

Plan pracy magisterskiej

Maciej Gąbka

26 lipca 2011

Spis treści

1	Wstęp	4
2	Liga Robocup	5
2.1	Omówienie projektu robocup	5
2.1.1	Krótki opis genezy	5
2.1.2	Podział na ligi robotów	5
2.2	Specyfika ligi small-size (F180)	6
2.2.1	Zasady	6
2.2.2	Budowa robota	6
3	Założenia projektowe	7
4	Projekt Player/Stage/Gazebo	8
4.1	Koncepcja projektu	8
4.2	Architektura	8
4.2.1	Stage	9
4.2.2	Gazebo	9
4.2.3	Player	9
4.3	Modelowanie obiektów	9
4.4	Zasady modelowania w Gazebo 0.10	9
4.5	Realizacja środowiska ligi Robocup	9
5	Architektura aplikacji sterującej drużyną robotów	10
6	Sterowanie modelem robota z <i>Small-size League</i>	12
6.1	Omówienie omnikierunkowej bazy jezdnej	12
6.1.1	Opis położenia kół	12
6.1.2	Opis kinematyki oraz dynamiki bazy	12
6.2	Dryblowanie z piłką	12

7	Algorytmy unikania kolizji	13
7.1	Krótki przegląd algorytmów unikania kolizji	13
7.2	Zalety algorytmu RRT w stosunku do CVM	14
7.3	Szczegółowa zasada algorytmu RRT	14
7.4	Szczegóły implementacji zastosowanego algorytmu	14
7.5	Wyznaczanie prędkości prowadzącej do punktu docelowego . .	15
8	Testy algorytmu nawigacji robota	16
8.1	Zachowanie algorytmu RRT w środowisku statycznym	16
8.2	Środowisko dynamiczne	17
8.3	Prowadzenie piłki	17
9	Testy algorytmu planowania i koordynacji działań	18
9.1	Testy planowania działań na poziomie pojedynczego robota (Tactics)	18
9.2	Testy planowania działań na poziomie drużyny	18
9.3	Środowisko dynamiczne	18
10	Podsumowanie	19
11	Dodatek	20
11.1	Instrukcja konfiguracji Gazebo	20
11.2	Kod źródłowy modeli	20

Rozdział 1

Wstęp

Rozdział 2

Liga Robocup

Streszczenie

Rozdział zostanie w całości poświęcony opisowi projektu Robocup. Więcej informacji na ten temat należy poszukiwać na oficjalnej stronie projektu <http://www.robocup.org>. Zaprezentowana zostanie pokrótce historia samej idei projektu, jej rozwój oraz poszczególne ligi wchodzące w skład całego projektu. Głównie zostanie omówiona *Small-size League*, ze szczególnym uwzględnieniem modelu robota i ograniczeń z tym związanych.

2.1 Omówienie projektu robocup

2.1.1 Krótki opis genezy

2.1.2 Podział na ligi robotów

- Najstarsza z lig, liga symulacyjna
- Omówienie ligi *Small-size League*
- Omówienie ligi *Middle-size League*
- Omówienie ligi *Standard Platform League*
- Omówienie ligi *Humanoid League*
- *RoboCupRescue* - wykorzystanie dorobku robotyki w służbie ludziom

2.2 Specyfika ligi small-size (F180)

W paragrafie zostaną ujęte i dokładnie opisane zasady ligi (dzięki temu można będzie wyjaśnić rodzaj architektury zastosowanej w aplikacji sterującej). Po krótko zostanie przedstawiony model robota wykorzystywany w lidze ze szczególnym uwzględnieniem elementów służących do podawania oraz prowadzenia piłki. Dodatkowo zostanie zaprezentowany schemat komunikacji wykorzystywany w lidze.

2.2.1 Zasady

- zaprezentowanie wymogów technicznych dotyczących :
 - wymiarów robota
 - wymiarów boiska
 - liczby robotów
 - sposobów prowadzenia piłki
- omówienie podstawowych zasad takich jak:
 - sygnalizowanie przewinień
 - rola arbitra podczas rozgrywki

2.2.2 Budowa robota

- Zaprezentowanie konstrukcji robotów z ligi *small-size*
 - omówienie bazy jezdnej
 - zaprezentowanie urządzenia umożliwiającego prowadzenie piłki
 - rola znaczników umieszczonych na konstrukcji mechanicznej robota
 - dostępne czujniki (określone przez regulamin)

Rozdział 3

Założenia projektowe

Rozdział ten powinien określać założenia projektowe, którymi kierowałem się tworząc oprogramowanie, i które miały wpływ na przyjęcie konkretnych rozwiązań.

Rozdział 4

Projekt Player/Stage/Gazebo

Rozdział powinien zawierać krótki opis symulatora. Najwięcej uwagi należy poświęcić konstrukcji modeli robotów, oraz wprowadzonym poprawkom do symulatora, tj. tarcia w dwóch płaszczyznach oraz tarcia przy ruchu obrotowym.

4.1 Koncepcja projektu

Krótkie wprowadzenie - symulator open source, stworzony w celu modelowania robotów, umożliwiający symulowanie ich działania, ale także dostarczający interfejsy pozwalające na sterowanie rzeczywistymi robotami. (Licencja GPL, podać strony www oraz wiki).

4.2 Architektura

Schemat architektury systemu i przepływu informacji: Gazebo + Player + Stage + aplikacja kliencka + rzeczywisty robot. Krótki opis każdego z komponentów i jego przydatności w kontekście rozważanego zadania (symulacja ligi).

4.2.1 Stage

4.2.2 Gazebo

4.2.3 Player

4.3 Modelowanie obiektów

Krótki wstęp na temat modelowania, opisać bibliotekę stosowaną do wizualizacji, do wykrywania kolizji.

4.4 Zasady modelowania w Gazebo 0.10

Ta część rozdziału będzie opisywać ogólne zasady modelowania w symulatorze Gazebo w wersji 0.810. Zostanie pokazany schemat (krok po kroku) tworzenia prostego świata (plik .world), obiektów, połączeń między nimi, sposobu kontroli przez aplikację kliencką. Proste przykłady, mające dać ogólne pojęcie o możliwościach środowiska, bez wchodzenia w szczegóły.

4.5 Realizacja środowiska ligi Robocup

Prezentacja modeli wykonanych na potrzeby ligi: boisko oraz roboty, wspomnieć o konieczności napisania własnego kontrolera. Screeny prezentujące modele. Pełne kody źródłowe w xml opisujące modele.

Rozdział 5

Architektura aplikacji sterującej drużyną robotów

W rozdziale zostanie zaprezentowana architektura STP – Skill Tactics Play■ szerzej opisana w [10]. Podