bardziej złożona niż u bakterii, jednak też stosunkowo prosta. Mrówki posiadają czułki, które mogą służyć jako

przekazniki informacji. Należy tu nadmienić, iż czułki te są tak samo zbudowane u każdego osobnika, więc nadanie i odbiór informacji odbywa się w ten sam sposób. Wyżej opisany mechanizm działający pomiędzy mrówkami może zostać przeniesiony do świata robotów, które mogą dzielić się informacjami (danymi) jeśli posiadałyby takie same czujniki. Informacje jednego z robotów byłyby identycznie interpretowane przez drugiego (lub też przez ich większą ilość). Takie podejście do komunikacji pomiędzy jednostkami w grupie znajduje szerokie zastosowanie w szczególności w zespołach o architekturze homogenicznej.

2.2.3 Wilki

Wilki żyją w naturalnych warunkach w stadach zwanych watahami (oprócz osobników odrzuconych żyjących samotnie) po 2-20 osobników (zwykle od 5 do 8), gdzie obowiązuje ścisła hierarchia obca innym gatunkom. Oznaczają one swój teren poprzez pozostawianie charakterystycznych substancji na ziemi – podobnie jak to czynią



mrówki w celu przekazania pewnych informacji. Ich przynależność do grupy odznacza się w wielu aspektach ich życia – polują w grupie, gdzie pojedyncze osobniki dokładnie obserwują zachowanie współtowarzyszy. Jest to zachowanie jednoznacznie bazujące na komunikacji poprzez stan (podejmowanie decyzji dotyczącej działania na podstawie zachowań innych agentów). Szczególnie należy podkreślić fakt, iż pojedyncze osobniki skupiają się na obserwacji lidera w grupie, który decyduje o określonych działaniach. Każdy z osobników w stadzie potrafi rozpoznać wszystkich w stadzie poprzez wygląd bądź też zapach. Komunikacja z kolei odbywa się z wykorzystaniem sygnałów dźwiękowych i przybieranie określonych postaw. Dystans komunikacji ograniczony jest w tym przypadku do zasięgu wzroku i słuchu.

Podobnie jak wilki postępują inne gatunki ssaków, lecz ich zachowanie i sposób komunikacji mogą odznaczać się pewnymi szczególnymi cechami, charaktery-

stycznymi dla danej grupy. Pośród nich można odnaleźć elementy komunikacji jawnej, poprzez stan czy też ukrytej.

2.2.5 Ludzie

Gatunek ludzki w pełni wykorzystuje różne sposoby komunikacji. Ludzie porozumiewają się pomiędzy sobą wykorzystując wiele zmysłów, w które obdarzyła ich natura. W odróżnieniu od innych gatunków potrafią oni porozumiewać się zarówno werbalnie jak i bez używania jakichkolwiek dźwięków. Jednostki ludzkie współpracują ze sobą na wiele różnych sposobów i żyją wspólnie tworząc wiele społeczności. Pojedyncze jednostki łączą się w związki z innymi (przyjaciele, małżeństwo, organizacje) lub już w chwili rozpoczęcia życia są przynależne do pewnej grupy (rodzina, państwo, itp.) gdzie mogą działać razem w celu osiągnięcia wspólnych korzyści. Tworząc grupę osobników są ściśle od siebie zależne i mimo, że często mogą dążyć do osiągnięcia pewnego celu to jednak potrafią czasami przejawiać zachowania czysto altruistyczne. Spoglądając jednak na niewyobrażalnie wielką złożoność jaką charakteryzują się związki pomiędzy jednostkami ludzkimi nie da się jednoznacznie i wyczerpująco przedstawić zależności występującymi pomiędzy nimi. Można jednak zaprezentować część sposobów jakimi mogą się porozumiewać poszczególne „osobniki” choć i tych jest bardzo dużo. Wymienić w tym miejscu należy przede wszystkim komunikację bezpośrednią:

- werbalną (rozmowa);
- poprzez gesty (język migowy, wskazywanie);

Są to formy porozumiewania się, w których istotną rolę mogą również odegrać takie elementy jak mimika twarzy, czy też odpowiednie akcentowanie, gdyż ich właściwe użycie może całkowicie zmieniać sens wypowiedzi i jej charakter. Istnieje również odpowiednik komunikacji pośredniej (niejawnej), gdzie poprzez „mowę ciała” można przekazać (świadomie lub nieświadomie) pewne informacje o sobie i swoich odczuciach.



Powyższe sposoby porozumiewania się mają jednak pewną wadę, a mianowicie przy ich wykorzystaniu niezbędna jest jednoczesna interakcja dwóch osób, gdzie wyróżnia się nadawcę i odbiorcę komunikatów (wszystkie jednostki mogą jednocześnie brać udział w wymianie informacji, lecz to może powodować niejednokrotnie swego rodzaju chaos). Wraz z rozwojem cywilizacji zrodziły się inne formy przekazu informacji między ludźmi, które możemy odczytywać i często podziwiać za swą formę. Począwszy od starożytnych hieroglifów, poprzez rzeźby, aż do współczesnych obrazów. Szczególną formą przekazu jest oczywiście język pisany, który odgrywa ogromną rolę w czasach wszechobecnego Internetu.

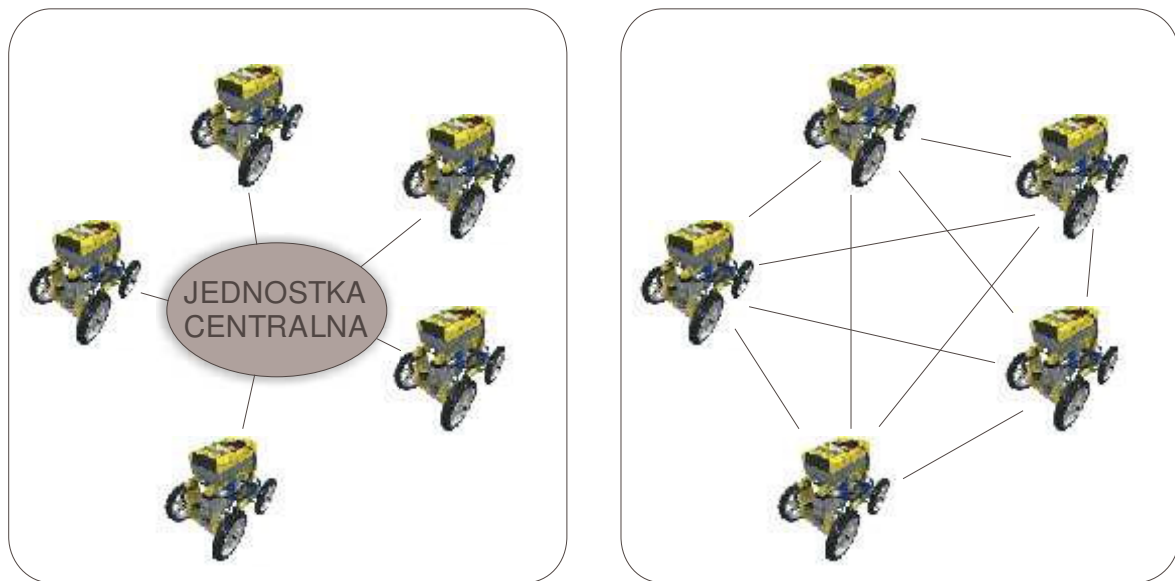
Komunikacja pomiędzy ludźmi, jak wcześniej wspomniano jest bardzo złożona i nie sposób wyszczególnić tu jej wszystkie aspekty mogące wpływać na porozumiewanie się jednostek. Może przybierać różne formy - jest niemalże w swej istocie nieograniczona...

2.3 Architektura złożonych systemów inteligencji grupowej

Grupa robotów musi z natury mieć charakterystyczne elementy, które sprawia, że będzie ją można określić mianem współpracujących urządzeń. Jednym z elementów jaki jest charakterystyczny dla systemów będących w sferze zainteresowań jest **architektura grupy**. Jest to kluczowa część infrastruktury, której kształt należy określić przy projektowaniu i implementacji systemów złożonych, ponieważ to ona decyduje w dużej mierze o możliwościach i efektywności pracy zespołu robotów mobilnych. Można w tym miejscu wyodrębnić takie elementy jak: budowa systemu ze względu na konfigurację grupy, cechy charakterystyczne jednostek wchodzących w skład zespołu, sposoby profilowania akcji.

2.3.1 Konfiguracja grupowa współpracujących urządzeń mobilnych;

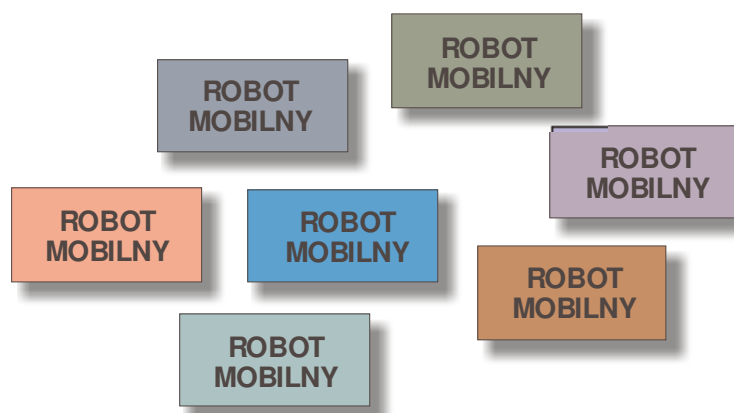
Podstawowym elementem, jaki należy brać pod uwagę przy opisie interesujących rozwiązań z zakresu pracy zespołowej jest określenie czy system jest **scentralizowany** czy też **zdecentralizowany**.



Rys 1. Przykłady architektury systemów składających się z wielu jednostek mobilnych.

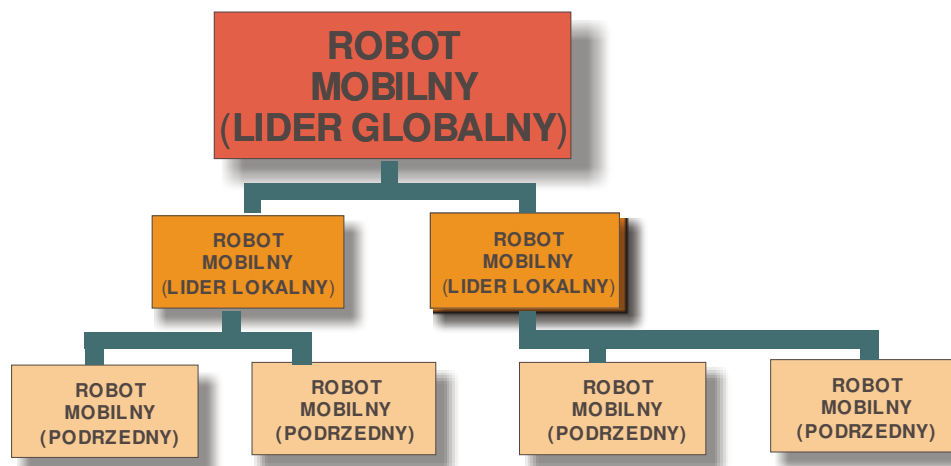
Architektura opierająca się na scentralizowanym układzie jednostek charakteryzuje się jedną jednostką pełniącą funkcję kontroli w zespole. Natomiast układ zdecentralizowany przedstawia sobą układ, gdzie nie ma powyższego, tylko jednego, urządzenia kontrolującego pełniącego funkcję nadrzędną. Wyróżnić tu z kolei można podział na dwie podgrupy – podejścia do struktury dotyczącej konfiguracji zespołów:

Architekturę rozproszoną, gdzie wszyscy członkowie danego zespołu mają równe uprawnienia dotyczące kontroli czy też sterowania.



Rys 2. Przykład systemu przedstawiającego typ architektury rozproszonej.

Architekturę hierarchiczną, którą opisać można jako konstrukcję zespołu opartą na lokalnych centrach, stanowiących przez jednostki nadrzędne w podgrupach, w stosunku do innych urządzeń wchodzących w skład jednego zespołu.



Rys 3. Przykład systemu przedstawiającego typ architektury hierarchicznej.

Należy tu podkreślić, że nie ma tu określonej budowy urządzeń i wykonywanych przez nie działań. Określone są jedynie niejako „uprawnienia” dotyczące pełnionej w zespole roli – czy jednostka jest tylko elementem wykonującym polecenia innych, czy też sama może decydować odnośnie podejmowanych działań przez siebie i inne (jej podległe lub też nie) urządzenia.

Systemy zdecentralizowane są często określane jako bardziej wydajne i efektywniejsze w porównaniu do systemów scentralizowanych⁸. Można wymienić tu przede wszystkim:

- większą odpornością na błędy funkcjonowania;
- wynikające z charakteru architektury wykorzystanie podziału pracy;
- niezawodność całego systemu;
- skalowalność;

Jakkolwiek wyżej przedstawiona architektura jest obiektem częstych badań i ma wiele zalet, to jednak w praktyce wykorzystywane jest niejako połączenie obu

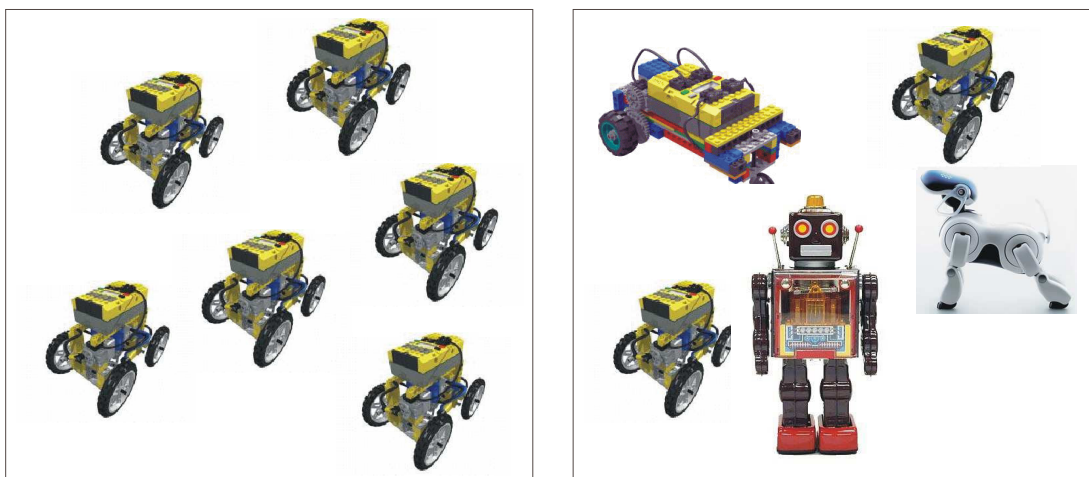
stylów architektury – zbliżone w swej funkcjonalności do **architektury hierarchicznej**.

2.3.2 Cechy indywidualne i charakterystyka zespołu

Każda grupa, której członkowie współpracują ze sobą może zawierać różne typy urządzeń względem siebie zróżnicowanych. Określenie typu i charakteru tych jednostek jest jednym z fundamentalnych elementów, obok postaci architektonicznej zespołu, w celu zobrazowania i skonstruowania grupy współdziałających robotów.

Wyróżnić można dwa typy zespołów urządzeń współdziałających ze sobą:

- **homogeniczne;**
- **heterogeniczne;**



Rys 4. Zobrazowanie postaci grup robotów i typu jednostek wchodzących w ich skład.

O homogenicznych grupach można mówić wtedy, gdy możliwości wykonywania zadań i budowa robotów są identyczne. Jeżeli jest inaczej i mamy do czynienia z różnymi typami urządzeń, to wtedy określamy dany zespół jako heterogeniczny. Z uwagi na to, iż istnieją spore trudności przy zróżnicowaniu jednostek wchodzących w skład grupy, o wiele lepszym rozwiązaniem jest rozpatrywanie i eksploracja dziedziny robotów homogenicznych. Jako przykład powyższego stwierdzenia można wyobrazić sobie, system, w którym każdy robot ma inne „umiejętności”, a podczas wykonywania określonego zadania potrzebujemy wykorzystać (w różnych punktach

przestrzeni) dwa urządzenia o identycznych lecz unikalnych właściwościach – nie możliwe będzie wtedy dokończenie pracy ze względu na posiadanie zbyt małych zasobów. Innym z kolei argumentem za wykorzystaniem urządzeń z rodziny robotów homogenicznych jest to, że o sile zespołu i jego możliwościach decydowały będą nie pojedyncze jednostki, lecz cały kolektyw – nie będzie tu istniało pojęcie najsilniejszego i najsłabszego ogniwa.

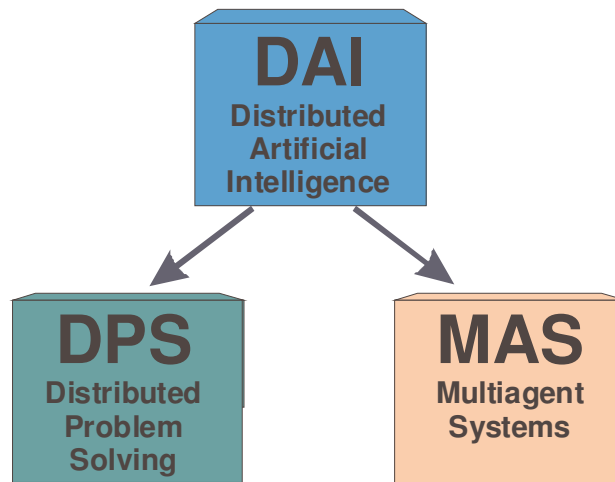
2.3.3 Profilowanie akcji współdziałających jednostek

Kolejnym elementem mogącym być przedstawionym jako charakterystyczna część struktur grupowych jest modelowanie akcji przez członków zespołu. Jeżeli jednostki współdziałające mogłyby „przewidywać” możliwe ruchy swoich „towarzyszy” to wtedy komunikacja pomiędzy nimi byłaby w znacznym stopniu uproszczona. Jeden obiekt posiadałby w takim przypadku reprezentację profilu innej jednostki (szczególnie przy wykorzystaniu architektury heterogenicznej) i potrafiłby oddziaływać w pewien sposób na inne pracujące obiekty.

2.4 Rozproszona sztuczna inteligencja - DAI

Rozproszona sztuczna inteligencja (ang. DAI – Distributed Artificial Intelligence) jest dziedziną nieco konkurującą z inteligencją grupową określaną w języku angielskim jako Swarm Intelligence. DAI składa się z rozproszonych jednostek inteligentnych, a z kolei systemy typu SI to grupy bazujące na prostych jednostkach, ze słabo rozwiniętymi umiejętnościami poznawczo-behawioralnymi. Warto jednak zwrócić uwagę na ten rodzaj współdziałania w grupie, ponieważ istnieją tu pewne zależności, które można wykorzystać w systemach Swarm Intelligence.

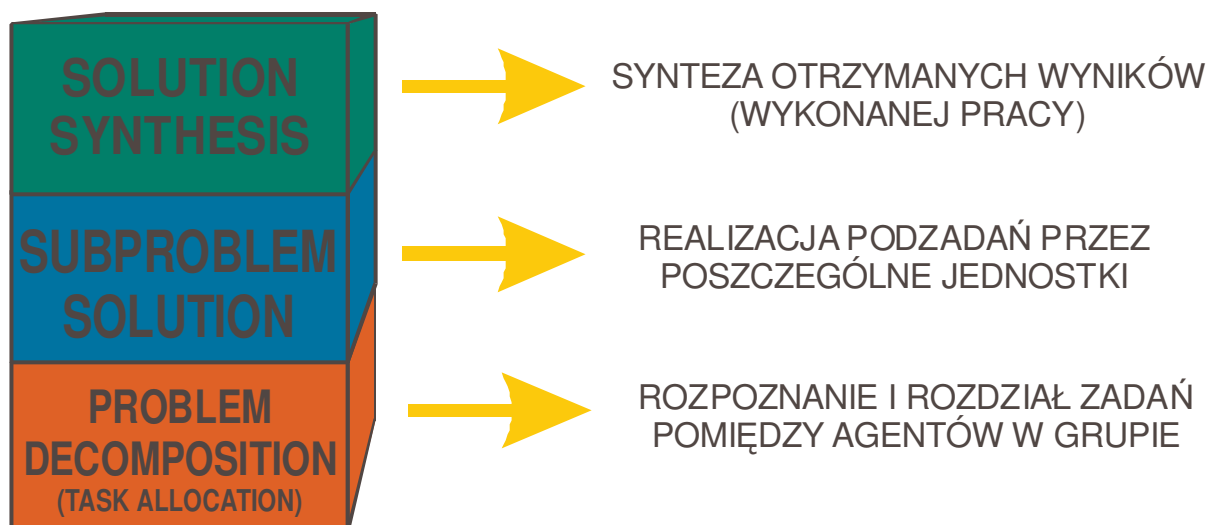
Zwracając uwagę na charakterystyczne cechy jakimi odznaczają się systemy DAI można tu wyróżnić tu dwa wyraziste podejścia przedstawione na diagramie poniżej:



Rys 5. Podział aplikacji wykorzystujących elementy rozproszonej sztucznej inteligencji.

DPS – Distributed Problem Solving – Rozproszone Rozwiązywanie Problemów (Rozproszona Realizacja Zadań).

W tym podejściu do tematu komunikacji w systemach sztucznej inteligencji głównym elementem jest sposób wykonywania zadań (ang. problem solving) przy użyciu wielu urządzeń. Jednostki wchodzące w skład grupy mogą współpracować poprzez niezależne wykonywanie podzadań (ang. subproblems & task-sharing) oraz przez komunikowanie się ze „współtowarzyszami” w celu wymiany informacji i przetworzonych danych. W dziedzinie DPS można wyróżnić kilka istotnych elementów, które składają się na kolejne fazy pracy systemu:



Rys 6. Fazy przetwarzania informacji w aplikacjach rozproszonej sztucznej inteligencji.

Najistotniejszym i zarazem pierwszym elementem w bloku podziału pracy jest przydzielenie zadań poszczególnym agentom. Pojawia się tu problem, gdy musimy wybrać jednostki najlepiej przystosowane do wykonania określonego zadania, aby praca została przeprowadzona w optymalny sposób. Następną częścią jest element wykonawczy, w którym następuje realizacja zadań, a na samym końcu przeprowadzana jest synteza działań i wyników.

2.5 Komunikacja, interakcja, współpraca

Komunikacja i współpraca są ściśle ze sobą powiązane. Komunikacja to nieodłączna część interakcji w środowisku współdziałających robotów, a z kolei współpraca opiera się w ogromnej mierze na interakcji pomiędzy agentami. Istnieje wiele sposobów, w jaki porozumiewanie się pomiędzy jednostkami może być reprezentowane. Należy tu wziąć pod uwagę kilka istotnych elementów charakterystycznych, których obecność w danym środowisku może przedstawiać kluczową rolę w sposobie przekazywania informacji pomiędzy członkami grupy:

- Dystans interakcji;
- Stopień równoczesności interakcji;
- Jawność przekazu;
- Wyrazistość interpretacji (ang. sophistication of interpretation)⁹;

2.5.1 Właściwości komunikacji

Dystans interakcji jest to dystans pomiędzy agentami w trakcie odbywającego się procesu komunikacji (lub przeprowadzenia jej próby). Mogą zostać tu wyszczególnione takie elementy jak:

- połączenie fizyczne;
- kontakt wizyjny;
- zakres słyszalny;
- kontakt dalekiego zasięgu;

Stopień równoczesności interakcji – jest to czas pomiędzy wysłaniem sygnału przez jednostkę pragnącą przekazać informację otoczeniu, a odebraniem jej przez innego agenta (lub wielu agentów). Może to być komunikacja natychmiastowa (na przykład przy połączeniu bezpośrednim – fizycznym) lub też z wydłużonym czasem reakcji-odbioru (przy pozostawieniu informacji pod pewną postacią, aż do czasu odebrania jej przez innych agentów)

Jawność przekazu – przekazywanie informacji może odbywać się w zamierzony sposób, gdzie przekazujący określoną informację robot podejmuje pewne działania służące do przesłania wiadomości (zaprogramowane, wyuczone, zaprojektowane). Może to być również efekt „uboczny” – następuje tu przekazanie informacji nawet wtedy, gdy nadawca nie ma bezpośredniej intencji na uczynienia tego.

Wyrazistość interpretacji (ang. sophistication of interpretation) – istotna jest również skala poziomu zrozumienia pomiędzy członkami grupy. Gdy jeden nadawca wysyła informację, inne mogą ją różnie zinterpretować lub też w ogóle zignorować – musi tu zostać zachowana swego rodzaju klarowność w przekazywaniu informacji, aby zrozumienie pomiędzy członkami grupy było możliwie jak największe.

2.5.2 Podstawowe kategorie komunikacji

Komunikacja niejawna (ang. implicit communication) – w tej kategorii komunikacji pomiędzy robotami mobilnymi dochodzi do przekazywania informacji przez środowisko. Jednostki pozostawiają w miejscu przebywania swoistą wiadomość, która może być odbierana przez innych agentów. W podobny sposób zachowują się w przyrodzie mrówki, które wydzielają i pozostawiają feromon na drodze swojej wędrówki – inne owady z tej samej kolonii mogą podążać jej śladem w celu odnalezienia ważnych surowców (np. pożywienia). Jako przykład można wziąć pod uwagę roboty ścinające trawę na łące. Pozostawiają one w środowisku informację o tym czy dany obszar został już przystrzyżony, czy też dopiero trzeba to zrobić. Urządzenia z założenia mogą wykryć czy trawa została przycięta czy też nie – jeżeli wysokość trawy jest odpowiednia to znaczy, że któraś z jednostek już pracowała w tym obsza-

rze i nie trzeba przeprowadzać operacji ścinania – jeżeli wysokość łodyg jest zbyt duża wtedy robot przystępuje do pracy.

Tak samo może odbywać się komunikacja w środowisku robotów z rodziny heterogenicznych, gdzie wszechstronny „robot przewodnik” potrafi odnaleźć odpowiednią drogę w nieznanym środowisku i pozostawia na podłożu zakodowaną informację (charakterystyczną woń, kolor, przedmioty). Mniej skomplikowane, choć wszechstronne jednostki będą podążały jej śladem odczytując właśnie te informacje.

- Korzyści komunikacji niejawnej
 - Prostota – często elementy komunikacji niejawnej pojawiają się zgoła automatycznie;
 - Niezawodność;
 - Brak zależności od jakiegokolwiek technicznego medium transmisji;
- Ograniczenia i niedogodności komunikacji niejawnej
 - Komunikacja niejawna jest ograniczona do możliwości czujników robota i sposobu zbierania przez nie informacji ze środowiska;
 - Ilość przekazywanych danych jest ograniczona do zakresu możliwości pozostawienia ich w środowisku i czasu ich przetrwania;
 - Informacja o odnalezieniu interesującego obiektu nie może być od razu przekazana innym jednostkom;

Komunikacja poprzez stan (ang. state communication) – w tej kategorii przekazywanie informacji odbywa się przez obserwację zachowań innych jednostek. Na podstawie tak zdobytych danych zostaje podjęta decyzja o przeprowadzeniu określonej akcji. Jako przykład może posłużyć poszukiwanie źródeł światła. Jeśli grupa robotów znajduje się w danym obszarze i nie dostrzegane są żadne źródła światła to z założenia poruszają się one w przypadkowy i nie mający zasadniczego celu sposób. Wszystkie jednostki mogą obserwować siebie nawzajem i poruszają się w ten sam sposób, aż nie nastąpi określone wydarzenie. Jeżeli jeden z robotów dostrzeże źródło światła i zacznie podążać w jego stronę, inni agenci nie widząc światła a jedynie zdecydowany ruch jednego z nich w pewnym kierunku, założą że odnalazł on coś

czego „poszukują” - zaczął poruszać się w tę samą stronę. Jest to komunikacją związaną z obserwacją stanu innych jednostek – tzw. komunikacja poprzez stan.

Jednak istnieje tu kilka bardzo ważnych kwestii:

- Roboty muszą posiadać umiejętność rozróżniania innych robotów od otaczającego je środowiska;
- Roboty muszą rozróżniać celowo podjęte akcje od „zwykłego” zachowania;
- Jednostki muszą dokładnie i precyzyjnie interpretować zachowanie innych agentów w grupie;

Taki rodzaj komunikacji jest używany szczególnie w aplikacjach gier zespołowych, gdzie wykorzystanie informacji o położeniu, sposobie poruszania się i stanie innych urządzeń może mieć kluczową rolę w dla wyniku rozgrywki.

- Korzyści komunikacji poprzez stan:
 - Niezawodność – komunikacja bez medium, jedynie poprzez obserwację.
- Ograniczenia i niedogodności komunikacji poprzez stan:
 - Duża trudność w przekazaniu informacji o znalezieniu interesującego surowca.
 - Niemożność przekazania informacji innej niż procedura (z ang.) „follow me” – podążaj za mną.

Komunikacja poprzez stan posiada niedoskonałości, które mogą spowodować do niejednoznacznej i mylnej interpretacji zachowania innych jednostek w grupie. Jak pokazują doświadczenia laboratoryjne odróżnienie celowej akcji podjętej przez robota od przypadkowego poruszania się jest bardzo trudne.

Komunikacja bezpośrednia – **zamierzona** – (ang. explicit communication) - jest to rodzaj komunikacji, w której przekazywanie informacji jest jawne i celowe. Robot pragnący wymienić dane z innym urządzeniem rozpoczyna procedurę komunikacji i specjalnie podejmuje kroki ku temu by przekazać jakąś informację. Nie jest to „pro-

dukt uboczny” zachowania się robota w środowisku, czy też bierne oczekiwanie na akcje i działania innych jednostek w grupie. Często jest tak, iż w tym rodzaju komunikacji potrzebne są dodatkowe zasoby sprzętowe w postaci modułu komunikacyjnego. Mogą to być:

- IEEE 802.11 – ustandaryzowana bezprzewodowa komunikacja radiowa;
- IrDA – transmisja danych w podczerwieni;
- Bluetooth – transmisja z wykorzystaniem fal radiowych ze szczególną strukturą powiązań w sieci;

W tej kategorii komunikacji bardzo ważnym elementem jest topologia w grupie pomiędzy agentami. Może to być struktura hierarchiczna (drzewiasta) lub też rozproszona.

Struktura hierarchiczna – w skład grupy wchodzi podstawowe jednostki i jej liderzy. Komunikacja odbywa się tylko pomiędzy „szeregowymi” agentami i ich liderem, który pełni rolę nadrzędną.

Struktura mieszana – każdy agent z założenia może komunikować się z dowolnym innym agentem i wymieniać z nim informacje.

3. Eksploracja i konstrukcja map środowiskowych

3.1 Samo-lokalizacja jednostki mobilnej

Wyzwanie dla użycia robotów mobilnych w celu nawigacji w nieznanym środowisku podejmowana była wielokrotnie w różnego rodzaju projektach i aplikacjach^{15,16}. Dla przykładu: roboty mogą pomagać asystować pacjentom w szpitalach czy też osobom w muzeach, badać obszary niebezpieczne dla ludzi, wykonywać zadania w głębinach morskich, obserwować i ochraniać określony obszar terenu. Lokalizacja urządzenia w każdym z tych środowisk jest niezmiernie ważna dla poprawnego funkcjonowania robota mobilnego, choć jest ona bardzo często ogromną przeszkodą do pokonania. Wydaje się więc być rzeczą naturalną, iż „przeskoczenie” tego problemu usprawniłoby działanie każdej projektowanej aplikacji.

Proces lokalizacji robota w środowisku pozwala na określenie aktualnego położenia względem otaczającego (ograniczonego) obszaru. Istnieją dwa rodzaje sposobów określania pozycji urządzenia mobilnego:

a) względny (ang. relative)

Odometria (z ang. odometry, dead reckoning) jest to metoda polegająca jedynie na względnym oszacowaniu położenia.

b) bezwzględny (ang. absolute)

Realizowany jest przy indywidualnym wykorzystaniu takich elementów jak: czujniki, markery (znaki sygnalizacyjne), dopasowanie do mapy.

Dodatkowo istnieją metody oszacowania pozycji przy użyciu filtru Kalmana i tzw. (z ang.) „feasible poses” – możliwe do wykonania ruchy. Wszystkie powyższe są wykorzystywane zamiennie w różnych aplikacjach i często używane jako aranżacja kombinacyjna systemu lokalizacji.

3.1.1 Pozycjonowanie względne i odometria – identyfikacja błędów.

Główną wadą metod odometrycznych jest nieograniczona akumulacja błędów. Jednakże jeśli zostanie ulepszony ten sposób szacowania pozycji robota w znacznym stopniu można uprościć instalacje innych czujników zewnętrznych.

Najbardziej popularnym i powszechnym sposobem lokalizacji urządzenia w terenie jest użycie metody nazywanej odometrią. Dzieje się to dlatego, iż ilość komponentów potrzebnych do zrealizowania danej aplikacji jest z reguły niewielka i mało kosztowna. Jako przykład może posłużyć najprostsza aplikacja robota, który wykorzystuje dwa napędzane różnicowo koła. Na każdym z nich instalowany jest dekodery służący jako licznik kąta obrotu. Nakładki te pozwalają zliczać kąt „przebyty” przez każde koło z osobna poprzez rejestrację otrzymanych impulsów, co jest wystarczającą informacją dla tego typu pomiaru. Otrzymane dane wykorzystywane są z kolei do wyliczenia dystansu, jaki przebyły koła urządzenia. Pozycja robota jest więc szacowana na podstawie informacji otrzymanych z poprzednich wskazań i wyliczeń – do poprzednich wartości określających położenie dodawane są po prostu nowe dane. Aby uniknąć dużych błędów w obliczeniach dane muszą być przetwarzane w sposób ciągły i szybki. Duża łatwość uzyskiwania potrzebnych do lokalizacji danych jest olbrzymią zaletą tego sposobu estymacji położenia urządzenia – nie potrzeba tu dodatkowego, skomplikowanego osprzętu, a wynik jest otrzymywany natychmiast bez kosztownych obliczeń.

W projektowanych aplikacjach wykorzystywane są również żyroskopy do oszacowania kąta położenia. Najczęściej używanymi czujnikami są żyroskopy mechaniczne, optyczne i geomagnetyczne. Oferują one bardziej dokładny pomiar kąta położenia niż techniki używane w odometrii. Dodatkowo, kiedy zalety żyroskopu są łączone z pomiarami otrzymanymi z kół i wcześniejszych, „historycznych” wskazań, uzyskuje się wyższą dokładność estymacji położenia. Jednakże te typy czujników oprócz niewątpliwych zalet mają również swoje wady. Mianowicie istnieje tu zjawisko dryftu, które ma znaczący wpływ na odzwierciedlenie rzeczywistych zmian w położeniu do zmian rejestrowanych przez układ pomiarowy urządzenia.

Dla żyroskopu mechanicznego dokładność i sama wielkość dryftu pomiaru jest zależna od jakości wykonania, a co z tym idzie również kosztu urządzenia pomiarowego.

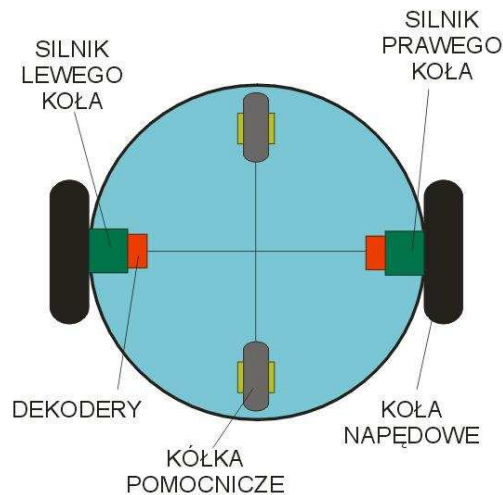
Zjawisko dryftu istnieje również w żyroskopach optycznych, lecz sama jego wielkość jest stosunkowo niewielka w odniesieniu do urządzeń mechanicznych¹⁰. Należy tu jednak nadmienić, iż żyroskopy optyczne posiadają małą rozdzielczość przy określaniu kąta obrotu, co należy uwzględniać przy szacowaniu błędów systematycznych występujących przy pomiarach.

Z kolei urządzenia geomagnetyczne, które są używane najczęściej w różnego rodzaju aplikacjach, mogą wprowadzać błędy wskazań przy źródłach pola elektromagnetycznego i blisko obiektów przewodzących. Uniemożliwia to ich prawidłową pracę i wykorzystanie wewnątrz małych obiektów.

Borenstein w swojej pracy [10] sugeruje połączenie różnych typów sensorów, co jest najbardziej obiecującym pomysłem na uzyskanie zadowalających wyników. Gdy na przykład połączy się w jednej aplikacji czujniki, które odpowiadają za wykrycie przeszkód przed robotem-pojazdem (np. ultradźwiękowe) razem z dekoderni umieszczonymi w kołach wtedy dostarczone informacje mogą być analizowane w szerszym zakresie.

Jednak zbyt duża ilość informacji uzyskanych z różnych typów czujników może spowodować, iż oszacowanie dokładnej pozycji stanie się niemożliwe. Wskazania mogą bowiem różnić się od siebie na tyle, że proces lokalizacji będzie przerwany wskutek zbyt dużej akumulacji błędów – z tego względu liczba czujników z uwagi na nadmiarowość musi zostać ograniczona do niezbędnego minimum.

Jako przykład modelu urządzenia służącego do oszacowania pozycji może posłużyć robot napędzany różnicowo – (z ang.) „DIFFERENTIAL DRIVE ROBOT”.



Rys. 7 Przykład poglądu robota mobilnego bazującego na różnicowym napędzie kół.

Założenia i zależności przyjmowane w obliczeniach:

$$E_w = \frac{\pi \cdot D}{n \cdot R_d}$$

gdzie:

E_w – współczynnik określający przeliczenie impulsów z dekodera obrotów na przesunięcie liniowe;

D – średnica koła napędowego;

n – przekładnia pomiędzy silnikiem, a kołem napędowym;

R_d – rozdzielczość dekodera (określana jako liczba impulsów przypadających na jeden obrót koła napędowego)

Na obliczenia składają się rachunki mające na celu określenie dystansu przebytego przez każde z kół z osobna.

$$\Delta L_{L,i} = E_w \cdot N_{L,i} \qquad \Delta L_{R,i} = E_w \cdot N_{R,i}$$

gdzie:

$N_{R,i}$, $N_{L,i}$ – ilość impulsów zliczonych przez dekodery.

Obliczenie przesunięcia przez centralny punkt robota wyznaczyć można z poniższego wzoru:

$$\Delta L_i = \frac{\Delta L_{L,i} + \Delta L_{R,i}}{2}$$

Zmianę kierunku ustawienia czoła robota można określić przy pomocy zależności:

$$\Delta K_i = \frac{\Delta L_{L,i} - \Delta L_{R,i}}{z}$$

gdzie:

z – odległość pomiędzy kołami napędowymi robota (powinna być mierzona od punktu styczności jednego koła z podłożem do punktu styczności drugiego koła)

Pozycja punktu centralnego robota określana jest zatem jako poniższa zależność i może być wykorzystywana dla określenia położenia całego robota w przestrzeni dwuwymiarowej przy założeniu postrzegania go jako punkt.

$$x_i = x_i + \Delta L_i \cdot \cos(K)$$

$$y_i = y_i + \Delta L_i \cdot \sin(K)$$

Określenie lokalizacji robota jest więc stosunkowo łatwe z wykorzystaniem powyższych niezbyt skomplikowanych równań. W takiej aplikacji stosuje się elementy, które nie są bardzo kosztowne i jednocześnie nie sprawiają większych problemów przy eksploatacji (jak na przykład dekodery). Jednak przeprowadzanie badań i korzystanie z takiego rozwiązania ma również pewne ograniczenia. Jako jeden z dobitnych przykładów może posłużyć „poślizg” robota po powierzchni, po której się porusza (na przykład, gdy teren jest pochyły i przyczepność kół jest niska – zsuwanie się pojazdu - lub na płaskim podłożu, lecz nawierzchni śliskiej – plama oleju, lód). Mogą być wtedy zarejestrowane obroty kół, które nie odpowiadają rzeczywistości, tj. nie są proporcjonalne do faktycznego przesunięcia liniowego pojazdu w przestrzeni. Może również wystąpić sytuacja odwrotna – robot może się przemieścić, lecz zarejestrowanie tego faktu przez dekodery zainstalowane przy kołach nie będzie miało miejsca.

Jest jednak jeszcze kilka innych elementów mający znaczący wpływ na wykorzystanie (lub też nie) metod odometrycznych w poruszaniu się robota. Chodzi tu przede wszystkim o niedokładności w odwzorowaniu rzeczywistego przemieszczenia się robota w przestrzeni do rejestracji tego zdarzenia przez czujniki – dekodery. Można tu rozróżnić dwa rodzaje błędów, które mogą wystąpić w aplikacjach bazujących na metodach odometrycznych:

- a) Błędy systematyczne;
- b) Błędy niesystematyczne;

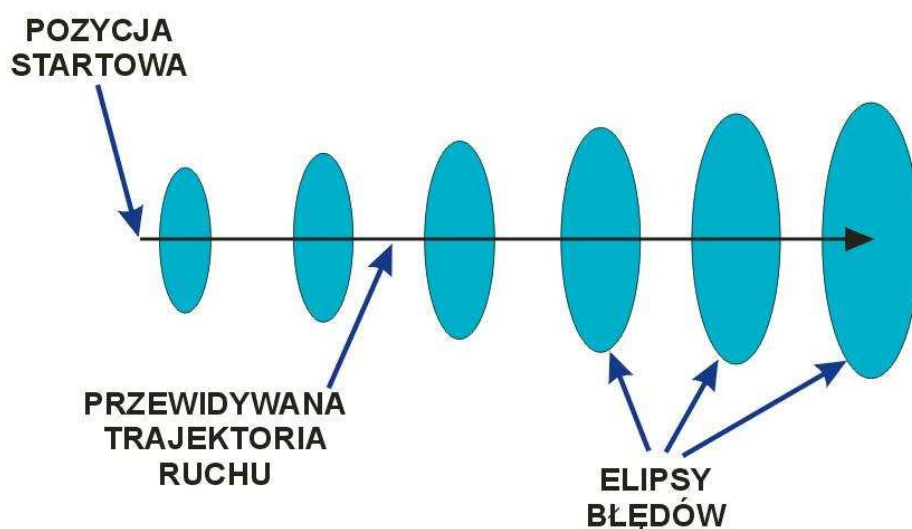
Błędy **systematyczne** mogące wystąpić podczas ruchu robota w terenie to przede wszystkim:

- Niedokładność wykonania kół robota - nierówne średnice;
- Brak stałej wartości odległości pomiędzy punktami styczności każdego z kół z podłożem;
- Złe pozycjonowanie kół - niepewność stałego kontaktu z podłożem;
- Ograniczona rozdzielczość dekodatorów;

Błędy **niesystematyczne** to:

- Przemieszczanie się po nierównym terenie;
- Poślizgi kół na powierzchni z uwagi na:
 - śliskie podłoże;
 - zbyt szybkie, nie rejestrowane zmiany położenia (obroty);
 - czynniki zewnętrzne;
 - „zgubienie” kontaktu z podłożem koła napędowego;

Błędy systematyczne sprawiają wiele problemów z uwagi na ich akumulację w odniesieniu do czasu ruchu pojazdu. Istotnym czynnikiem jest określenie zakresu „niepewności” co do aktualnej pozycji robota (ang. position uncertainty). W swoim podejściu Borenstein ([10]) stosuje elipsy opisujące jak może kształtować się rozkład błędów pozycjonowania w trakcie pokonywania przez pojazd określonej drogi.



Rys. 8 Elipsy błędów – niedokładności pozycjonowania pojazdu.

Elipsy rosną wraz ze zwiększającym się przebyтым przez robota dystansem, aż do momentu, gdy nastąpi oszacowanie bezwzględne pozycji tegoż pojazdu - następuje wtedy „wyzerowanie” elipsy błędu.

Konkluzja:

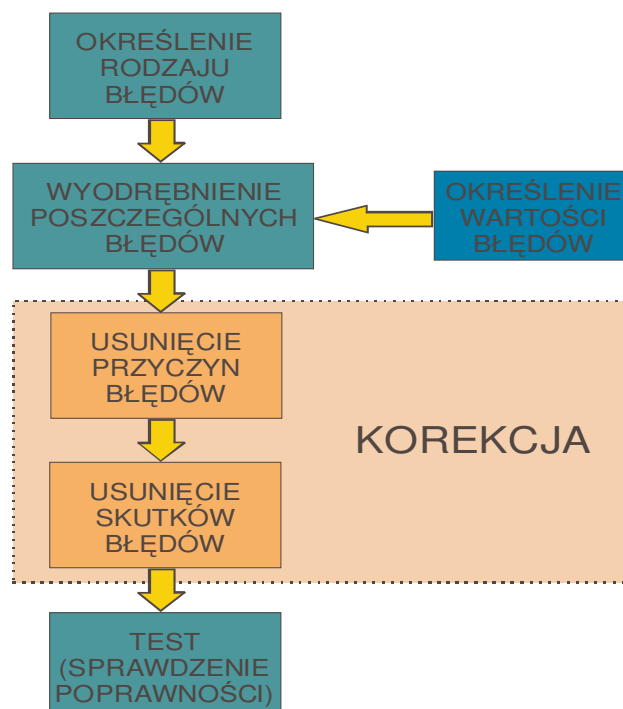
Błędom systematycznym można zaradzić, ponieważ do ich częściowego wyeliminowania posłużyć może odpowiednio dokładnie wykonany i precyzyjny osprzęt. W pomieszczeniach zamkniętych, przy stosunkowo dobrym stanie podłoża, błędy systematyczne mają największy wpływ na dokładną lokalizację robota, jednak przy nieregularnych terenach zewnętrznych zauważyć można dominację błędów niesystematycznych.

3.1.1.1 Pomiary i eliminacja błędów

Metody odometryczne ukazują szczególny sposób na rozwiązanie problemu lokalizacji robota w terenie. Jednak, jak wiadomo, obarczona jest ona błędami, które należy wyeliminować, aby bez przeszkód móc ją stosować.

Niezależnie od sposobu pomiaru wielkości wejściowych należy zwracać baczną uwagę na identyfikację zakłóceń i ewentualne wystąpienie awarii (powodującej nieprawidłowe wskazania) w stosowanym osprzęcie. Poniższy zaproponowany

schemat ukazuje, w jaki ogólny sposób należy odnosić się do istniejących problemów z określaniem pozycji robota w terenie.



Rys. 9 Przykład ogólnego algorytmu identyfikacji i korekcji błędów pojawiających się w aplikacji.

Pierwszym krokiem ku usunięciu niepożądanych zakłóceń w pracy jest określenie rodzaju występujących błędów. Zwracać powinno się tu uwagę na możliwe elementy, mogące mieć znaczny wpływ na poprawność funkcjonowania aplikacji. Trzeba się zastanowić nad typem i charakterem zakłóceń z jakimi możemy mieć do czynienia. Następnie należy wyodrębnić te rodzaje błędów, które są najbardziej znaczące i określić wartości jakie sobą reprezentują. Kolejnym krokiem jest korekcja wskazań, co osiąga się poprzez eliminację występujących zakłóceń. Ostatnim etapem jest sprawdzenie poprawności przeprowadzonej korekcji.

Jedną z istotnych trudności do pokonania w środowisku robotów mobilnych jest ilościowy pomiar błędów. Brak jasno zdefiniowanych reguł i procedur oszacowania błędów jest wyraźnym mankamentem w interesującej nas dziedzinie. Borenstein oraz Feng¹⁰ opracowali pewne metody, które stosowane na uproszczonym modelu robota mobilnego dają zadowalające wyniki w określaniu i eliminacji najczęściej wy-

stępujących zakłóceń w lokalizacji pojazdów. Wyszczególnili oni dwa rodzaje błędów uważanych za najbardziej znaczące i właśnie dla nich opracowali algorytm mogący pomóc w realizacjach aplikacyjnych opracowań dotyczących robotów mobilnych.

Jedną z tych wielkości jest błąd odpowiadający nierównym wielkościom średnic kół napędowych robota, które pełnią jednocześnie funkcję dla pomiaru przebytej drogi (odometria):

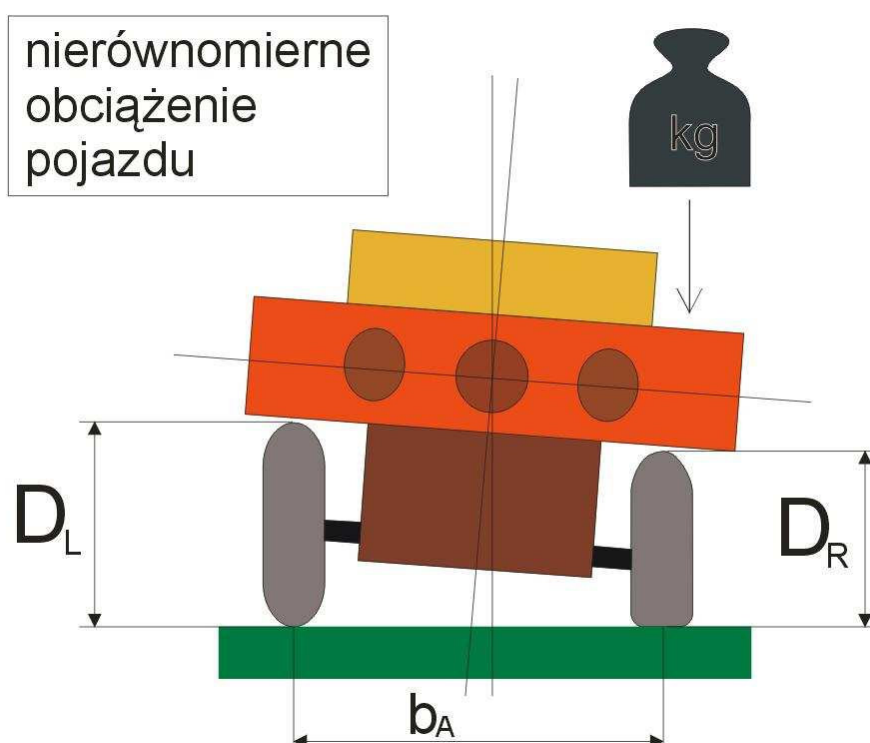
$$E_d = \frac{D_R}{D_L}$$

D_R oraz D_L są odpowiednio średnicami prawego i lewego koła pojazdu. Należy zwrócić uwagę, że np. przy nieodpowiednim (nierównomiernym) obciążeniu pojazdu lub złym napompowaniu kół może wystąpić deformacja jednego z nich.

Drugą wielkością jest błąd wynikający z różnicy pomiędzy wartościami odległości pomiędzy punktami styków kół pojazdu z powierzchnią, po której się poruszają. Wartość obliczeniowa b_N może się bowiem różnić od wielkości rzeczywistej b_A , co w znacznym stopniu będzie wpływało na wyniki obliczeń bazujących na wielkościach fizyczno-gabarytowych elementów pojazdu.

$$E_b = \frac{b_A}{b_N}$$

b_N oraz b_A są to wielkości odpowiadające wartościom odpowiednio: Nominalnej (obliczeniowej) oraz Aktualnej (rzeczywistej).



Rys. 9 Nierównomierne obciążenie pojazdu i deformacja kół – zmiana średnicy obliczeniowej.

3.1.2.2 Oszacowanie błędów systematycznych

Charakter błędów powstałych przy oszacowaniu pozycji i wykonaniu ruchu dobrze obrazuje metoda nazywana jako UMBmark a opracowana na uniwersytecie Michigan¹⁰.

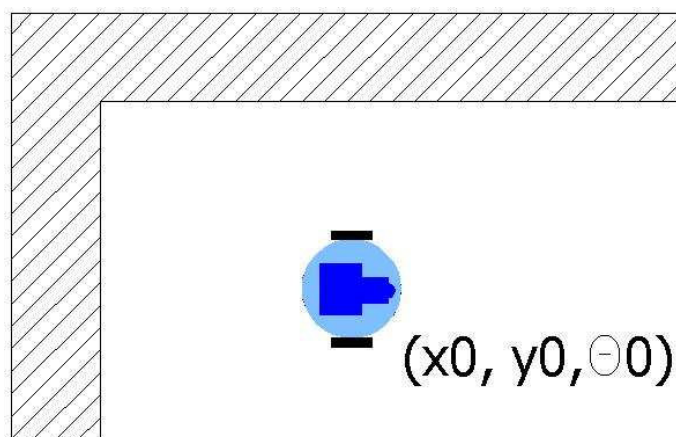
Jest to sposób na określenie wielkości błędów, jakie mogą pojawić się podczas ruchu robota w terenie i oszacować swego rodzaju tolerancję, jaka odpowiadać może nie skorygowanemu wynikowi pomiarów wykonanych przez robota mobilnego.

Metoda ta składa się z pewnych etapów – jednokierunkowy test kwadratu i dwukierunkowy test kwadratu - które zostaną zaprezentowane poniżej.

Przebieg procesu określania błędów przebiega następująco:

Robot rozpoczyna jazdę w pozycji x_0, y_0 . Sam „start” pojazdu powinien rozpoczynać się w niewielkiej odległości od „punktu orientacyjnego” względem, którego

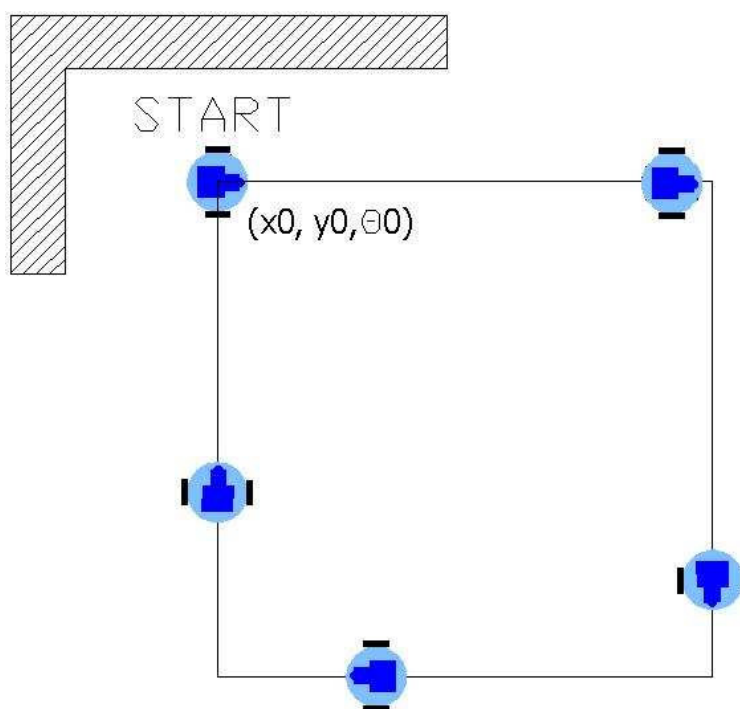
mógłby z dużą dokładnością określić swoje położenie. Jako przykład mogą posłużyć dwie prostopadłe ustawione do siebie ściany, które będą dobrym punktem orientacyjnym dla takiej aplikacji.



Rys. 10 Pozycjonowanie pojazdu względem otaczającego środowiska – pozycja startowa.

Test polega na przebyciu przez robota dystansu wzdłuż kwadratu o określonej długości boków. Urządzenie mobilne będzie więc rozpoczynało i kończyło w tym samym miejscu swoją podróż. W idealnych warunkach i przy bardzo dokładnych elementach wykonawczych urządzenia taki scenariusz byłby z pewnością możliwy do wykonania, lecz rzeczywistość skazuje nas na sytuację, które nie będą odzwierciedlały dokładnie zaplanowanej pracy. Tak więc pojawią się błędy przy pozycjonowaniu-lokalizacji robota względem środowiska.

Ostateczna pozycja pojazdu będzie więc różna od tej, w której rozpoczął się ruch pojazdu.



Rys. 11 Teoretyczna ścieżka ruchu robota w „łęczce kwadratu”.

Następnym krokiem jest zmierzenie pozycji robota na podstawie wskazań czujników innych niż te używane w pomiarach odometrycznych. Powinno się to odbyć możliwie dokładnie i w odniesieniu do miejsca, w którym umieszczony był punkt START pojazdu (np. względem prostopadłych ścian).

Otrzymane wyniki są porównywane ze sobą i tym samym uzyskuje się błędy pozycjonowania:

$$\Delta x = x_A - x_C$$

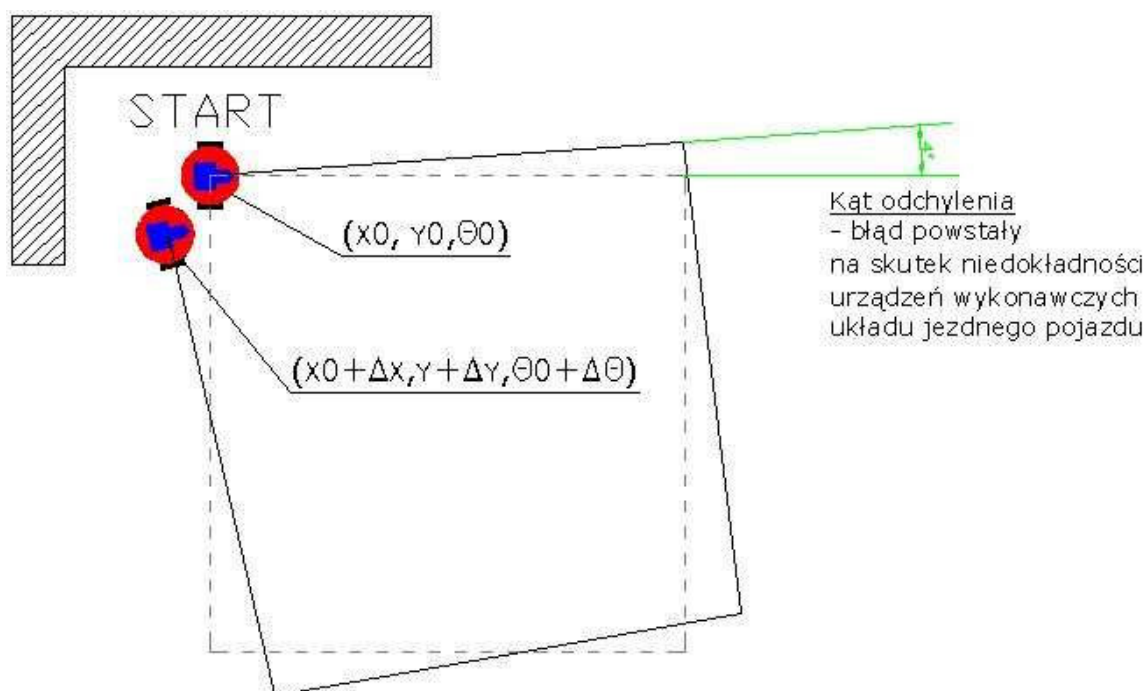
$$\Delta y = y_A - y_C$$

$$\Delta \Theta = \Theta_A - \Theta_C$$

x_A, y_A, Θ_A - wielkości bezwzględnego pomiaru

x_C, y_C, Θ_C - wielkości obliczone

$\Delta x, \Delta y, \Delta \Theta$ - błąd wynikający z niedokładności urządzeń



Rys. 12 Rzeczywista ścieżka ruchu robota.

W teście tym widać wyraźnie, iż niedoskonałość urządzeń jest nazbyt widoczna i pojawiają się zakłócenia wprowadzane przez elementy wykonawcze (bądź też środowisko zewnętrzne).

Powracając do koncepcji, że dwa z podstawowych błędów odgrywają dominującą rolę można stwierdzić, co następuje:

Błąd pozycjonowania spowodowany jest przez:

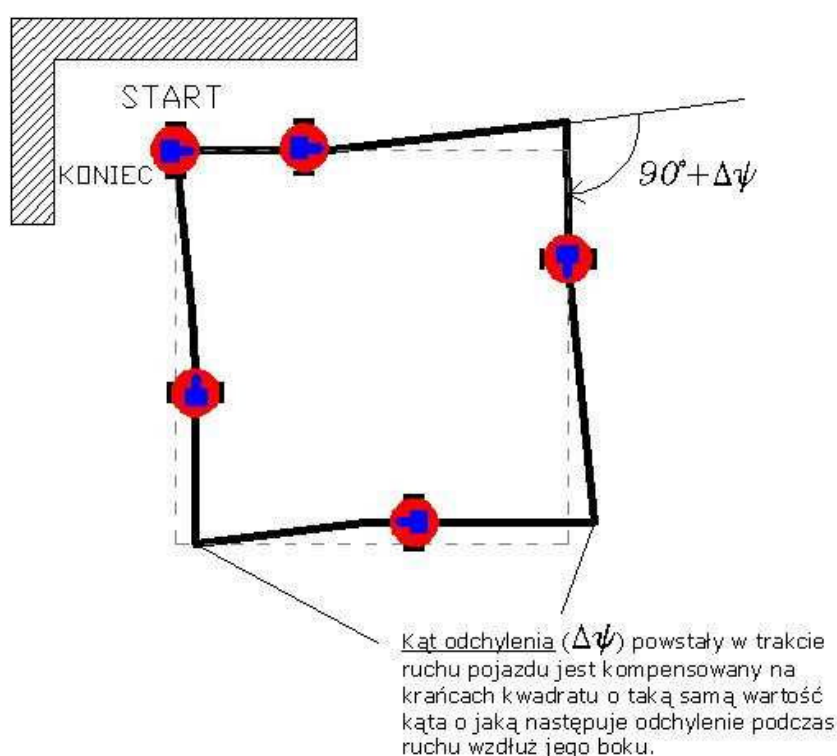
- a) nierówności średnic kół pojazdu;
- b) niedokładność w odległościach pomiędzy punktami styków kół robota z powierzchnią, po której się porusza;

W pokazanym przykładzie następuje stopniowe schodzenie z ustalonej trajektorii ruchu. Może być to spowodowane wystąpieniem co najmniej jednego rodzaju błędu lub też ich obu jednocześnie (gdzie jeden z nich może być znacznie większy niż drugi).

Sytuacja, gdzie dwa różne typy zakłóceń mogą spowodować nieprawidłowe funkcjonowanie aplikacji prowadzi do tego, iż korekcja jednego z nich sprawi, że

praca układu będzie prawidłowa tylko dla tego jednego szczególnego przypadku (jazdy w jedną stronę wzdłuż boków kwadratu). Eliminacja przyczyny powstania błędu (np. przy zmianie oprogramowania sterującego robota) przyniesie tylko pozorne korzyści, a rzeczywista przyczyna powstawania niedokładności nie zostanie do końca wyeliminowana.

Poniżej przedstawiony jest rysunek prezentujący wstępną korekcję zaistniałych zakłóceń. Choć wydaje się, że jest to postępowanie stosunkowo poprawne, to jednak należy zauważyć, że korekcja następuje tylko i wyłącznie w przypadku, gdy robot porusza się w jedną stronę – zgodnie ze wskazówkami zegara, czyli jedynie dla tak postawionej aranżacji aplikacyjnej.



Rys. 13 Przykład wstępnej korekcji pozycjonowania robota.

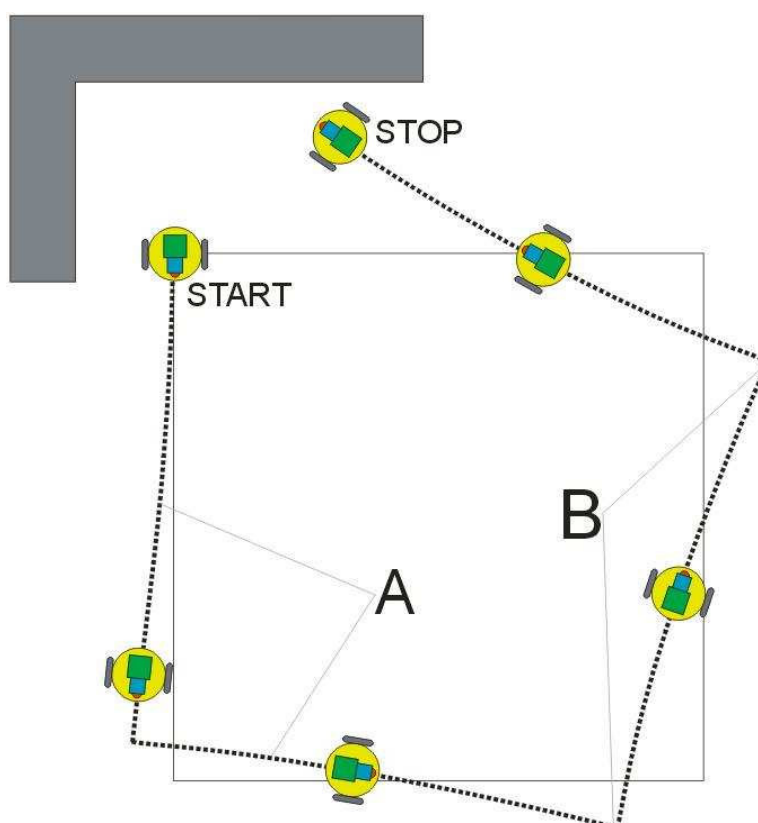
Powyższy test pokazuje jak mylące mogą być założenia przy subiektywnej ocenie wyników i obserwacji. Uszczegółowiony przykład ilustruje, iż jedynie jednokierunkowy eksperyment polegający na powrocie do miejsca rozpoczęcia ruchu

wzdłuż boku kwadratu jest nieodpowiedni do oszacowania jakości i dokładności układu jezdnego robota.

Aby pozbyć się tego problemu Borenstein i Feng ([10]) zaproponowali „**dwukierunkowy test kwadratowy**” nazywany często testem **UMBmark** – z ang.: „University of Michigan Benchmark”.

UMBmark wymaga, aby test odbył się wzdłuż boku kwadratu w dwóch kierunkach, tj. zgodnie i przeciwnie do ruchu wskazówek zegara.

Poniżej zaprezentowany jest rysunek, na którym można zauważyć wynik „utajonego” podwójnego błędu, gdy ruch pojazdu w przeprowadzonym teście odbywa się w odwrotnym kierunku niż w pierwotnej wersji „jednokierunkowego eksperymentu”.



Rys. 14 Efekt dwóch dominujących błędów systematycznych E_d oraz E_b , w przypadku testu „kwadratu” dla ruchu robota w kierunku przeciwnym do ruchu wskazówek zegara.

A – Łuk, zamiast prostej ścieżki ruchu pojazdu – ukazuje się tu efekt błędu nierówności średnic kół pojazdu: $E_{DL} \neq E_{DR}$.

B – Obrót o kąt nierówny wartości 90 stopni – potwierdza to niedoskonałość pozycjonowania (orientacji) robota z uwagi na niedokładności dla wzajemnej odległości kół pojazdu.

Uwidocznione są tu dwa dominujące błędy systematyczne, które kompensują się wzajemnie, dla danego kierunku ruchu, jednak ich wypadkowa suma dla innego przypadku ruchu jest na tyle duża, iż otrzymane wyniki różnią się radykalnie. Aby stwierdzić, iż błędy te nie są spowodowane jedynie przypadkowym czynnikiem można przeprowadzić je wielokrotnie i dopiero wtedy porównać wyniki.

W miejscu tym można stwierdzić, co następuje:

- Pozycje końcowe robota po przejechaniu określonego etapu „zamykają się” w pewnych dwóch obszarach;
- „Rozrzut” wyników i powstanie swego rodzaju klastrow spowodowane jest błędami niesystematycznymi ;

Podczas przeprowadzania powyższego eksperymentu można określić stopień wpływu „czynników przypadku” na ruch i dokładność pozycjonowania robota w środowisku – czyli skali i średniej wielkości błędów niesystematycznych. Wartości te mogą być z łatwością oszacowane i być jednocześnie wskaźnikiem błędów maksymalnych jakie mogą wystąpić w danej aplikacji.

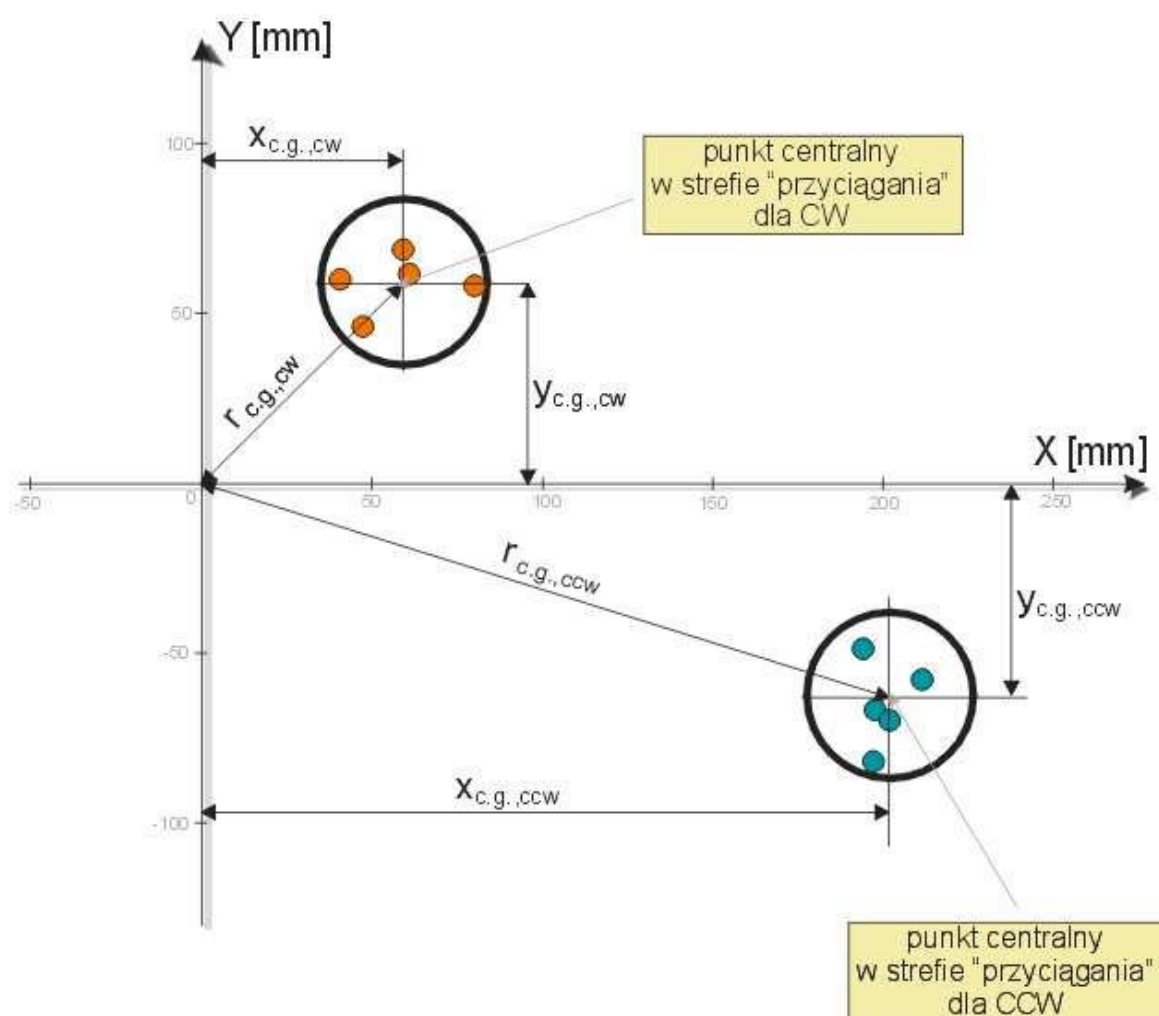
Zdefiniowane są tu takie współczynniki środków ciężkości (ang. „centers of gravity”), które pozwalają na oszacowanie zakresu „rozrzutu” wyników i jednocześnie określenia ich wpływu na postać kształtowania się zakłóceń w danej aplikacji.

CW – Clockwise – w stronę zgodną z ruchem wskazówek zegara;

CCW – Counter clockwise – w stronę przeciwną do ruchu wskazówek zegara;

$$x_{c.g.,cw/ccw} = \frac{1}{n} \sum_{i=1}^n \Delta x_{i,cw/ccw};$$

$$y_{c.g.,cw/ccw} = \frac{1}{n} \sum_{i=1}^n \Delta y_{i,cw/ccw};$$



Rys. 15 Przykładowe wyniki dla dwukierunkowego testu „kwadratu”.

Przesunięcia te pokazują różnice pomiędzy poszczególnymi testami, gdzie jeden pojazd przemierza określoną trasę kilkakrotnie (w dwóch kierunkach) i za każdym razem zatrzymuje się w innym miejscu. Dokładność z jaką potrafi on skończyć

swoją „podróż” określają wielkości „ $r_{c.g.,cw}$ ” oraz „ $r_{c.g.,ccw}$ ” których wyznaczenie polega na obliczeniach przeprowadzonych na podstawie z wykresu graficznego zamieszczonego powyżej. Tak więc wartości przesunięcia dwóch „centrów” od pozycji „teoretycznej” – czyli tej, w której pojazd zatrzymałby się w idealnym przypadku można wyliczyć z poniższych zależności:

$$r_{c.g.,cw} = \sqrt{(x_{c.g.,cw})^2 + (y_{c.g.,cw})^2}$$

$$r_{c.g.,ccw} = \sqrt{(x_{c.g.,ccw})^2 + (y_{c.g.,ccw})^2}$$

Ostatecznie można zdefiniować maksymalny błąd dla odometrii (metody (z ang.) „dead reckoning”) wyznaczany przy uwzględnianiu zakłóceń systematycznych:

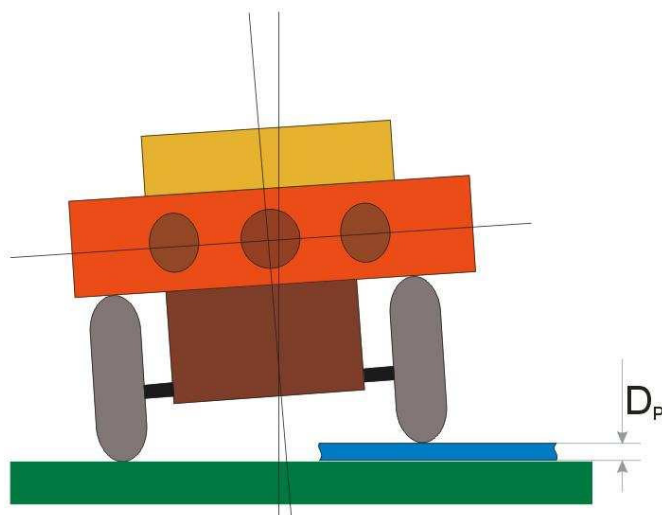
$$E_{\max, syst} = \max(r_{c.g.,cw}; r_{c.g.,ccw})$$

ZAWSZE NALEŻY UWZGLĘDNIĆ MAKSYMALNĄ WARTOŚĆ DLA PRZESUNIĘĆ „CENTRÓW” Z UWAGI NA TO, IŻ W APLIKACJACH PRAKTYCZNYCH TRZEBA PRZYJMOWAĆ NAJWIĘKSZE ODCHYLENIE OD WARTOŚCI OCZEKIWANEJ (TEORETYCZNEJ) – czyli trzeba spodziewać się, że wystąpią maksymalne błędy pozycjonowania.

3.1.2.2 Oszacowanie błędów niesystematycznych

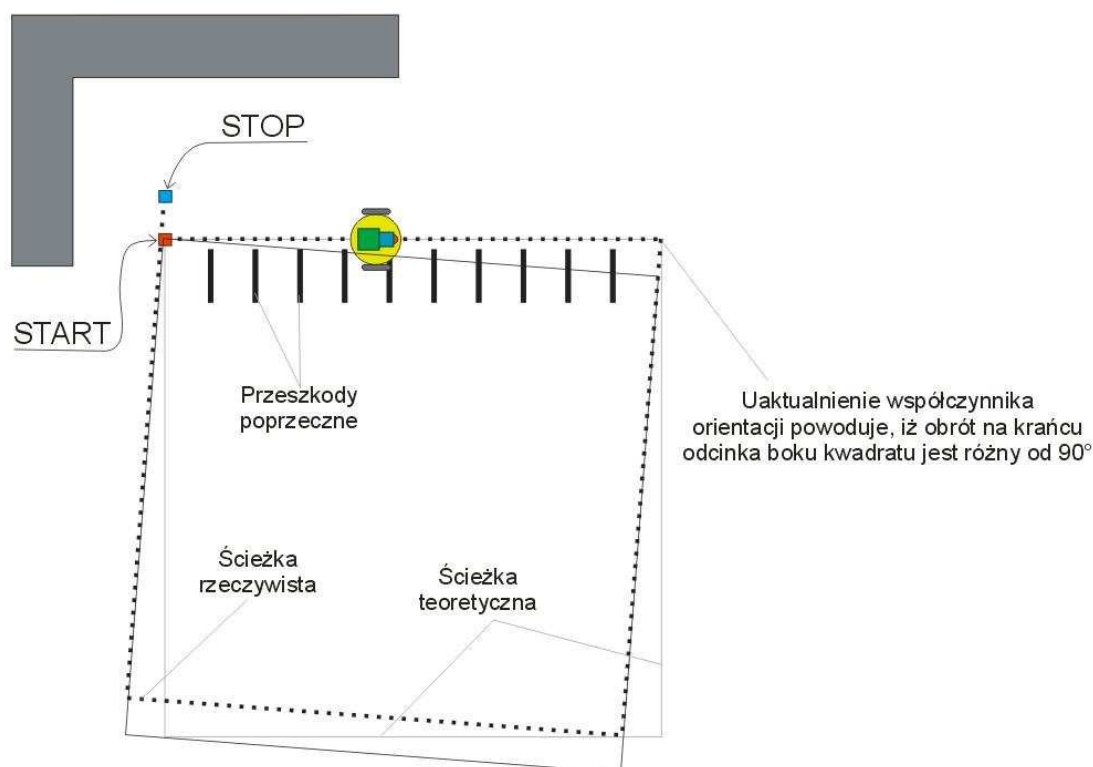
O wiele bardziej złożonym i trudniejszym zadaniem jest szacowanie błędów **niesystematycznych**, które mogą pojawić się w aplikacji na skutek działań zewnętrznych. Opracowane zostały jednak niezbyt złożone metody, które pozwalają na przeprowadzenie testu i oszacowanie tego rodzaju zakłóceń. Test ten nosi nazwę rozszerzonej metody „UMBmark” (z ang. „extended UMBmark”).

Polega on, co prawda na takim samym zachowaniu się pojazdu jak dla UMB-mark dwukierunkowego (czyli ruchu wzdłuż boków kwadratu w dwóch kierunkach) lecz dodatkowo „wzbogaca” się trasę robota o sztuczne przeszkody w postaci lokalnych wzniesień. Utrudnienia te powodują, że koła pojazdu zachowują się nieco inaczej aniżeli dla równego terenu i powstają zakłócenia w ruchu pojazdu powodujące błędy. W aplikacji praktycznej można użyć małych belek poprzecznych (przewód elektryczny, nieruchoma rurka, itp. o określonej średnicy D_p) rozmieszczonych równomiernie wzdłuż jednego z boków kwadratu. Przeszkoda taka umieszczana powinna być tylko na trasie jednego z kół – najlepiej tylko pod tym, które jest we wnętrzu kwadratu dla określonego kierunku jazdy.



Rys. 16 Przykładowe umiejscowienie przeszkody dla pojazdu w teście określającym błędy niesystematyczne.

Przejechanie przez taką trasę będzie obarczone błędem wynikającym z przesuwania się niekontrolowanego kół wskutek nierównomierności podłoża. Choć nie będzie to nazbyt widoczne to jednak pojawi się odczuwalna różnica pomiędzy tą aplikacją, a testem bez takich przeszkód.



Rys. 17 Zmiany ścieżki pojazdu wskutek umieszczenia sztucznych przeszkód na drodze przejazdu.

Na powyższym rysunku można zaobserwować jak wygląda symulacja dla tego rodzaju testu i jak może zmieniać się położenie pojazdu przy uwzględnieniu czynników zakłócających jego prawidłową pracę.

Parametrem, który pozwalał będzie na ocenę i jednocześnie oszacowanie wielkości powstałych błędów niesystematycznych jest zmiana „orientacji” robota względem środowiska. Dzieje się to dlatego, iż wpływ wymienionych wyżej przeszkód na zmianę położenia robota (a nie jego orientacji) jest stosunkowo znikomy. O wiele ciekawsze i interesujące wnioski można wysnuć, gdy bierze się pod uwagę właśnie zmianę kąta orientacji pojazdu – zachodzą tu bardzo widoczne zmiany.

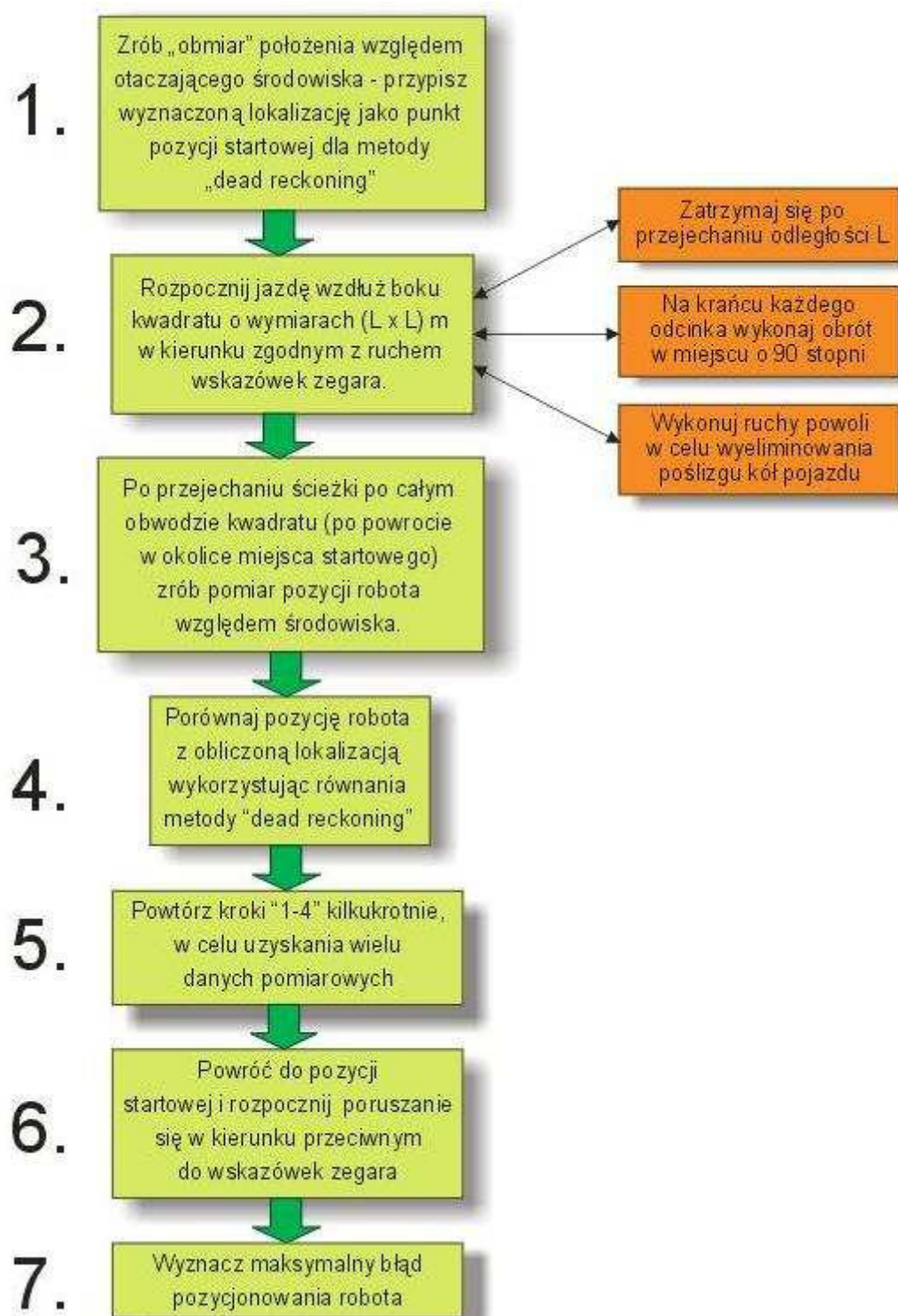
Z należyłą dokładnością należy tu oczywiście podać wielkości, które pozwalają na ocenę tego błędu:

$$\Delta\Theta_{\acute{s}r}^{ns} = \frac{1}{n} \sum_{i=1}^n |\Delta\Theta_{i,cw}^{ns} - \Delta\Theta_{\acute{s}r,cw}^s| + \frac{1}{n} \sum_{i=1}^n |\Delta\Theta_{i,ccw}^{ns} - \Delta\Theta_{\acute{s}r,ccw}^s|$$

ns	Skrót od niesystematyczny
s	Systematyczny
śr	Skrót od średni
cw	ang. clockwise - zgodnie z ruchem wskazówek zegara
ccw	ang. counter-clockwise - przeciwnie do ruchu wskazówek zegara
n	Liczba kroków w iteracji powtórzeń ruchu robota
$\Delta\Theta_{\acute{s}r}^{ns}$	jest to błąd średni bezwzględny orientacji robota

Należy tu nadmienić, iż jest tu wyszczególniony czynnik odpowiadający za wielkość zakłóceń (niedokładności) systematycznych. Jest on odejmowany od wielkości niesystematycznych uzyskiwanych w każdej iteracji algorytmu, dzięki czemu uzyskuje się rzeczywistą wartość błędu.

Całkowity algorytm pozwalający oszacować błędy w teście „kwadratu” dla rzeczywistego układu można przedstawić następująco:



Rys. 18 Postać blokowa algorytmu wykrycia i oszacowania błędów ruchu

3.1.2 Pozycjonowanie bezwzględne

Techniki lokalizacji bezwzględnej pomagają zwiększyć dokładność pozycjonowania względnego. Każde z podejść pozwala na lepszą zdolność lokalizacji robota w przestrzeni, lecz każde z nich ma również swoje wady. Istnieje wiele rozwiązań w łączeniu różnych technik pozwalających na osiągnięcie stosunkowo dobrych wyników w interesującym zagadnieniu, jednak każda z nich posiada niedoskonałości. Wiele czynników takich jak na przykład czas, pieniądze, moc obliczeniowa wpływają na wybór najlepszego rozwiązania dla danej aplikacji – należy często osiągnąć kompromis pomiędzy nimi, aby cel postawiony zadaniu został osiągnięty jak najmniejszym kosztem.

3.1.2.2 Znaczniki

Znaczniki, które mogą być wykorzystane przy lokalizacji bezwzględnej w terenie można podzielić na dwie grupy:

- a) czynne
- b) bierne

Czujniki czynne charakteryzują się tym, iż posiadają nadajnik i odbiornik sygnału. Jako przykład może posłużyć tu system **GPS** (Global Positioning System), który wysyła i jednocześnie odbiera sygnały.

Z kolei głównym elementem w urządzeniach biernych jest tylko odbiornik – tutaj można stworzyć układ znaczników biernych umieszczając w środowisku charakterystyczne kody (np. w postaci znanych pasków kodowych).

Znaczniki pozwalają na bardzo dokładne zlokalizowanie robota, jednak wymagają dużego wkładu z zewnątrz. Niektóre aplikacje z wykorzystujące wyżej wymienione techniki mogą być jednak nieprzydatne w pewnych okolicznościach, jak na przykład GPS w budynkach. Z kolei umieszczenie w terenie, w którym ma się odbywać praca robota pewnych elementów służących lokalizacji może być po prostu niemożliwe do zrealizowania.

3.1.2.3 Filtr Kalmana

Filtr Kalmana używany do lokalizacji składa się z dwóch modeli:

- stanu;
- pomiaru;

Model stanu opisuje, w jaki sposób pozycja robota zmienia się wraz z czasem jako następstwo (wynik) sterowania i zakłóceń procesu ruchu. Z kolei model pomiaru dostarcza dane otrzymane z czujników badających otaczające środowisko. Celowość użycia filtru Kalmana w takim postępowaniu to przede wszystkim oszacowanie lokalizacji robota przy wykorzystaniu trzech istotnych elementów:

- a) poprzedniej pozycji robota
- b) charakterystyki sterowania
- c) danych z czujników

3.1.3 Czujniki i rozwiązania aplikacyjne

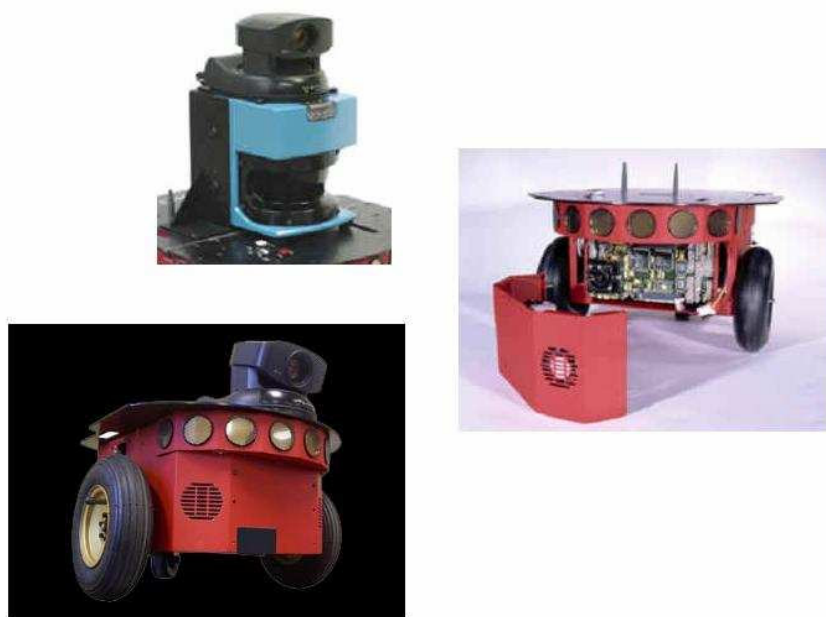
Istnieje wiele różnych sposobów pozwalających na skuteczne określanie pozycji robota w terenie. Spośród licznych rozwiązań można wybrać te, które będą odpowiadały stawianym wymaganiom i spełniały określone przez projektanta funkcje. Można wyróżnić tu przede wszystkim:

Żyroskopowy system korekcji - jest on używany w aplikacjach, gdzie mogą wystąpić niekorzystne czynniki wpływające na funkcjonowanie systemu pozycjonowania. (rozwiązanie to zostało szerzej opisane w poprzednich rozdziałach).

Skaner laserowy – używany do zobrazowania 3-wymiarowego przestrzeni, mający zasięg od kilku do kilkudziesięciu metrów i kąt widzenia od 90 do 180 stopni.

Czujniki zderzakowe – pozwalają na kontaktowe badanie przestrzeni i pomagają w przemieszczaniu się robota bez wysokiego ryzyka uszkodzenia. Nawet czujniki ultradźwiękowe i laserowe posiadają swoje tzw. “martwe punkty”, a niedokładne odczyty danych środowiskowych mogą spowodować niespodziewane “spotkania” z przeszkodami i awarię urządzenia.

Wszystkie te rozwiązania i w ich skład wchodzące podzespoły posiadają interfejs ARIA (skrót z ang. ActiveMedia Robotics Interface Application) pozwalający na wykorzystanie zalet programowania obiektowego i może być używany pod takimi systemami operacyjnymi jak Linux czy też Windows.



Rys. 19 Przykłady czujników wykorzystywanych w aplikacjach robotów mobilnych.

3.2 Budowa i kompozycja mapy

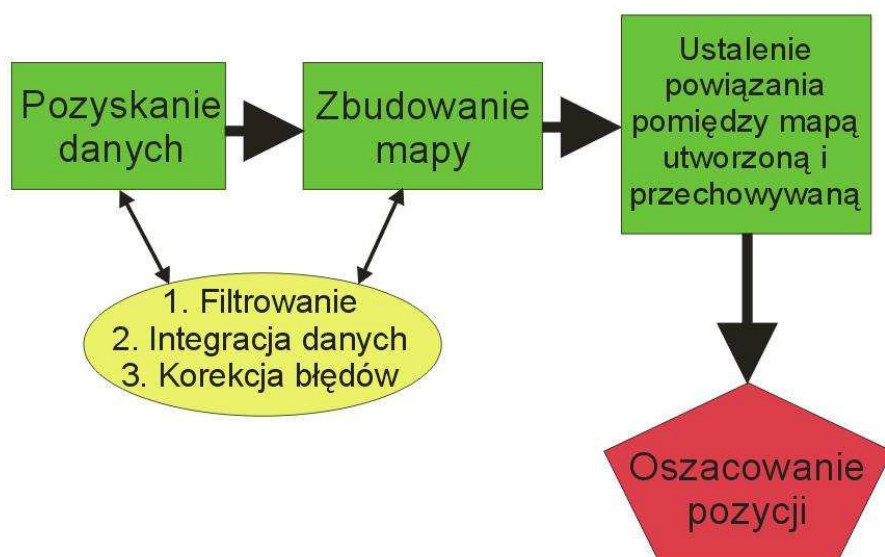
Wspomniane wcześniej techniki lokalizacji pomagają w pozycjonowaniu pojazdów mobilnych względem otaczającego je środowiska. Jednak, gdy etap samolokalizacji robota będzie już osiągnięty, wtedy naturalnym krokiem jest postawienie przed danym urządzeniem kolejnego zadania. Jako przedmiot rozważań przyjęto w tym miejscu konstrukcję mapy, która będzie tworzona na podstawie danych uzyskanych z otaczającego robota środowiska.

3.2.1 Pozycjonowanie względem mapy

Pozycjonowanie na podstawie mapy jest elementem, dzięki któremu pojazd ma możliwość określenia swojego położenia względem bliskiego otoczenia (jest to nazywane często dopasowaniem map). Metoda ta jest podobna do umieszczania

znaczników w środowisku. Robot będzie odtwarzał mapę terenu w trybie eksploracji lub też będzie posiadał potrzebne dane już w swojej pamięci przed podjęciem jakiegokolwiek akcji. Jeśli urządzenie mobilne zawierałoby mapę środowiska w swojej pamięci to wtedy, z oczywistych względów, zwiększona zostaje ingerencja czynników zewnętrznych w funkcjonowanie robota i zachwiana zostaje autonomiczność takiej aplikacji. Jeśli z kolei robot przeprowadza misję eksploracji i rozpoznawania terenu wtedy mogą pojawić się duże błędy w lokalizacji i niemożliwe jest skuteczne określenie położenia urządzenia. Tak więc metoda ta jest przydatna tylko w wypadku, gdy można wyposażyć robota mobilnego w dane mapy dotyczące środowiska, w którym ma się on poruszać. Często takie postępowanie jest postrzegane jako odwieczny problem co było pierwsze: „jajko czy kura”, ponieważ aby dokonać lokalizacji robota w środowisku potrzebna jest mapa tego środowiska. Ale jeśli odbywa się to w dotychczas nie rozpoznanym terenie wtedy musimy najpierw uzyskać dane dotyczące mapy, aby na ich podstawie móc oszacować pozycję pojazdu. Tak więc problem dopasowania mapy jest niekiedy kłopotliwy ze względu na swoją naturę. Jednak częściowe wykorzystanie tego sposobu „poznawania” terenu będzie z pewnością pomocne w budowaniu już samej mapy przez urządzenia mobilne i wtedy na jej podstawie inne jednostki (na przykład mniej złożone i mniej inteligentne roboty) mogłyby wykorzystywać do swoich celów.

Podstawowy schemat samej procedury jaką można wykorzystać do naszych celów przedstawia się następująco:



Rys. 20 Postać blokowa algorytmu pozycjonowania pojazdów względem mapy.

Główne korzyści z powyższej metody to:

Metoda ta wykorzystuje naturalną strukturę środowisk wewnętrznych, tj. pomieszczeń zamkniętych bez żadnych specjalnych modyfikacji, aby pozyskać dane bez konieczności wprowadzania zmian w terenie;

Wygenerowanie mapy i jej uaktualnienie może polegać na pozycjonowaniu bazującym na dopasowaniu mapy;

Dokładność eksploracji i „nauka” robota może zostać powiększona właśnie przez metody dopasowania mapy;

Ograniczenia metody

Nie ma wystarczająco dużo charakterystycznych elementów środowiska możliwych do użycia w prezentowanym sposobie postępowania;

Używane czujniki są wystarczająco dokładne do oszacowania elementów otaczającego terenu – nie ma potrzeby przeprowadzania dodatkowych obliczeń;

Urządzenia biorące udział w zadaniu nie posiadają wystarczającej mocy (obliczeniowej) do przeprowadzenia operacji odbierania danych z czujników i jednoczesnej ich analizy;

Metody bazujące na dopasowaniu mapy są bardzo istotne dla całościowego ujęcia problemów dotyczących lokalizacji, eksploracji i budowaniu map jak na przykład globalne planowanie ruchu czy też ogólnie rzecz ujmując - unikania różnego rodzaju przeszkód.

3.2.2 Metoda FastSLAM

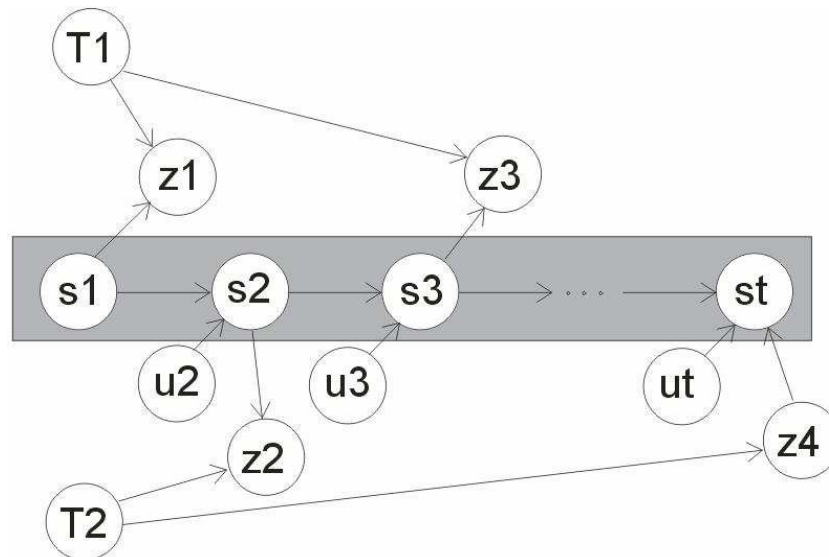
Jak zostało to już wcześniej przedstawione zdolność do jednoczesnej lokalizacji i dokładnej analizy otaczającego środowiska jest uważane za kluczowy element dla stworzenia prawdziwie autonomicznego robota.

Jednakże istnieją pewne ograniczenia wynikające z mocy obliczeniowej jednostek w stosunku do obecności obserwowanych charakterystycznych punktów w środowisku i konieczności przeanalizowania ich położenia. Jako przykład może posłużyć algorytm bazujący na filtrze Kalmana, który potrzebuje na obliczenia czas proporcjonalny do kwadratu liczby punktów charakterystycznych, aby rejestrować dane z każdego czujnika.

Szczególnie interesującym rozwiązaniem jest zastosowanie algorytmu nazywanego FastSLAM ¹⁷, który pozwala na wyznaczenie pełnego obrazu zależności

między pozycją robota i charakterystycznymi punktami (odniesienia) znajdującymi się w środowisku.

Sam problem można przedstawić przy pomocy poniższego schematu:



Rys. 21 Postać blokowa algorytmu pozycjonowania pojazdów względem mapy.

Pojazd porusza się z pozycji **s1** i przechodzi sekwencję danych sterujących **u1**, **u2**, ..., **ut**. Podczas wykonywania ruchu obserwuje otaczające punkty charakterystyczne terenu.

Dla czasu $t=1$:

Następuje rejestracja położenia punktu **T1** spośród dwóch możliwych do przetworzenia: **{T1, T2}**. Sam pomiar odnotowywany jest jako **z1** (oznaczający dystans i ukięrunowanie).

Dla czasu $t=2$:

Obserwowany jest kolejny punkt **T2** i określane jest **z2**.

Podczas kolejnych odczytów danych definiowane są nowe elementy pozwalające na lokalizację robota w środowisku.

Pozycje robota, jak wspomniano wcześniej, opisywane są jako funkcja danych sterujących **u1, u2, ..., ut**. Każdy z pomiarów uzyskanych z procesu eksploracji i „pozna-

wania” otoczenia przedstawiany jest jako funkcja położenia punktu T_k i pozycji robota. Sam pomiar reprezentowany jest przez $z_1, z_2, \dots, z_t$.

Metoda FastSLAM pozwala więc na oszacowanie położenia robota w środowisku na podstawie lokalizacji punktów charakterystycznych przy wykorzystaniu danych sterujących u i pomiarów z . Co więcej, dokładnie poznane położenia pojazdu $s_1, s_2, \dots, s_t$ sprawiają, że pomiary poszczególnych punktów stają się niezależne. Jeśli więc posiada się informację o położeniach, można ją podzielić na K podzadań, co w znaczący sposób ułatwia przeprowadzenie niezbędnych obliczeń.

4. Eksploracja nieznanego terenu w ujęciu inteligencji grupowej robotów – symulacja

W projekcie symulacji nie ma określonego w sposób szczegółowy sposobu poruszania się robota w środowisku. Część z elementów, jakkolwiek newralgicznych i ważnych, jest poza zakresem opracowania. Pośród najważniejszych komponentów należy wymienić przede wszystkim:

- brak uwzględnienia obrotów kół;
- pojazd przemieszcza się od komórki do komórki po elementach siatki;
- aktualizacja pozycji następuje w ściśle określonych momentach (jest to przedstawione przy wykorzystaniu licznika ruchu – kroku iteracyjnego)

SYMULACJA MA ZA ZADANIE PRZEDSTAWIĆ, ŻE MIMO UŻYCIA BARDZO PROSTYCH ROBOTÓW MOŻNA UZYSKAĆ ZADOWALAJĄCO DOBRE WYNIKI PRZY EKSPLOKACJI NIEZNAJNEGO TERENU PRZEZ ZESPÓŁ URZĄDZEŃ MOBILNYCH.

4.1 Opracowanie projektowe symulacji dla grupy współdziałających robotów

PROJEKT SYMULACYJNY STWORZONY DLA:

- MAŁYCH JEDNOSTEK MOBILNYCH W STOSUNKU DO OTACZAJĄCEGO ŚRODOWISKA;
- KOMUNIKACJA POMIĘDZY JEDNOSTKAMI ODBYWA SIĘ W SPOSÓB POŚREDNI (DANE PRZECHOWYWANE WE WSPÓLNEJ PAMIĘCI BEZ BEZPOŚREDNIEJ KOMUNIKACJI POMIĘDZY POJAZDAMI);
- MAPA TWORZONA DLA DUŻEGO OBSZARU - NIE MA USZCZEGÓŁOWIENIA STRUKTURY ŚCIAN – MAPA MA POSRTAĆ RASTROWĄ.

4.2 Założenia dla projektu eksploracji terenu przez grupę robotów mobilnych

Założenia:

1) dla symulacji;

- cel projektu

Celem projektu jest przedstawienie grupy współdziałających robotów posiadających umiejętność rozpoznawania pewnego otaczającego je terenu. Zadanie postawione przed nimi to eksploracja i utworzenie mapy badanego przez nie obszaru.

W projekcie dotyczącym powyższego tematu będą zawarte:

- opis problemu badania nieznanego środowiska (zamkniętego pomieszczenia) przez grupę robotów;
- zaproponowanie algorytmu eksploracji;
- implementacja (symulacja komputerowa);
- podsumowanie – wyniki i wnioski;

2) dla terenu;

Typ terenu:

- płaski – nie ma lokalnych i globalnych przeszkód w postaci wzniesień, spadków, czy też innych elementów uniemożliwiających swobodne poruszanie się w badanym terenie. Pominięty zostanie problem samoistnego i niekontrolowanego ruchu (jak np. „ześlizgiwanie się” po powierzchni);
- statyczny – teren pozostaje w takim samej postaci przez cały czas eksploracji. Niemożliwe są zmiany geometryczne przestrzeni (przesuwanie się ścian, znikanie elementów na badanym obszarze, itp.);
- zamknięty – nie dopuszcza się eksploracji nieograniczonego obszaru – czas trwania symulacji musi być skończony;

3) dla pojedynczego robota;

- budowa robota

Symulacja czujników – robot będzie posiadał czujniki zbierające dane o środowisku w najbliższym otoczeniu (określony zakres, kąty działania, ukierunkowanie).

Pojedyncza jednostka dzieli się informacjami z innymi agentami w określony sposób odwzorowując uzyskane dane we wspólnej pamięci.

Robot wykrywa tylko te obiekty, których nie może pokonać bez zmiany kierunku poruszania się (duże przeszkody, ściany, itp.). Małe obiekty, jak kamienie w rzeczywistym świecie, nie będą wykrywane i dodatkowo zakłada się iż ich pokonanie nie sprawi jednostce żadnego istotnego problemu.



Rys. 22 Przykład rzeczywistego robota mobilnego - PIONEER 2.

- poruszanie się w terenie według określonych reguł

Podstawowymi elementami będą tu:

- unikanie (omijanie) przeszkód;
- unikanie kolizji z innymi urządzeniami;

- eksploracja niezbadanego jeszcze terenu (zbadany przez inne jednostki fragment terenu będzie w miarę możliwości pomijany);

4) dla grupy robotów;

- roboty dzielą informację o odwiedzanych miejscach;
Będzie to główny element grupowego współdziałania – Informacja będzie przekazywana w sposób pośredni.
- unikanie kolizji;
- zbudowanie mapy terenu;

Kontynuowanie symulacji aż do całkowitego zbadania terenu i konstruowania wirtualnej mapy obszaru.

5) dla programu symulacyjnego;

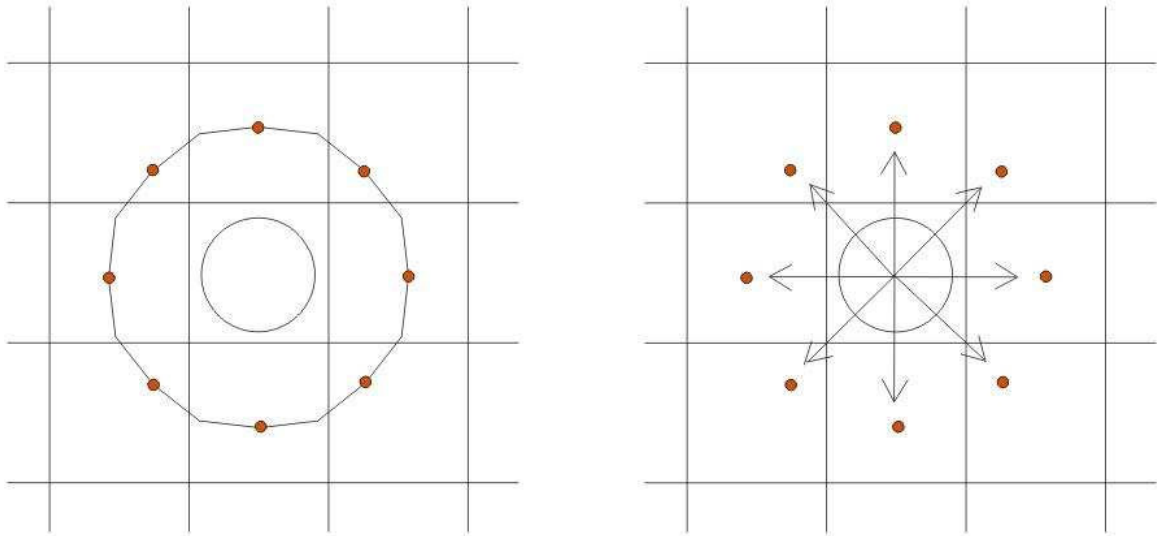
- wizualizacja jednostek pracujących w terenie – sposób poruszania się na określonym obszarze.
- możliwość zmiany ustawień symulacji (liczby pracujących robotów, prędkości (czasu) eksploracji, błędów, itp.)
- pokazanie odczytanej mapy na podstawie wskazań robotów.

6) wyniki i wnioski;

- przedstawienie charakterystyk pokazujących zależności pomiędzy poszczególnymi konfiguracjami grupy urządzeń.
- podsumowanie przeprowadzonej symulacji.

4.3 Modelowanie jednostek i algorytmy - koncepcja

Postać koncepcyjna, na podstawie której zrealizowana zostanie aplikacja symulacyjna przyjmie postać przedstawioną poniżej. Siatka rastrowa, która będzie jednocześnie płaszczyzną poruszania się wirtualnych jednostek, składa się z komórek określających teren – obecność przeszkód lub nie.



Rys. 23 Koncepcyjna postać robota umieszczonego na siatce rastrowej wraz z zasięgiem czujników odczytujących dane z bliskiego otoczenia.

Równanie koncepcyjne przedstawiające zależność służącą do określenia lokalizacji pojazdu:

$$L_{Rn}(x(k), y(k)) = f(L_{Rn}(x(k-1), y(k-1)), R_P(d_R, w_K), U_{NOISE})$$

$L_{Rn}(x(k), y(k))$ - aktualna lokalizacja;

$L_{Rn}(x(k-1), y(k-1))$ - funkcja lokalizacji wstecznej;

$R_P(d_R, w_K)$ - funkcja przemieszczania;

U_{NOISE} – funkcja zakłóceń

4.4 Postać aplikacyjna

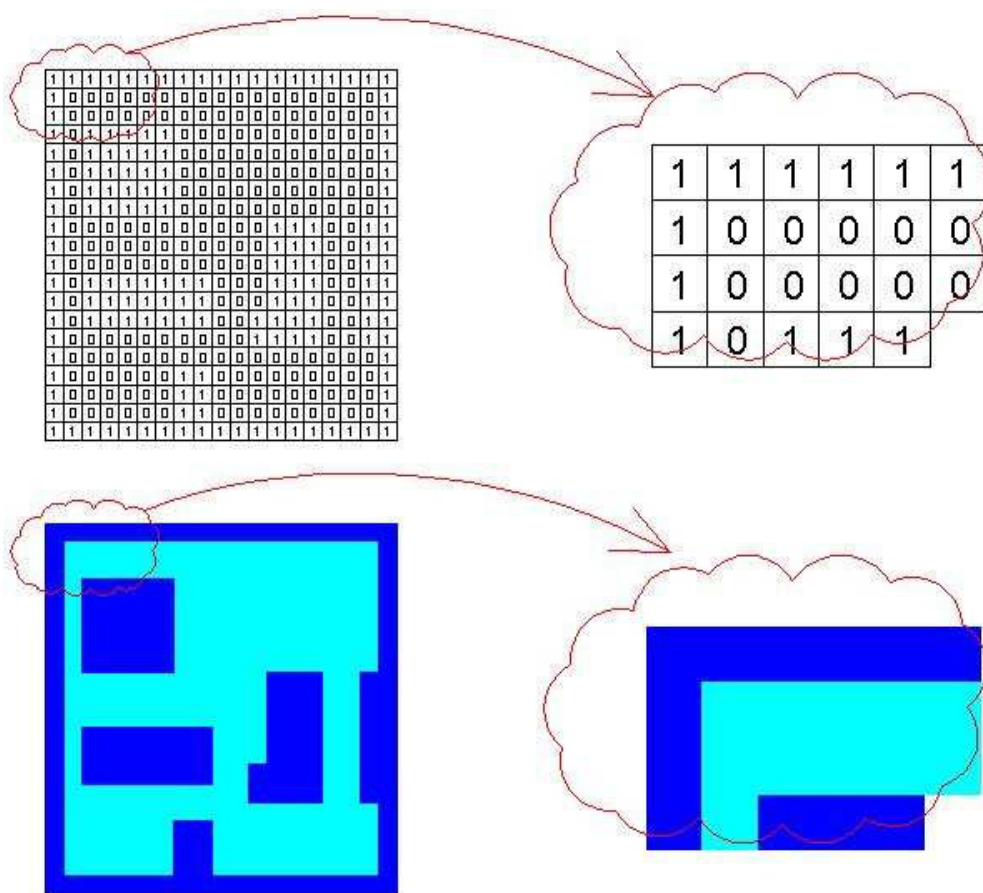
Jak wcześniej przedstawiono – w projekcie symulacyjnym nie ma odzwierciedlenia szczegółowo sposobu poruszania się robota w środowisku. Uwzględniono te parametry, które przedstawiają warunki, jakie spełniać powinny aplikacje odpowiadające tematyce eksploracji terenu i podstawom współpracy jednostek mobilnych. Do podstawowych elementów składowych procesu symulacji należy tu zaliczyć:

- model mapy rastrowej;
- postać funkcyjna i wykonawcza robota;
- schemat modelowania błędów i wykonania ruchu;
- sposoby określania pozycji robota w terenie (lokalizacja);
- odczyt danych z otaczającego środowiska;
- pozycjonowanie pojazdu (ukierunkowanie);
- sposoby wyboru ruchu;
- wykonanie ruchu;

Jako program do realizacji postawionych założeń został wybrany Matlab w wersji 6.5.0, który umożliwia prezentację badań dotyczących interesującego tematu i przedstawienia wyników powyższego procesu.

4.4.1 Model mapy rastrowej

Mapa jest określona jako obraz z zakresu grafiki rastrowej, gdzie można ją przedstawić przy pomocy pliku reprezentującego poziomo-pionową siatkę. Każdy z jej elementów ma postać kwadratu o wymiarach odpowiadających jednemu pikselowi w zakresie odpowiadającym bitmapie. Zaprezentowany jest tu zatem model przyporządkowania pikselowi położenia w dwuwymiarowej siatce oraz stanu danego piksela określany jako 1 lub 0 czyli jak dla schematu grafiki jednobitowej.



Rys. 24 Mapy w postaci siatki z odpowiednimi wartościami oznaczającymi jej zajętość (mapa zbudowana z rastrów).

Kolor każdego piksela jest definiowany osobno i jak widać zależy od wartości bitu dla danej komórki.

Mapę taką charakteryzują następujące właściwości:

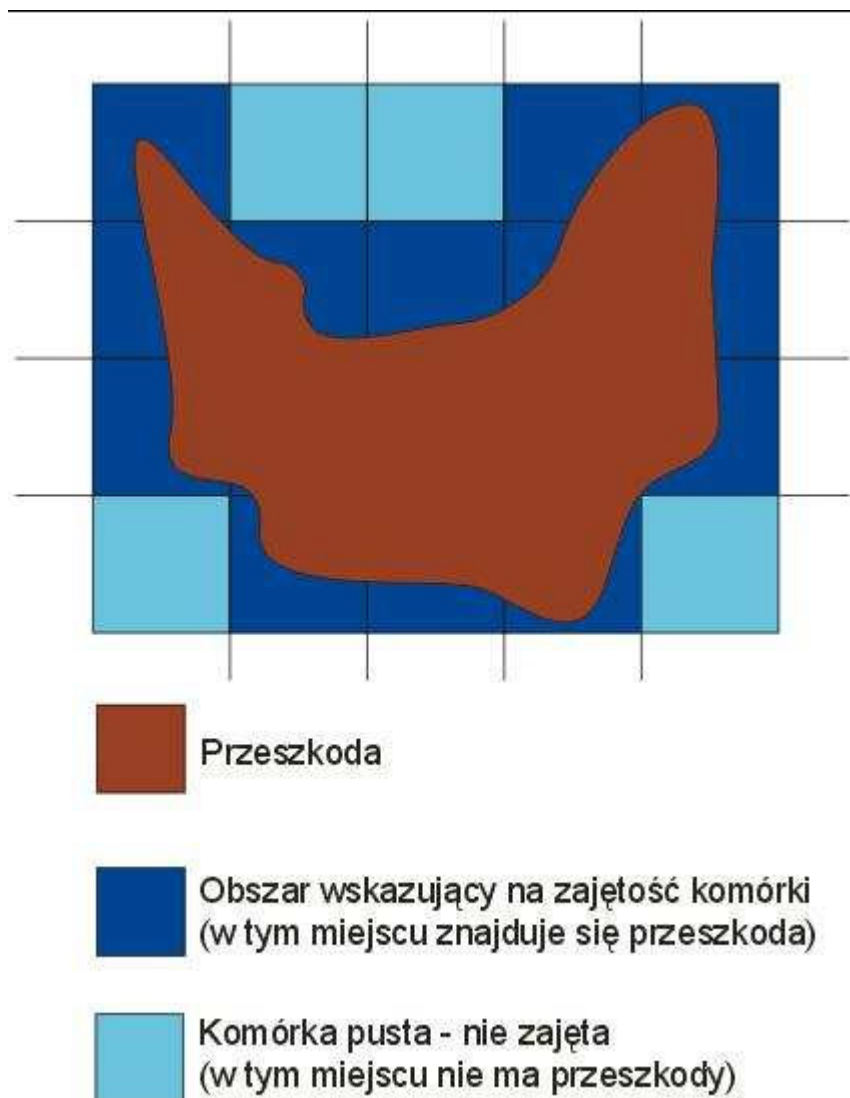
- wysokość i szerokość mapy liczona jest jako liczba pikseli i określana jest w programie w postaci macierzy o odpowiednich wymiarach;
- przypisanie wartości dla danej komórki jest jednocześnie przypisaniem danemu miejscu właściwości, która wskazuje na jej zajętość (oznaczenie przeszkody i ścianę lub obszaru otwartego);

	Tak nacechowany obszar oznacza przeszkodę i/lub ścianę
---	--



W ten sposób ujęta strefa oznacza wolną przestrzeń

- mapa ma charakter zamknięty, tj. jest ograniczona ścianami;
- mapa nie odzwierciedla małych przeszkód (znacznie mniejszych niż jeden raster), które uniemożliwiałyby pojazdowi poruszanie się w tym miejscu (jak określono w podrozdziale przedstawiającym założenia projektu);

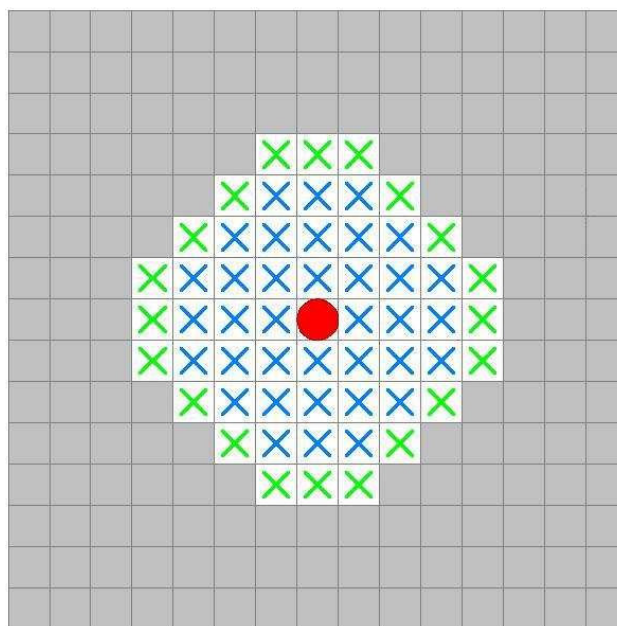


Rys. 25 Odzworowanie małej przeszkody i jej zniekształconej geometrii na mapie rastrowej

Powyższy rysunek przedstawia w jaki sposób małe przeszkody są odzwierciedlane na mapie.

4.4.2 Model robota

Pojazd jest scharakteryzowany jako jednostka mogąca odczytywać dane z otaczającego środowiska i na ich podstawie podejmować decyzje dotyczące wykonania ruchu. Ma on postać, która pozwala mu na przemieszczanie się po siatce mapy w dowolny sposób, ograniczony tylko do miejsc, gdzie występują przeszkody. Rysunek przedstawiony poniżej pokazuje koncepcyjną postać rozmieszczenia czujników (naniesioną na mapkę złożoną z rastrów) i sposobu poruszania się robota po zaprojektowanej mapie.



Rys. 26 Koncepcja modelu robota i położenia czujników.

Scharakteryzowane jest tu najbliższe otoczenie, którego każda komórka ma odpowiedni „adres” względem środka pojazdu. W trakcie odwołania programu przy wykonywaniu algorytmu ruchu następuje przypisanie właściwym komórkom pewnej

wartości. Może by to wskazanie na miejsce, w które pojazd musi się przemieścić lub też odwzorować na globalnej mapie jego stan (0 lub 1 – pusty lub zajęty)

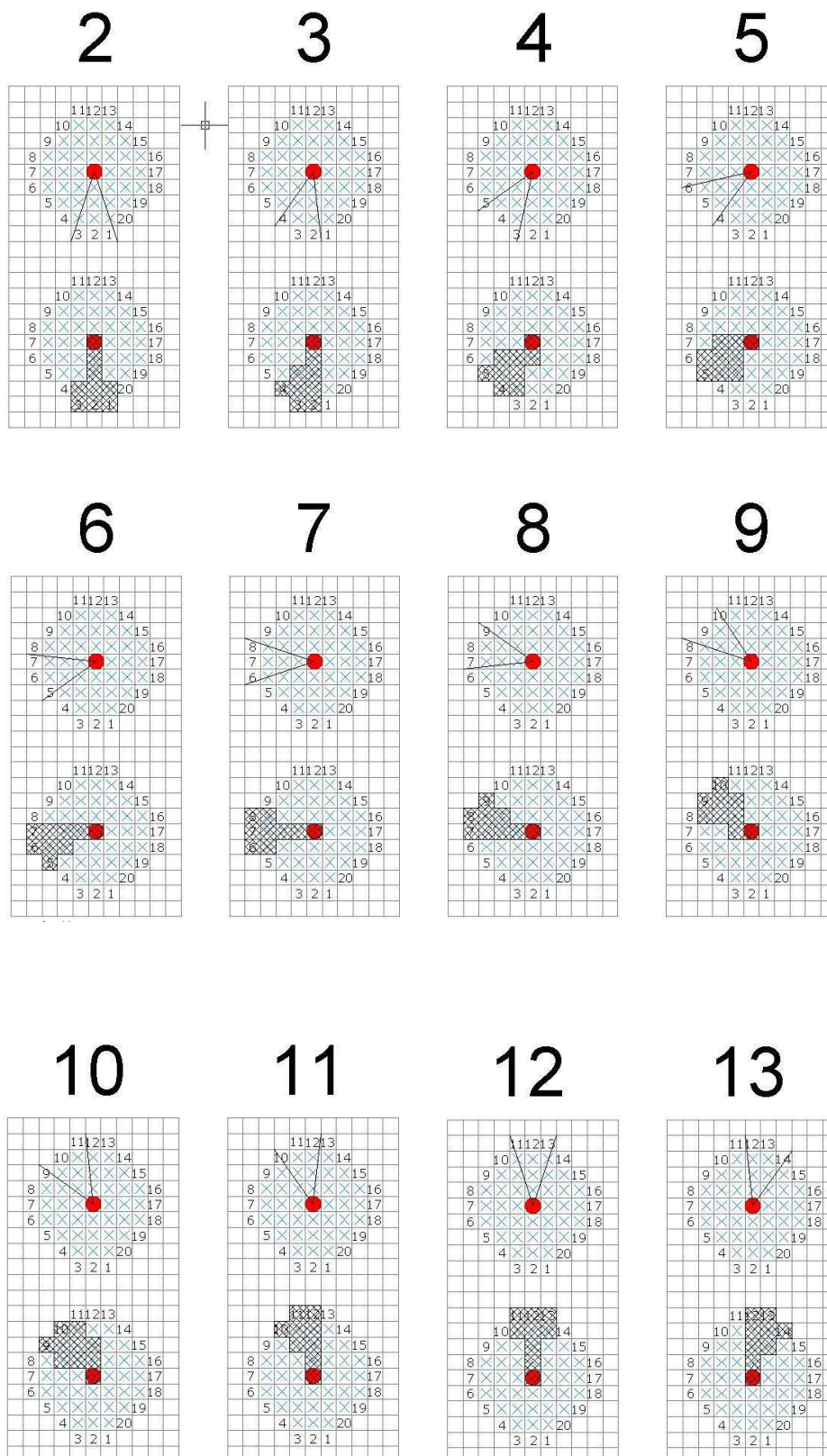
				$x_i = -1$ $y_i = -4$	$x_i = 0$ $y_i = -4$	$x_i = 1$ $y_i = -4$				
			$x_i = -2$ $y_i = -3$	$x_i = -1$ $y_i = -3$	$x_i = 0$ $y_i = -3$	$x_i = 1$ $y_i = -3$	$x_i = 2$ $y_i = -3$			
		$x_i = -3$ $y_i = -2$	$x_i = -2$ $y_i = -2$	$x_i = -1$ $y_i = -2$	$x_i = 0$ $y_i = -2$	$x_i = 1$ $y_i = -2$	$x_i = 2$ $y_i = -2$	$x_i = 3$ $y_i = -2$		
	$x_i = -4$ $y_i = -1$	$x_i = -3$ $y_i = -1$	$x_i = -2$ $y_i = -1$	$x_i = -1$ $y_i = -1$	$x_i = 0$ $y_i = -1$	$x_i = 1$ $y_i = -1$	$x_i = 2$ $y_i = -1$	$x_i = 3$ $y_i = -1$	$x_i = 4$ $y_i = -1$	
	$x_i = -4$ $y_i = 0$	$x_i = -3$ $y_i = 0$	$x_i = -2$ $y_i = 0$	$x_i = -1$ $y_i = 0$	x_{h1_00}	$x_i = 1$ $y_i = 0$	$x_i = 2$ $y_i = 0$	$x_i = 3$ $y_i = 0$	$x_i = 4$ $y_i = 0$	
	$x_i = -4$ $y_i = 1$	$x_i = -3$ $y_i = 1$	$x_i = -2$ $y_i = 1$	$x_i = -1$ $y_i = 1$	$x_i = 0$ $y_i = 1$	$x_i = 1$ $y_i = 1$	$x_i = 2$ $y_i = 1$	$x_i = 3$ $y_i = 1$	$x_i = 4$ $y_i = 1$	
		$x_i = -3$ $y_i = 2$	$x_i = -2$ $y_i = 2$	$x_i = -1$ $y_i = 2$	$x_i = 0$ $y_i = 2$	$x_i = 1$ $y_i = 2$	$x_i = 2$ $y_i = 2$	$x_i = 3$ $y_i = 2$		
			$x_i = -2$ $y_i = 3$	$x_i = -1$ $y_i = 3$	$x_i = 0$ $y_i = 3$	$x_i = 1$ $y_i = 3$	$x_i = 2$ $y_i = 3$			
				$x_i = -1$ $y_i = 4$	$x_i = 0$ $y_i = 4$	$x_i = 1$ $y_i = 4$				

Rys. 27 Umiejscowienia czujników – identyfikacja rastrów względem punktu centralnego.

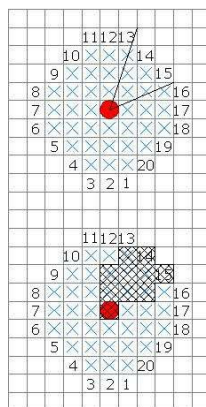
Jak widać, robot ten posiada czujniki pozwalające na wykrycie przeszkód znajdujących się w niewielkiej odległości i na ich podstawie odwzorować w globalnej pamięci to co znajduje się w jego sąsiedztwie.

Ograniczenia nałożone na projektowane oprogramowanie pozwalają na ukierunkowanie robota w 20 różnych położeniach, które, jak łatwo obliczyć, pozwalają na rozdzielczość kątową o jednostce odpowiadającej wartości 18-stu stopni. Aby zrealizować takie założenia należy określić zakresy, jakim odpowiadają ukierunkowania

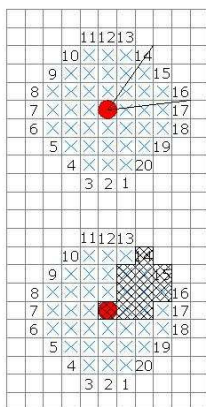
wirtualnych czujników. Przyjęta została następująca postać układu rastrów odwzorowująca kierunki i pozycje komórek, które pokrywa pole zasięgu czujek.



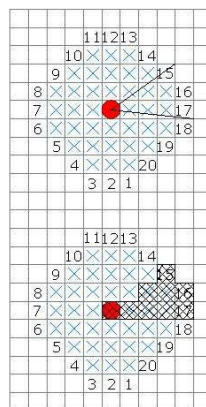
14



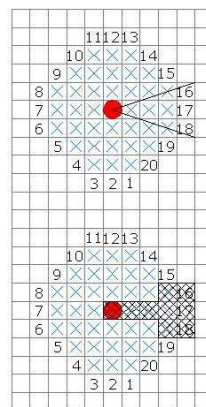
15



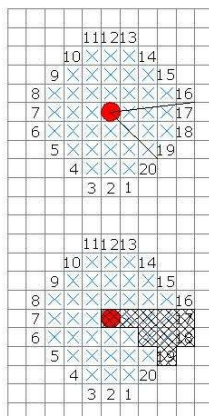
16



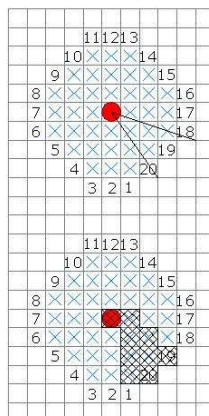
17



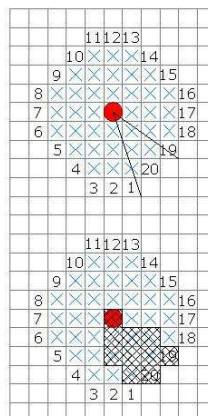
18



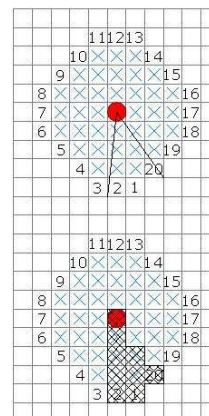
19



20



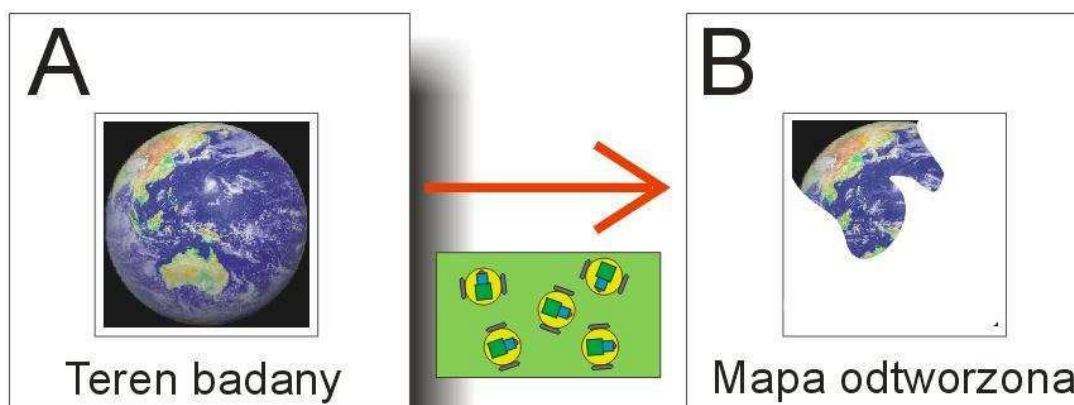
1



Rys. 28 Umiejscowienie czujników – ukierunkowanie pojazdu w 20-tu różnych pozycjach.

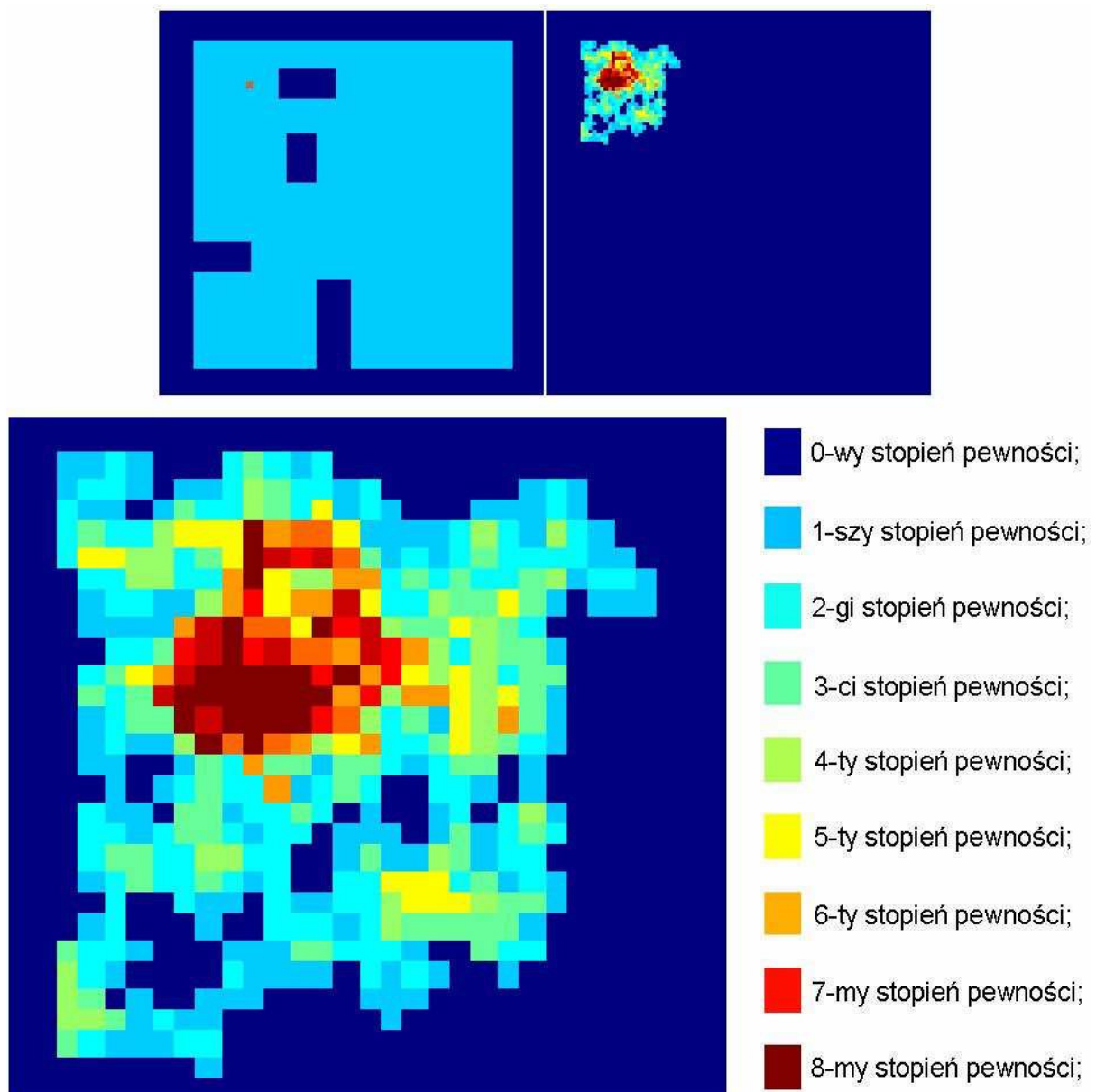
4.4.3 Odczyt danych ze środowiska i zapis w pamięci

Odczyt danych następuje bezpośrednio z badanej mapy. Przemieszczający się pojazd przy pomocy czujników może zbierać informacje dotyczące zajętości komórek – tj. określenia czy w danym miejscu znajduje się przeszkoda czy też nie. Tym samym zostaje tworzona przez zespół jednostek wspólna mapa, do której są na bieżąco (w czasie rzeczywistym) zapisywane informacje uzyskiwane podczas eksploracji wirtualnego terenu. Odzworowanie następuje więc w czasie przemierzania przez pojazd płaszczyzny środowiska i jednocześnie dane dotyczące odkrytego terenu zapisywane są na nowo kreowanej mapie.



Rys. 29 „Globalna” zasada działania symulacji

Bardziej dokładne zobrazowanie działania symulacji i sposobu zapisu odczytanych informacji do pamięci może przybliżyć poniższy rysunek. Jest tu zaprezentowana mapa wraz z jej odkrytym (zbadanym) fragmentem:



Rys. 30 Częściowa eksploracja mapy i powiększony fragment ze zróżnicowanym współczynnikiem pewności

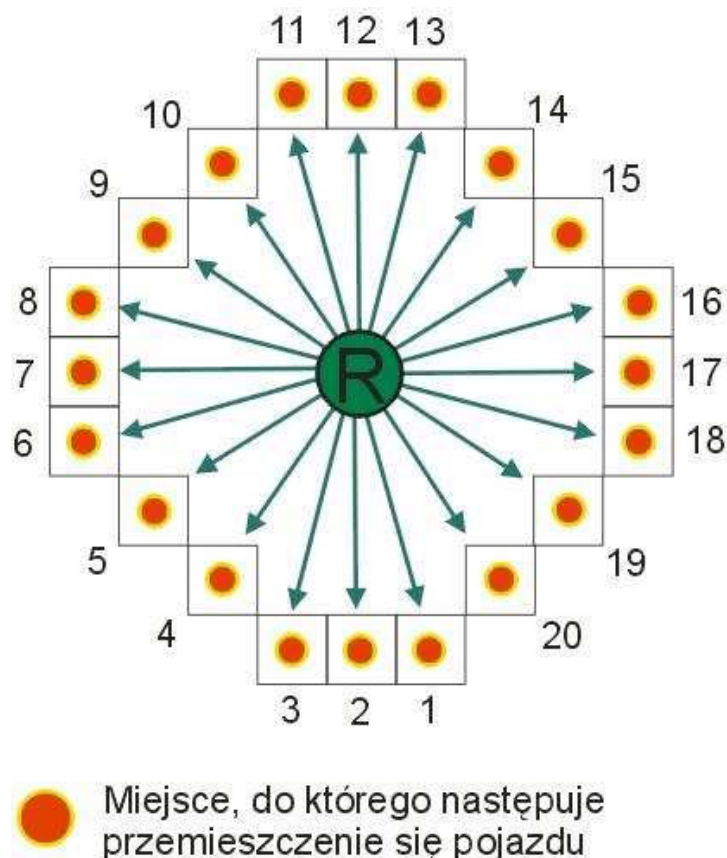
Stopień pewności (współczynnik pewności) określa jak pewne jest to, że zbadany obszar (w uogólnionej postaci jest to kwant powierzchni) zawiera tzw. element graniczny – przeszkodę lub ścianę, czy też jej nie zawiera. Im większy stopień pewności tym prawdopodobieństwo, iż badany obszar jest zbliżony do rzeczywistości dąży do jedności.

4.4.4 Schemat modelowania ruchu i błędów pozycjonowania

4.4.4.1 Wykonanie ruchu

Program zrealizowany jest iteracyjnie tj. krok po kroku. Za każdym krokiem następuje określenie możliwego położenia i jeżeli nie ma żadnych przeciwwskazań następuje wykonanie ruchu

Samo poruszanie pojazdu odbywa się w sposób ciągły – nie ma możliwości przemieszczania się w dowolny punkt mapy bez uprzedniego zbadania miejsca, które może być nową lokalizacją robota. Będzie to położenie wybrane spośród szeregu możliwych stanów (wyselekcjonowanie następuje na podstawie algorytmu ruchu) i mieszczące się w zakresie „dozwolonych odległości” (jak przedstawione zostało w podrozdziale dotyczącym modelu pojazdu symulacyjnego). Ruch będzie się więc odbywał w sposób niejako skokowy, gdzie przyrost jest określony przez zmienne ruchu – odpowiadające domyślnie położeniom kierunkowym pojazdu zawartymi w przyjętym szablonie podejmowanych decyzji (akcji).



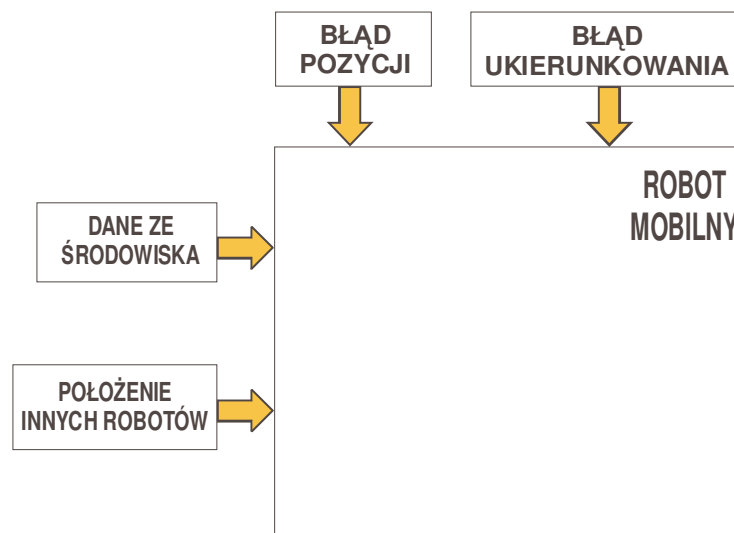
Rys. 31 Szablon modelowania ruchu

4.4.4.2 Błędy pozycjonowania

W rzeczywistych aplikacjach, w których wykorzystywane są urządzenia mobilne, nieodłącznym aspektem podczas wykonywania wielu akcji jest powstawanie różnego rodzaju błędów. Obarczone mogą być nimi:

- odczyty z czujników;
- proces określania lokalizacji;
- ruch pojazdu (np. ześlizgiwanie się kół i błąd przy odometrii)

Istnieje wiele metod, które pozwalają na niwelowanie tych niedoskonałości i zapobieganie zbyt dużym błędom. Jednak takie postępowanie nie zawsze przynosi zamierzone efekty. Dlatego w projektowanej symulacji należy wprowadzić pewne błędy, które będą pozwalały dostatecznie odwzorować układy rzeczywiste.



Rys. 32 Schemat blokowy danych zewnętrznych, które wpływają na obliczenia i szacunki przeprowadzane przez robota.

Dane ze środowiska to dane uzyskane od czujników.

Położenie robotów jest znane z uwagi na przechowywanie informacji we wspólnej pamięci. Określenie nowej pozycji robota odbywa się z wykorzystaniem tychże danych pochodzących właśnie od innych jednostek biorących udział w symulacji.

Błąd pozycji:

Ten rodzaj błędu określany jest przy pomocy wskaźników, które mogą zostać ustawione „ręcznie” w programie. Odnoszą się one do położenia centralnego punktu pojazdu na siatce rastrowej definiowanej przez współrzędne mapy dwuwymiarowej.

Δx	Błąd pozycjonowania względem osi X
Δy	Błąd pozycjonowania względem osi Y

Błędy te będą powodowały to, iż pojazd będzie mimowolnie przemieszczał się po rzeczywistej mapie, w strony określone osiami X oraz Y. Jednak same odwzorowanie w nowo tworzonej przestrzeni będzie mogło przybierać inne wartości niż te w badanym obszarze:

$A(x_i, y_i) \neq B(x_i, y_i)$	
A – rastrowa mapa badana x_i, y_i – współrzędne komórek siatki rastrowej	B – rastrowa mapa tworzona x_i, y_i – współrzędne komórek siatki rastrowej

Wprowadzanie zakłóceń ruchu może odbywać się na dwa sposoby:

- 1) systematycznie
- 2) niesystematycznie

Błędy systematyczne są wprowadzane do procesu w momencie wyznaczania kolejnych położenia pojazdu. Jak sama nazwa wskazuje mają one charakter regularny i występują co pewien okres. W rzeczywistych aplikacjach mogą one być wyeliminowane w trakcie wykonywania eksperymentów poprzez odpowiednią kalibrację pojazdu, jednak samo ich oszacowanie oraz poznanie skali występowania pomaga w takiej korekcji. Tak więc wprowadzenie zakłóceń w symulacji pomaga zobrazować w jaki sposób niedoskonałości urządzeń (biorąc pod uwagę systematyczność występowania błędów) mogą wpłynąć na odwzorowanie terenu.

W programie przyjęta jest zależność pomiędzy wprowadzaniem błędów, a czasem symulacji, która pozwala na wyznaczenie wielkości i jednocześnie częstości takich zakłóceń.

$$\Delta x_P(k) = \Delta x_P(k-1) + G_{BX}$$

$$\Delta y_P(k) = \Delta y_P(k-1) + G_{BY}$$

Δx_P , Δy_P – wartość pośrednie błędów;

G_{BX} , G_{BY} – współczynniki błędów;

k – współczynnik kroku dla czasu symulacji;

Wartości współczynników błędów zawierają się w przedziale:

$$G_{BX} = \{ -3, -2, -1, 0, +1, +2, +3 \};$$

$$G_{BY} = \{ -3, -2, -1, 0, +1, +2, +3 \};$$

Same błędy wprowadzane bezpośrednio do algorytmu ruchu będą kumulacją poprzednich zniekształceń, a wartości bezwzględne $|G_{BY}|$, $|G_{BX}|$ będą przedstawiały sobą częstotliwość z jaką będą się one pojawiać. Znak „+” lub „-” oznaczać będzie w którą stronę następować będzie odchylenie od położenia rzeczywistego.

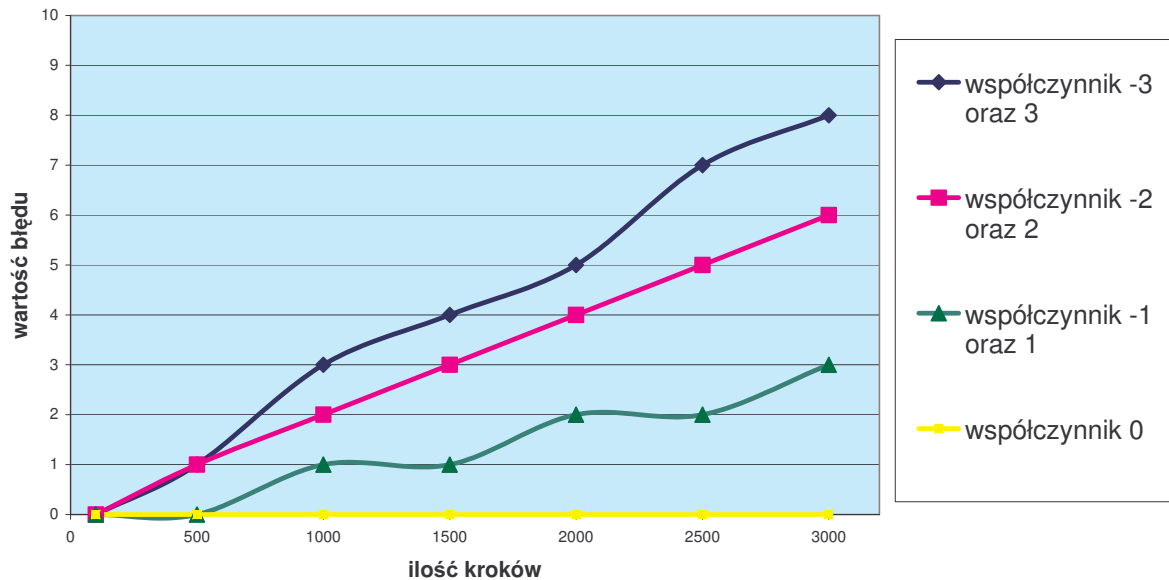
$|-3|=|3| \rightarrow$ odpowiada częstości pojawiania się zakłóceń co 334 cykli (kroków);

$|-2|=|2| \rightarrow$ odpowiada częstości pojawiania się zakłóceń co 500 cykli (kroków);

$|-1|=|1| \rightarrow$ odpowiada częstości pojawiania się zakłóceń co 1000 cykli (kroków);

$0 \rightarrow$ nie wprowadza do symulacji błędów pozycjonowania.

Zależności przedstawiające częstotliwości pojawiania się błędów oraz ich wartości



Rys. 33 Rysunek przedstawiający zależności pomiędzy czasem symulacji i ilością błędów dla zakłóceń systematycznych

Błędy niesystematyczne są wprowadzane losowo. Następuje to z wykorzystaniem funkcji prawdopodobieństwa i tak jak przy poprzednich założeniach, współczynników charakteryzujących częstość ich występowania. Nie ma tu jednak narzuconej sztywnej reguły określającej co ile kroków takie zakłócenie wystąpi. Jest to ściśle uzależnione od funkcji prawdopodobieństwa.

Wartości prawdopodobieństwa zostały przyjęte na podstawie przeprowadzonych symulacji:

$|-3|=|3| \rightarrow$ odpowiada prawdopodobieństwu 0.003;

$|-2|=|2| \rightarrow$ odpowiada prawdopodobieństwu 0.002;

$|-1|=|1| \rightarrow$ odpowiada prawdopodobieństwu 0.001;;

0 $\rightarrow$ nie wprowadza do symulacji błędów pozycjonowania.

Poniższy wykres przedstawia przykładową zależność pomiędzy wielkością błędów i czasem eksploracji dla różnych wartości prawdopodobieństwa wystąpienia różnych zakłóceń. Należy w tym miejscu podkreślić, iż każdorazowe przeprowadzenie symulacji może dawać w tym przypadku różne wyniki zważając na charakter probabilistyczny zaburzeń.

Zależności przedstawiające częstości pojawiania się błędów oraz ich wartości



Rys. 34 Rysunek przedstawiający przykładowe zależności pomiędzy czasem symulacji i ilością błędów dla pojawiających się zakłóceń niesystematycznych

Błąd ukierunkowania

Błąd kierunkowania musi zostać wprowadzony, aby w dostateczny sposób zobrazować i oddać pełną postać aplikacji robotów mobilnych. Jest to konieczne z uwagi na niedoskonałość urządzeń w rzeczywistych układach – szczególnie tych mało skomplikowanych, których algorytmy pozycjonowania posiadają dość znaczne marginesy błędów.

Modelowany jest on poprzez zmianę ukierunkowania pojazdu wskutek czego odczyty nie są prawidłowe i odwzorowywana mapa posiada liczne błędy. W symulacji należy wybrać jedną z możliwych wartości:

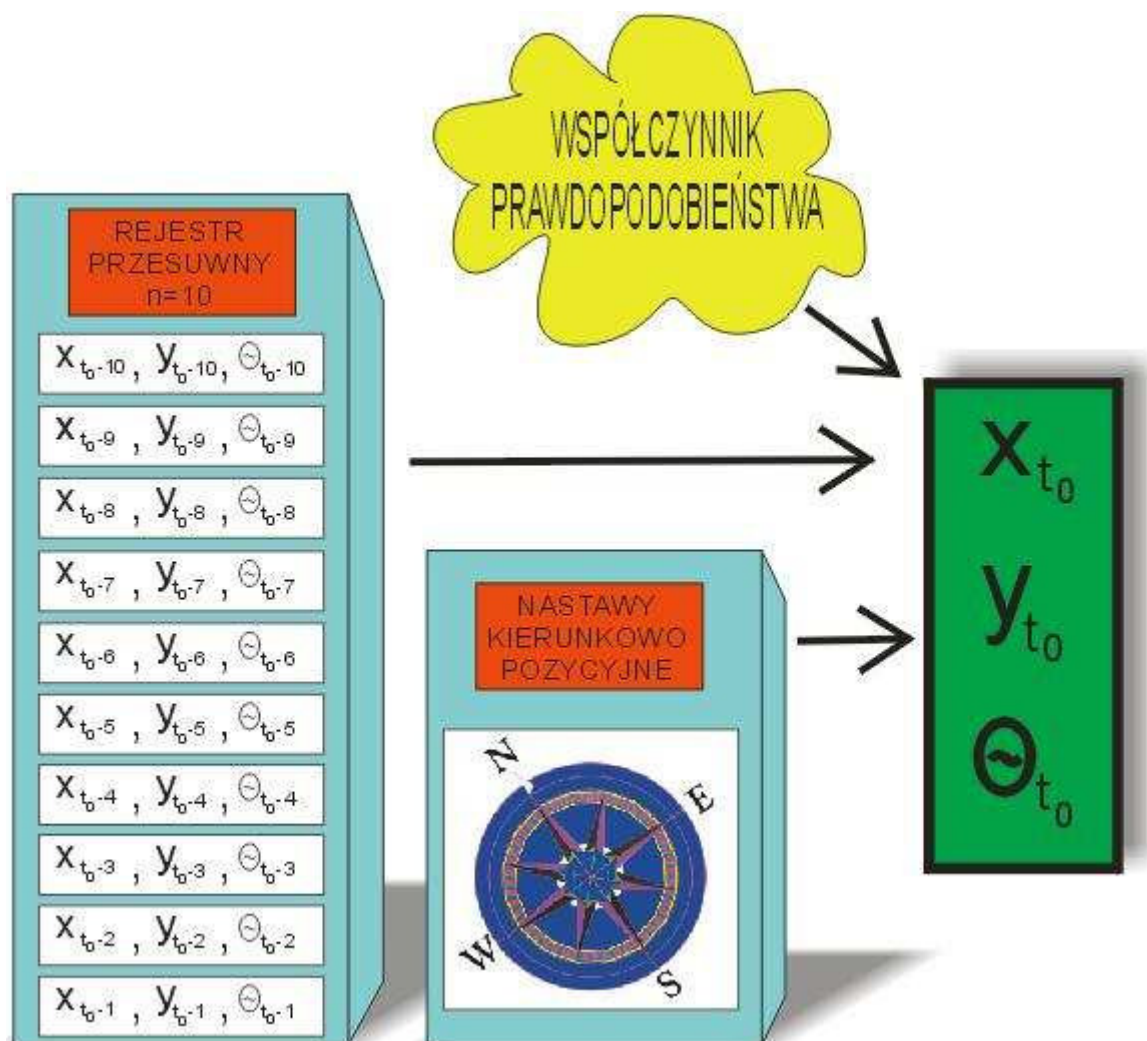
- 0 – brak błędu ukierunkowania;
- 1 – zmiana kierunku o 18 stopni w stronę zgodną z ruchem wskazówek zegara;
- +1 – zmiana kierunku o 18 stopni w stronę przeciwną do ruchu wskazówek zegara;

Uwarunkowania istniejące dla przedstawionego rozwiązania sprawiają, że odwzorowanie będzie w znacznej mierze odbiegało od rzeczywistej mapy. Tak jak we wcześniej opisanej metodzie badania zakłóceń UMBmark, odczyty ze środowiska nie będą odpowiadały faktycznemu ukierunkowaniu pojazdu.

4.4.6 Współdziałanie jednostek i algorytm ruchu

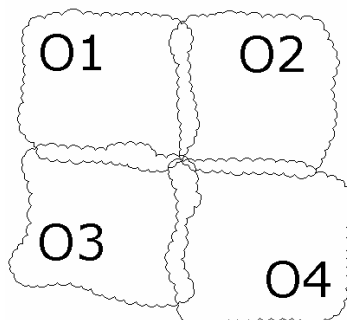
Algorytm poruszania polega na wyznaczaniu kolejnych położeń pojazdu na mapie, przy wykorzystaniu następujących danych:

- pochodzących z czujników;
- informacji o położeniach innych robotów;
- poprzednich lokalizacji pojazdu;
- współczynników kierunkowych (dla wyboru kierunku);
- wyznacznika w reprezentowanego przez funkcję prawdopodobieństwa;



Rys. 35 Elementy wpływające na wybór sposobu poruszania się robota w środowisku

Całkowity, znany użytkownikowi obszar, podzielony został na części, które są przeszukiwane przez poszczególne jednostki. Jeśli określone pole będzie badane przez kilka robotów, to niektóre z przypisanych im fragmentów będą niejako „zazębiały” się tworząc całkowity obraz dwuwymiarowej przestrzeni.



Rys.36 Pokrywanie kilku obszarów przez grupę robotów

Obszary te, to zbiory komórek, które składają się na całość siatki rastrowej.

Granice odkrytych obszarów można określić jako zbiór możliwych rozwiązań dla położenia czujników spośród wszystkich momentów symulacji:

$$O_{Ri} = \sum_{j=0}^k Z_{Ci}(j) \quad \text{gdzie} \quad Z_{Ci} \in O_G([x_G], [y_G])$$

gdzie:

O_{Ri} – zbadany obszar;

Z_{Ci} – zbiór klastrów, które pokrywają czujniki;

O_G – obszar odwzorowania czujników;

Aktualne położenie pojazdu:

$$P_{Ri}(x_i, y_i, k) = f_{RNDi}(d_R, w_k, O_G, X_{Ri})$$

$$X_{Ri} = [x_i, y_i]_{10 \times 2}$$

gdzie:

d_R – losowy współczynnik kierunkowości;

w_k – współczynnik wagowy kierunkowości;

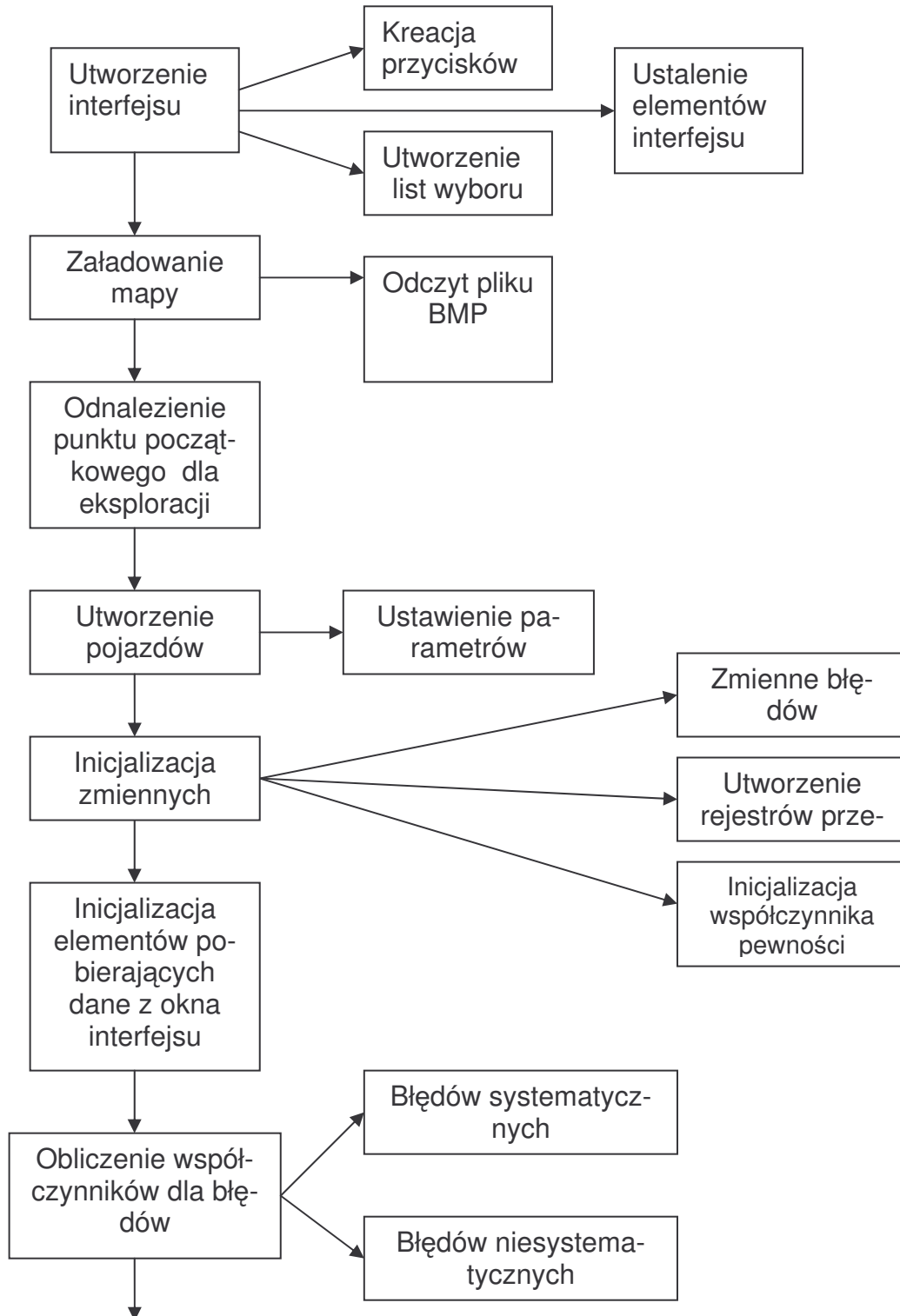
X_{Ri} – rejestr przesuwany przechowujący informację o 10-ciu poprzednich położeniach;

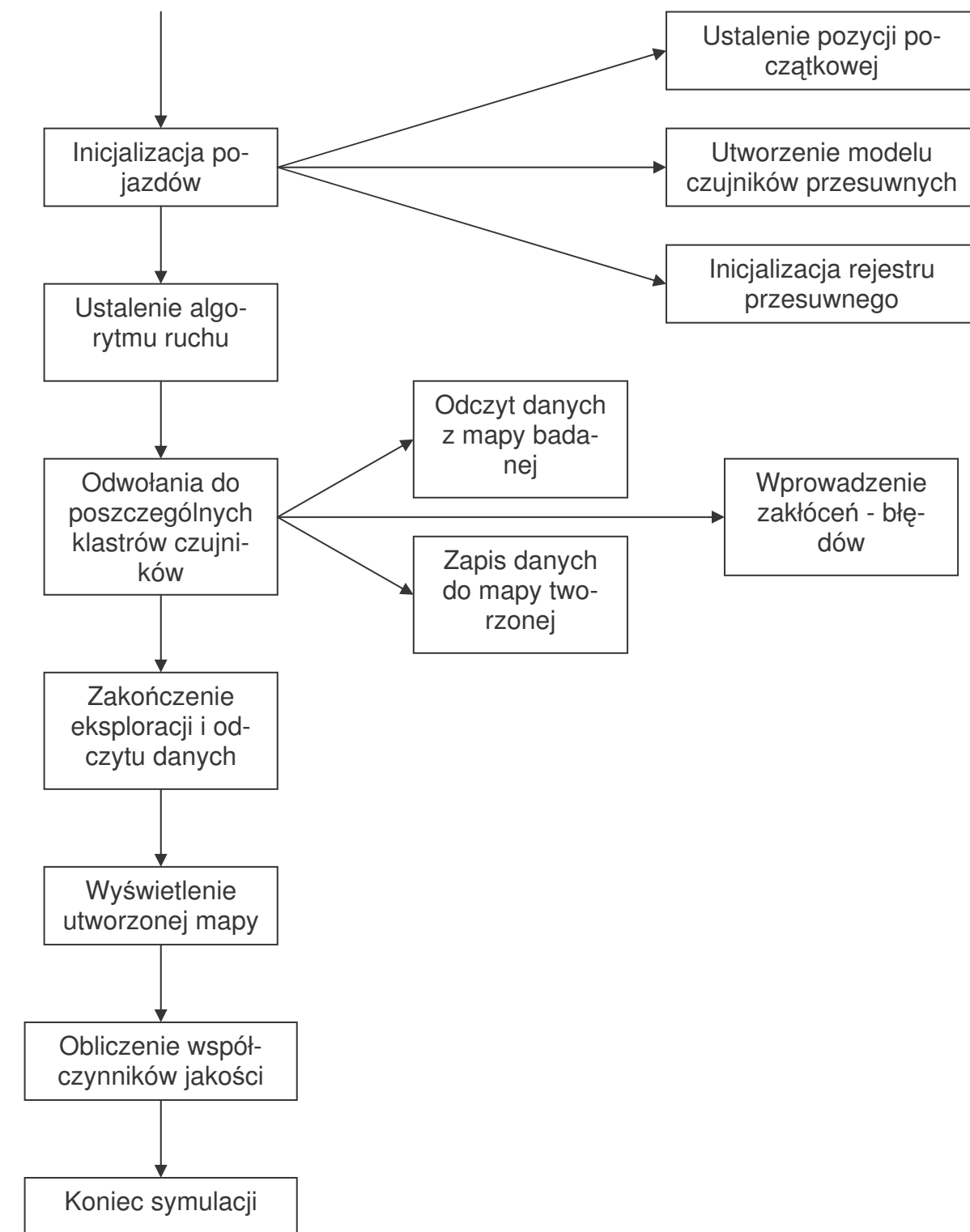
x_i, y_i – położenie i-tego pojazdu;

Jak przedstawione zostało w poprzednich rozdziałach, komunikacja jednostek w grupie może przybierać przeróżne formy. Małe jednostki posiadają z reguły ograniczone możliwości odnośnie porozumiewania się i polegają one przede wszystkim na niezbyt skomplikowanych, choć czasami bardzo skutecznych, mechanizmach.

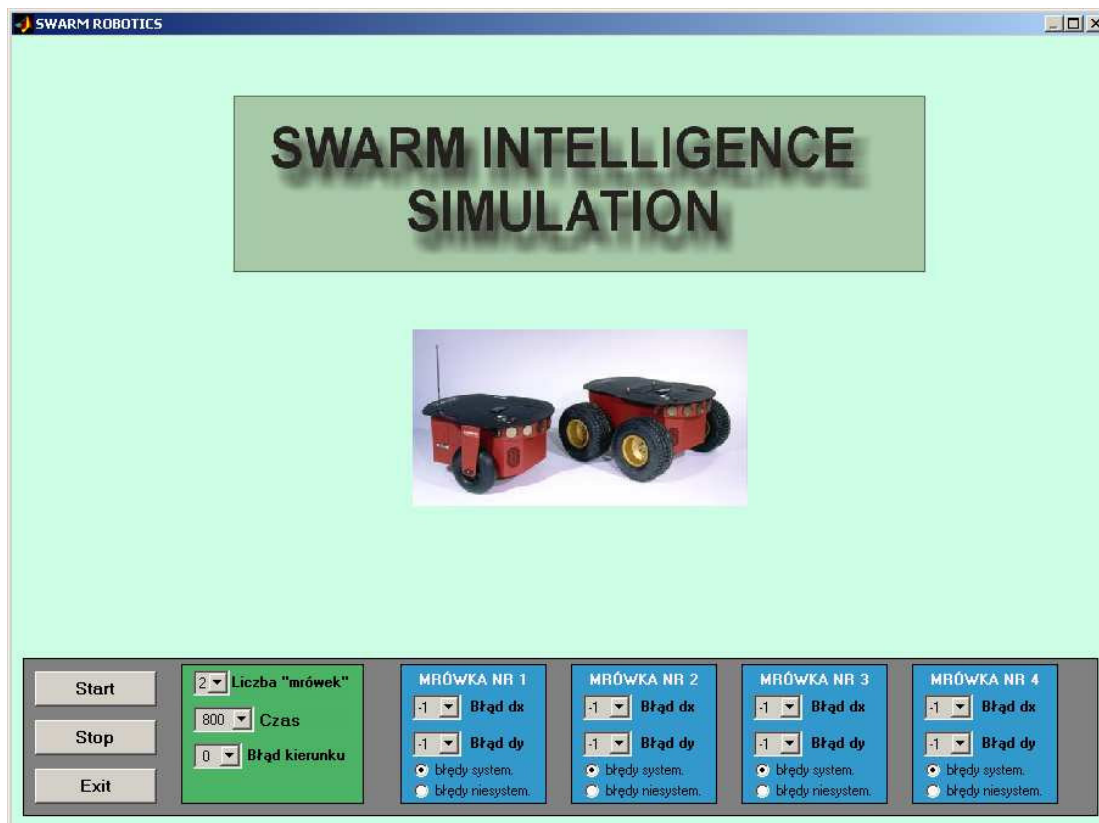
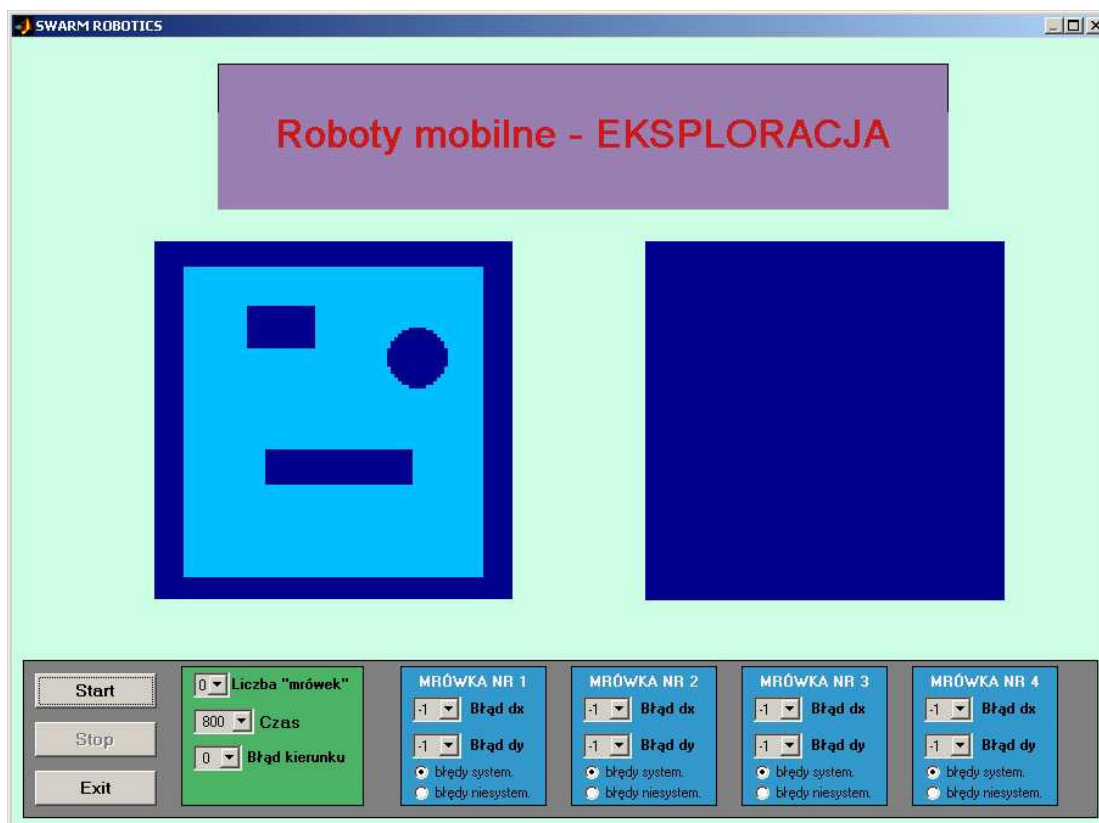
W badanej symulacji współdziałanie pojazdów polega przede wszystkim na przechowywaniu danych o odkrytym „terytorium” we wspólnej pamięci. Każdy z pojazdów będzie pozostawiał każdorazowo w miejscu swojego pobytu informację mówiącą o tym, że dana komórka mapy rastrowej była już badana. Ta sama informacja może być odczytana i uaktualniana przez tą samą lub inną jednostkę.

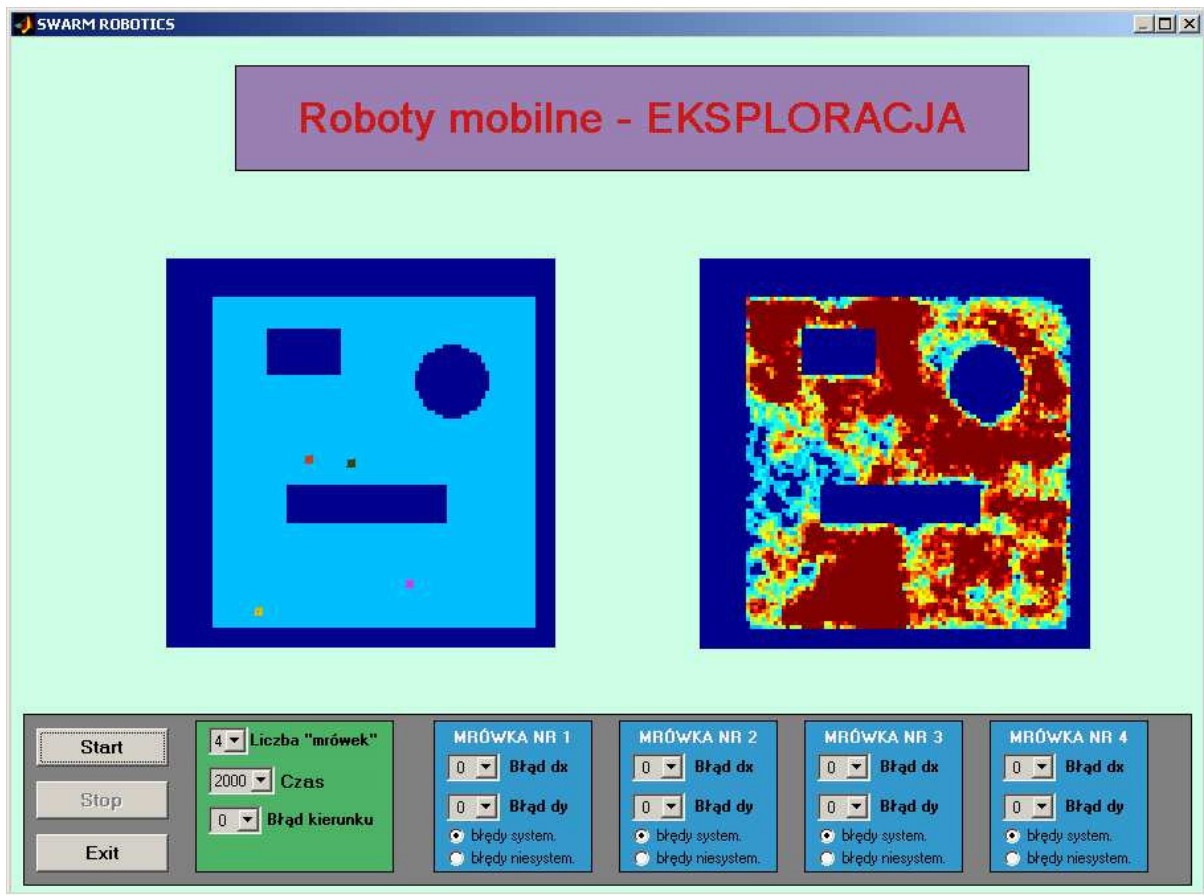
4.4.7 Schemat blokowy programu i interfejs graficzny





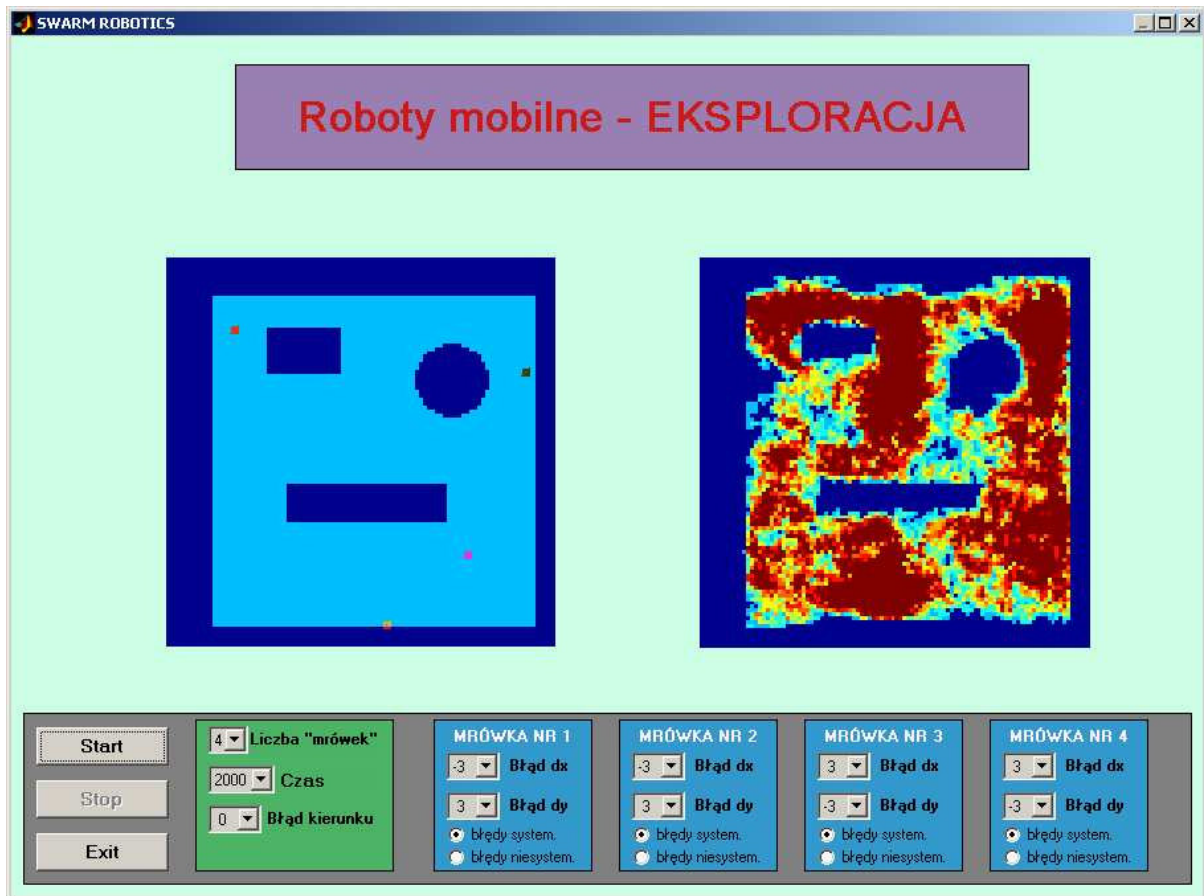
Powyższy schemat przedstawia elementy reprezentujące kolejne części kodu programu (symulacji). Wszystkie jego składowe zawarte są w kilku plikach pakietu MATLAB i pozwalają na przeprowadzenie prób eksploracji nieznanego terenu przez grupę robotów. Pliki te zawarte są w załączniku do pracy – na nośniku CD.

Ekran uruchomienia programu:Ekran przed przeprowadzeniem symulacji:

Przeprowadzona symulacja dla określonych parametrów:

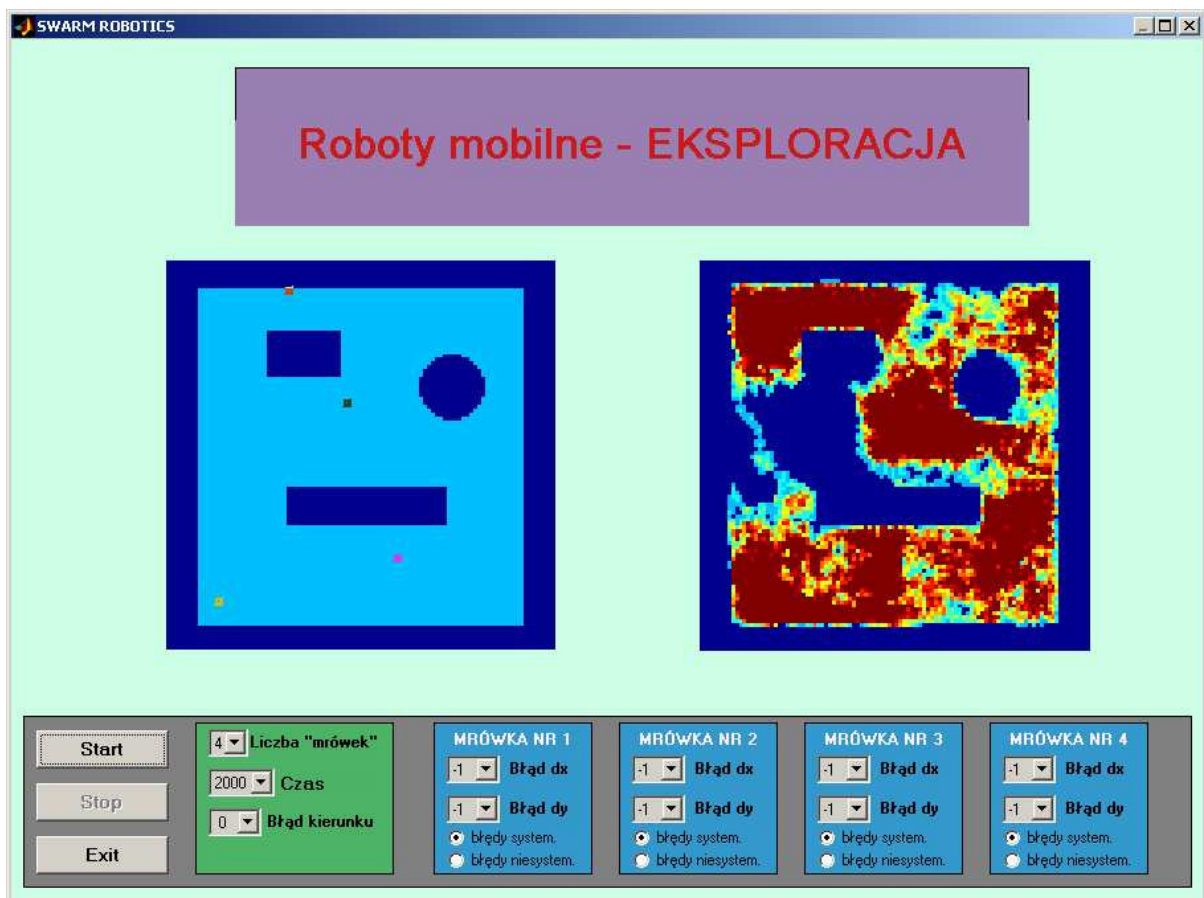
4	liczba pojazdów
2000	liczba iteracji
89,41%	współczynnik odkrycia
100 %	współczynnik dokładności

Błąd	MRÓWKA 1		MRÓWKA 2		MRÓWKA 3		MRÓWKA 4	
	dx	dy	dx	dy	dx	dy	dx	dy
Systematyczny	0	0	0	0	0	0	0	0
Niesystematyczny	-	-	-	-	-	-	-	-
Kierunku	0		0		0		0	



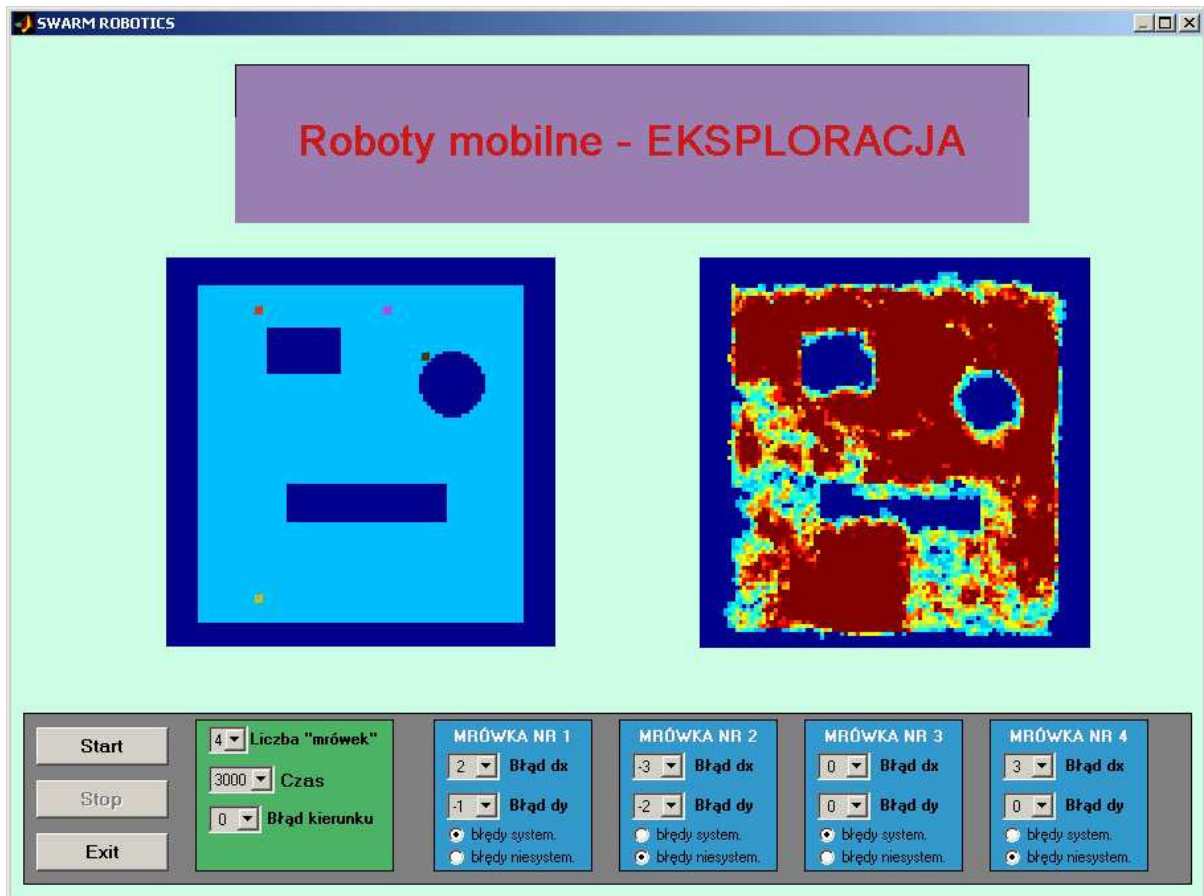
4	liczba pojazdów
2000	liczba iteracji
96,11%	współczynnik odkrycia
91,66 %	współczynnik dokładności

Błąd	MRÓWKA 1		MRÓWKA 2		MRÓWKA 3		MRÓWKA 4	
	dx	dy	dx	dy	dx	dy	dx	dy
Systematyczny	-3	3	-3	3	3	-3	3	-3
Niesystematyczny	-	-	-	-	-	-	-	-
Kierunku	0		0		0		0	



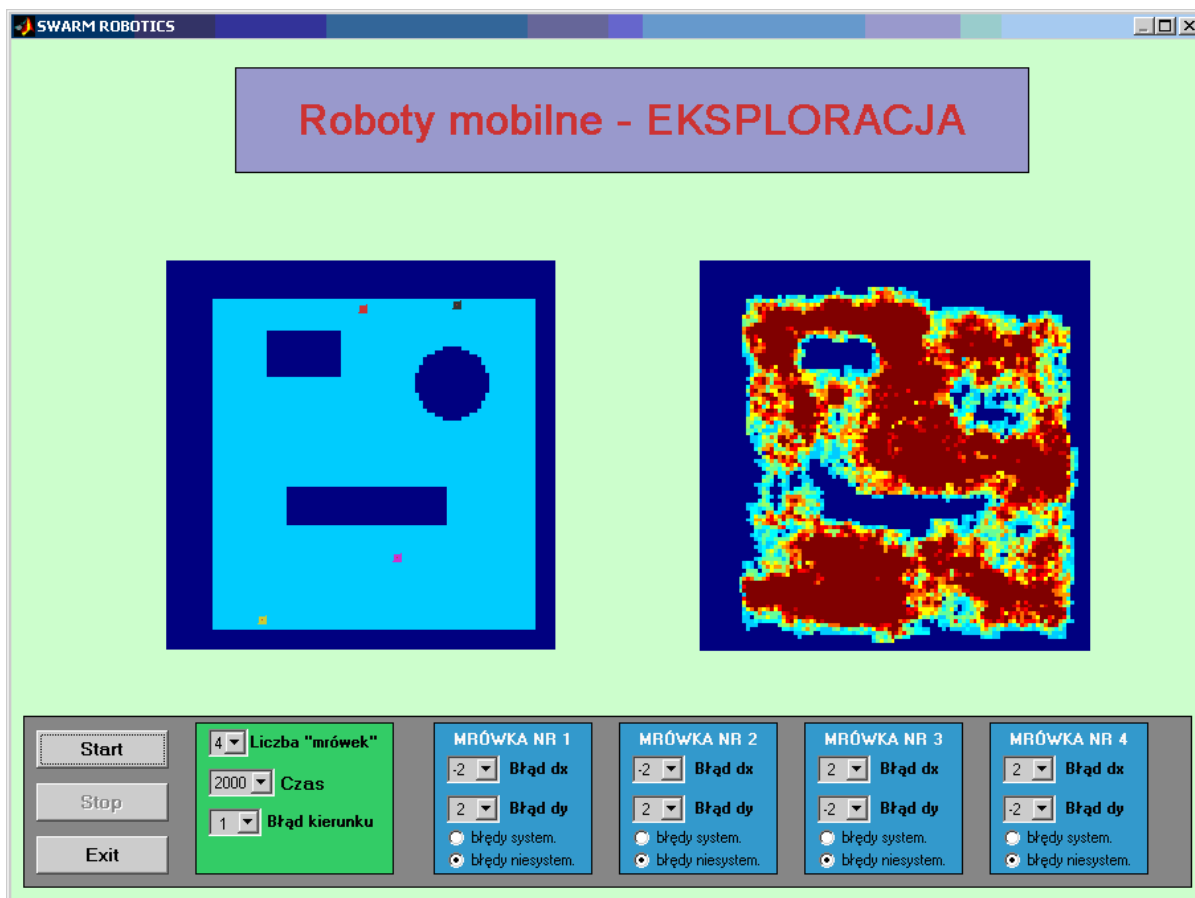
4	liczba pojazdów
2000	liczba iteracji
83,27%	współczynnik odkrycia
96,21 %	współczynnik dokładności

Błąd	MRÓWKA 1		MRÓWKA 2		MRÓWKA 3		MRÓWKA 4	
	dx	dy	dx	dy	Dx	dy	dx	dy
Systematyczny	-1	-1	-1	-1	-1	-1	-1	-1
Niesystematyczny	-	-	-	-	-	-	-	-
Kierunku	0		0		0		0	



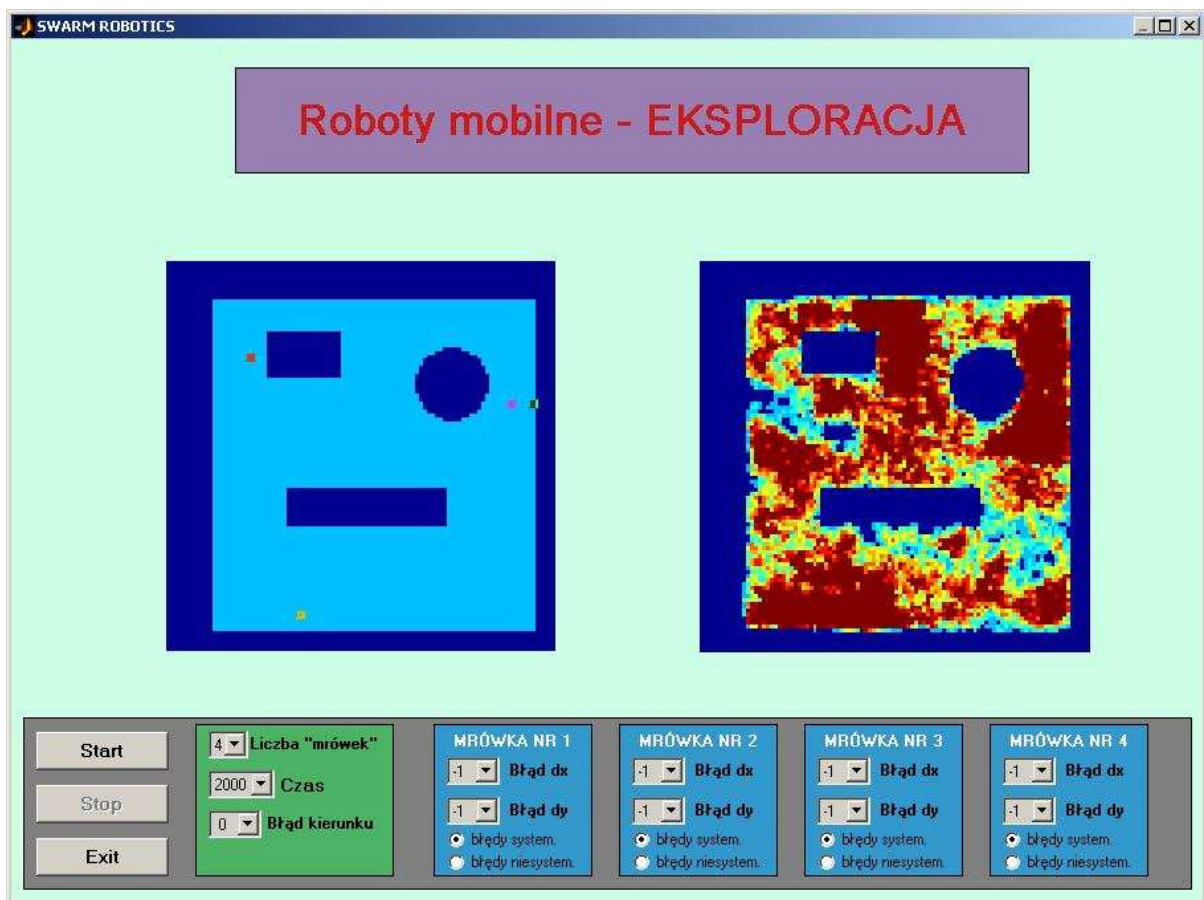
4	liczba pojazdów
3000	liczba iteracji
103,14 %	współczynnik odkrycia
89,87 %	współczynnik dokładności

Błąd	MRÓWKA 1		MRÓWKA 2		MRÓWKA 3		MRÓWKA 4	
	dx	dy	dx	dy	dx	dy	dx	dy
Systematyczny	2	-1	-	-	0	0	-	-
Niesystematyczny	-	2	-3	-2	-	-	3	0
Kierunku	0		0		0		0	



4	liczba pojazdów
2000	liczba iteracji
98,06%	współczynnik odkrycia
91,92 %	współczynnik dokładności

Błąd	MRÓWKA 1		MRÓWKA 2		MRÓWKA 3		MRÓWKA 4	
	dx	dy	dx	dy	dx	dy	dx	dy
Systematyczny	-	-	-	-	-	-	-	-
Niesystematyczny	-2	2	-2	2	2	-2	2	-2
Kierunku	1		1		1		1	



4	liczba pojazdów
2000	liczba iteracji
99,51 %	współczynnik odkrycia
86,18 %	współczynnik dokładności

Błąd	MRÓWKA 1		MRÓWKA 2		MRÓWKA 3		MRÓWKA 4	
	dx	dy	dx	dy	dx	dy	dx	dy
Systematyczny	-1	-1	-1	-1	-1	-1	-1	-1
Niesystematyczny	-	-	-	-	-	-	-	-
Kierunku	0		0		0		0	

Aby określić jakość symulacji zostały wprowadzone wskaźniki, które pozwalają na zobrazowanie zależności pomiędzy różnymi ustawieniami niektórych parametrów.

$K_{\%}$ - procentowy współczynnik odkrycia (zbadania) terenu – przedstawia on zależność pomiędzy „zawartością” rzeczywistej mapy (która jest znana użytkownikowi), a mapą już odkrytą – jest to wartość podana w procentach i określa jaka część mapy została odkryta (poznana).

$$K_{\%} = \frac{O_{GB}(x_i, y_i)}{O_{GA}(x_i, y_i)} \cdot 100\%$$

O_{GB} – Zbiór punktów (pole) mapy tworzonej;

O_{GA} – Zbiór punktów (pole) mapy badanej;

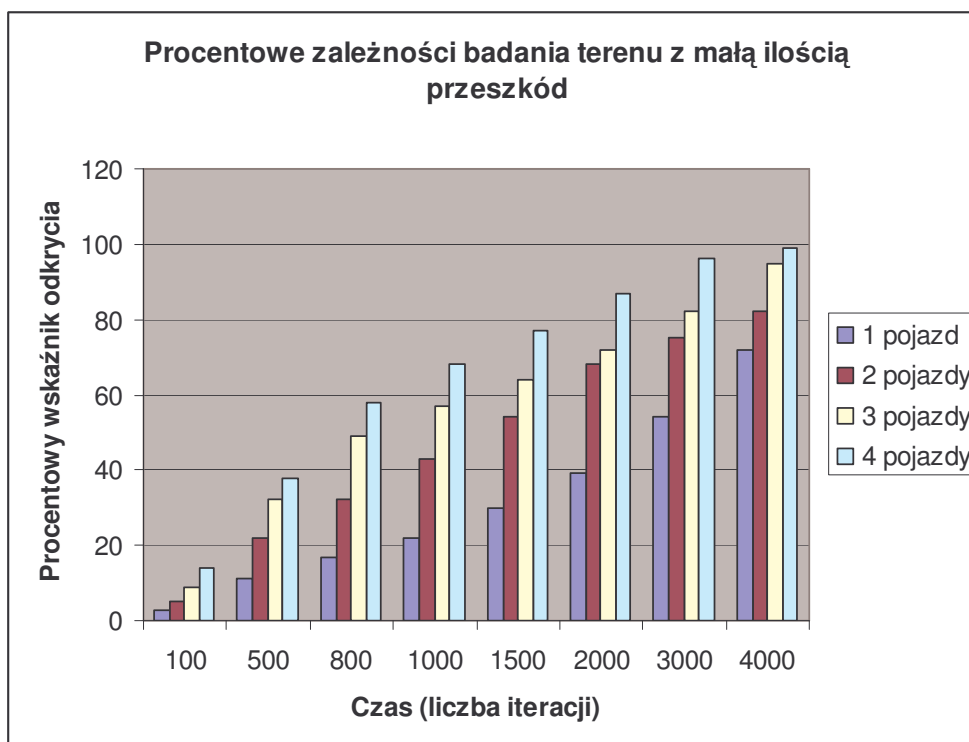
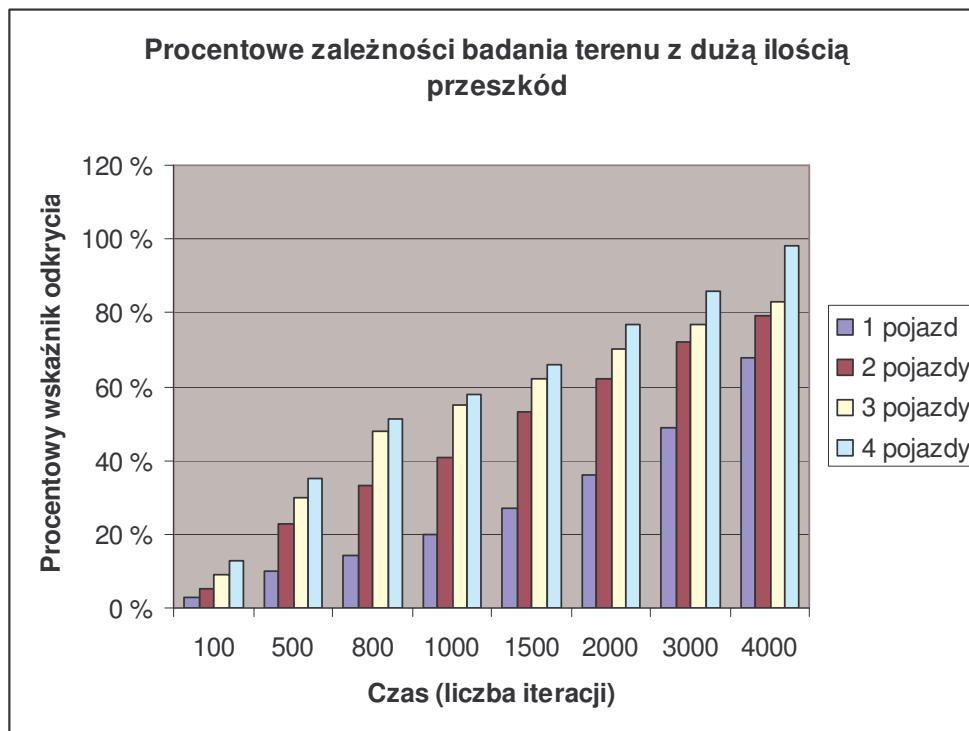
W przeprowadzonych symulacjach jego wartość zmienia się w granicach od 83,27% aż do 103,14%. O ile pierwsza z tych wielkości odpowiada szczególnemu przypadkowi, gdzie eksploracja mapy nie została przeprowadzona w optymalny sposób (część terenu nie została zbadana pomimo dosyć długiego czasu symulacji) to jednak druga z nich może rodzić pytania. Teoretyczna, maksymalna wielkość współczynnika odkrycia dla zamkniętego terenu nie może być większa niż 100%. Tutaj ma on wartość ponad 103% co jest spowodowane znacznymi błędami pozycjonowania pojazdów.

K_j - procentowy współczynnik dokładności odwzorowania – przedstawia on zależność pomiędzy rzeczywistą mapą, a mapą utworzoną;

$$K_{\%} = \sum_{i,j=0}^{\text{rozmiar}(mapa)} \frac{P_B(x_{Bi}, y_{Bi})}{P_A(x_{Ai}, y_{Ai})} \cdot 100\%$$

P_B – Punkty mapy tworzonej;

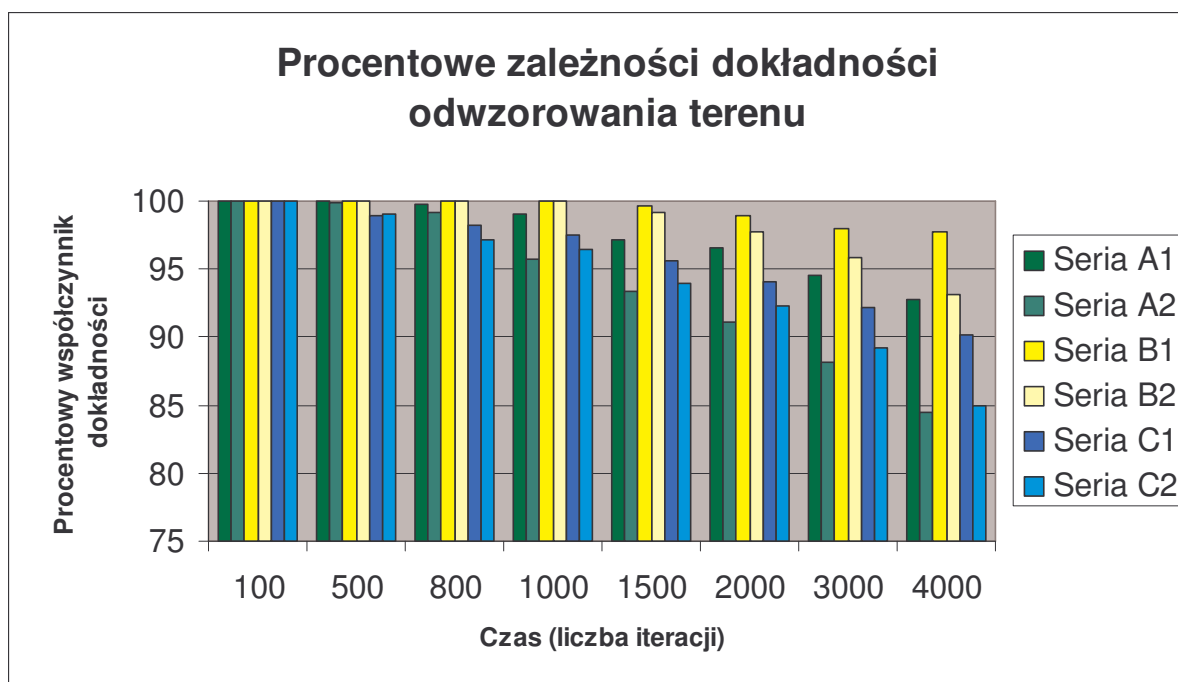
P_A – Punkty mapy badanej;



Rys. 37 Wykresy przedstawiające zależności procentowego wskaźnika odkrycia powierzchni w zależności od czasu badania oraz ilości pojazdów biorących udział w symulacji.

Powyższe wykresy zostały utworzone na podstawie wyników uzyskanych z przeprowadzonych symulacji. W badaniu tym, zmieniały się takie parametry jak liczba pojazdów i czas procesu eksploracji. Jak widać wyraźnie wskaźnik odkrycia powierzchni wzrasta wraz z liczbą robotów i czasem symulacji – większe wartości uzyskuje się również przy zmianie zajętości mapy przez przeszkody (im mniej utrudnień tym większy obszar jest zbadany). W celu uniknięcia zbyt dużych niedokładności odwzorowania, błędy pozycjonowania są równe zeru.

Następnym wskaźnikiem na jaki należy zwrócić uwagę jest wskaźnik dokładności odwzorowania terenu „Kj”. Zależności przedstawiające jego wielkość dla zmiennych wartości błędów zaprezentowane są poniżej:



Rys. 38 Wykresy przedstawiające zależności procentowego wskaźnika dokładności w zależności od czasu badania oraz błędów pozycjonowania pojazdów.

	Seria A1	Seria A2	Seria B1	Seria B2	Seria C1	Seria C2
dx	-2	-2	1	1	-3	-3
dy	2	2	-1	-1	0	0
teta	0	-1	0	-1	0	-1

Otrzymane wyniki ukazują, że im większe niedokładności pojazdów biorących udział w symulacji to dokładność w odwzorowaniu terenu maleje. Wraz z rosnącym

czasem badań wartość współczynnika dokładności również maleje (od 100% nawet do wartości poniżej 85%) co jest spowodowane kumulacją błędów pozycjonowania pojazdów.

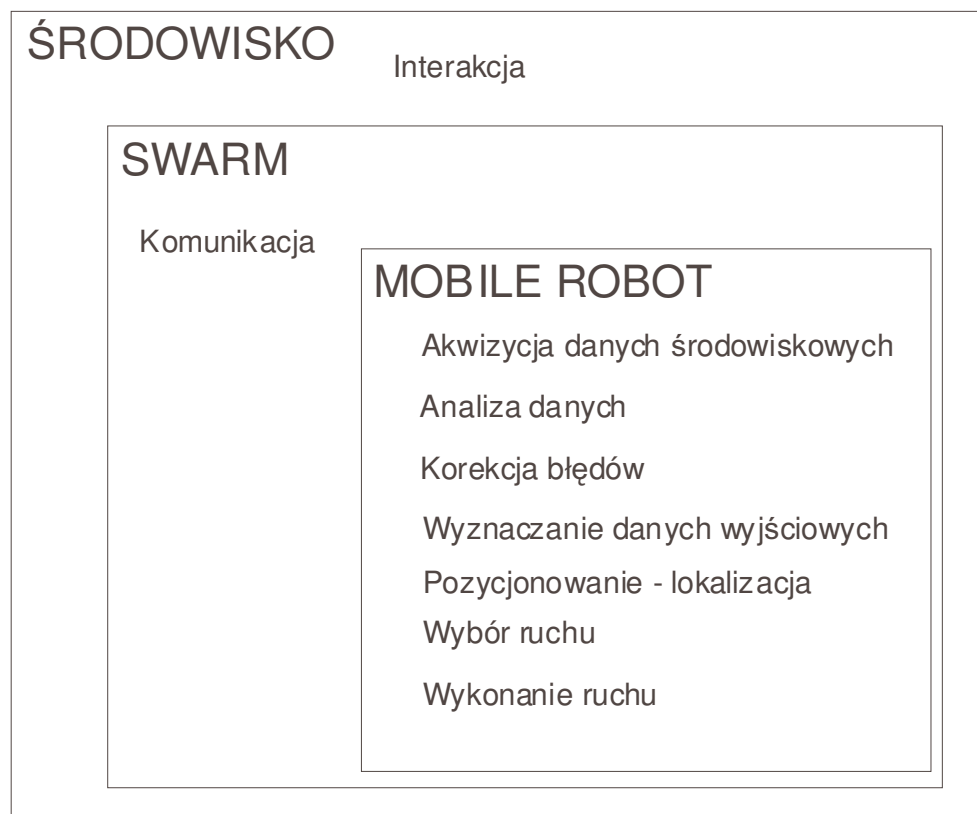
W przedstawionych wynikach ujęto te dane, które uzyskano dla jednakowych ustawień błędów we wszystkich pojazdach – uzyskuje się dzięki temu rezultaty odpowiadające grupom urządzeń homogenicznych.

Widać tu wyraźnie, iż dla większych niedoskonałości pojazdów otrzymuje się gorsze odwzorowanie terenu. Niedokładności w kreowaniu wirtualnej mapy wzrastają wraz z czasem eksploracji, co jest spowodowane kumulacją błędnych odczytów informacji ze środowiska.

5. Uwagi końcowe i wnioski

Opracowana symulacja pozwala na przeprowadzenie testów, które ukazują charakterystyczne elementy eksploracji nieznanego terenu przez grupę robotów mobilnych. Zostały tu przedstawione niezbyt skomplikowane, wirtualne pojazdy mogące zmieniać swoje położenie w terenie i odczytywać informacje z najbliższego otoczenia. Otrzymane wyniki pozwalają na oszacowanie jakościowe badań dotyczące budowania wirtualnej mapy środowiska. Różne konfiguracje, jakkolwiek dotyczące tylko podstawowych parametrów aplikacji, pomóc mogą w wyobrażeniu sobie, w jaki sposób następuje eksploracja nieznanymi obszarów. Takie przedstawienie powyższego zagadnienia nie daje jednak wyczerpujących informacji na temat zachowań robotów w grupie i nie pozwala na dostateczne zbadanie bardziej skomplikowanych terenów (jak np. labirynt). Bardziej zaawansowane techniki potrzebne są do pozycjonowania dużej grupy pojazdów, ich wzajemnej komunikacji i eksploracji obszarów w o wiele gorszych warunkach (bardziej zbliżonych do uwarunkowań rzeczywistych). Konkluzja z zastosowanego podejścia dawać może zobrazowanie, jakie czynniki mają fundamentalne znaczenie w funkcjonowaniu robotów mobilnych w badaniu nieznanymi obszarów. Ulepszenia zastosowanych w symulacji funkcji są nieodzowną częścią dalszego postępowania, by w miarę dobrze móc odwzorować rzeczywistość. Badania nad gotowymi układami (nie tylko wirtualnymi) muszą zostać przeprowadzone wcześniej w bardzo dokładny sposób, aby dokonać szacunków i ujawnić mogące pojawić się problemy.

Wnioski wyciągnięte z przeprowadzonych analiz i badań pozwalają stwierdzić, iż niezbędne jest opracowanie modeli interakcji oraz zależności pomiędzy elementami składowymi systemów. Schemat blokowy, przedstawiający istniejące relacje wśród komponentów aplikacyjnych może być zaprezentowany w następujący sposób:



Dalsza praca może być prowadzona w kierunku optymalizacji algorytmu poruszania się pojazdów w terenie, ich wzajemnej komunikacji, modelowaniu i eliminacji błędów. Jednak należy zwrócić uwagę na złożoność rzeczywistych urządzeń i ich niedoskonałość. Jest bowiem oczywistym faktem to, że przeprowadzanie symulacji nie w pełni będzie obrazować otaczający świat oraz zachodzące w nim procesy i zdarzenia. Należy zatem zawsze przeprowadzać eksperymenty na aplikacjach wykorzystujących fizyczne urządzenia i na podstawie otrzymanych wyników podejmować dalsze kroki w ich udoskonalaniu.

LITERATURA:

- ¹ R. Brooks, "A Robust Layered Control System for a Mobile Robot": "IEEE Journal of Robotics and Automation", Nr 1/1986, s. 14-23.
- ² "Popularny słownik języka polskiego" wyd. Wilga, 1999
- ³ D.Barnes, J.Gray. "Behaviour synthesis for cooperant mobile robot control". International Conference On Control, str. 1135-1140, 1991
- ⁴ M.Mataric. "Interaction and Intelligent Behavior", MIT, EECS, 1994
- ⁵ S.Premvuti i S.Yuta "Consideration on the cooperation of multiple autonomous mobile robots", IEEE/RSJ IROS, str. 59-63, 1990
- ⁶ Y.Uny Cao, Alex S. Fukunaga, Andrew B. Kahng, „Cooperative Mobile Robotics: Antecedents and Directions” 1997
- ⁷ Strona internetowa: en.wikipedia.org
- ⁸ D. J. Bruemmer, D. D. Dudenhoeffer, J. L. Marble, „Mixed-Initiative Remote Characterization Using a Distributed Team of Small Robots”
- ⁹ D.Jung, A.Zelinsky, "Grounded Symbolic Communication between Heterogeneous Cooperating Robots", KAP, 2000
- ¹⁰ J. Borenstein, H. R. Everett, L. Feng, "Navigating Mobile Robots", 1996
- ¹¹ J. Leonard, H. Durrant-Whyte, Ingemar J.Cox, "Dynamic Map Building for an Autonomous Mobile Robot", 1990
- ¹² J. Borenstein, U. Raschke, "A comparison of Grid-type Map building Techniques by Index of Performance", 1990
- ¹³ P. Thomson, "Sposoby Komunikacji Interpersonalnej", 1998
- ¹⁴ Strona internetowa: www.activrobots.com
- ¹⁵ D. Fox, W. Burgard, H. Kruppa, S. Thurn, "Collaborative Multi-Robot Localization", 23rd German Conf. on Artificial Intelligence, 1999
- ¹⁶ R. Simmons, D.Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, H. Younes, „Coordination for Multi-Robot Exploration and Mapping”
- ¹⁷ M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, „A Factored Solution to the Simultaneous Localization and Mapping Problem”

ZAŁĄCZNIKI:

Do pracy dyplomowej dołączono płytę CD, zawierającą wersje elektroniczne pracy (w formatach plików PDF i DOC) oraz kody źródłowe – pliki środowiska MATLAB. Aby móc rozpocząć symulację należy utworzyć katalog „Swarm” na dysku „C” komputera i skopiować do niego m-pliki oraz pliki obrazów. Kolejnym i zarazem ostatnim krokiem jest uruchomienie programu głównego „**esplo.m**” przy pomocy pakietu MATLAB.

PLIK	OPIS
Praca.doc	Wersja elektroniczna pracy w formacie DOC;
Praca.pdf	Wersja elektroniczna pracy w formacie PDF;
esplo.m	Program nadrzędny;
algorytm_ruchu_01.m	Funkcja wyznaczająca współrzędne przemieszczenia;
odwołanie_h1.m	Funkcja określająca typ błędu dla obiektu h1;
odwołanie_h2.m	Funkcja określająca typ błędu dla obiektu h2;
odwołanie_h3.m	Funkcja określająca typ błędu dla obiektu h3;
odwołanie_h4.m	Funkcja określająca typ błędu dla obiektu h4;
początek_przestrzeni.m	Funkcja wyznaczająca punkt startowy dla pojazdów;
prawdopodobieństwo.m	Funkcja obliczająca skoki i ich kierunki;
ukierunkowanie_01.m	Funkcja wyznaczająca kierunki ruchu;
wybor_kierunku.m	Funkcja określająca współczynnik prawdopodobieństwa;
załadowanie_mapy.m	Funkcja wczytująca mapę środowiskową;
zrownanie_czujnikow.m	Funkcja zrównująca czujniki i ich ukierunkowanie;
mapka_01.bmp	Mapa środowiskowa;
pioneer2_at.bmp	Obraz 1 dla interfejsu;
interface.jpg	Obraz 2 dla interfejsu;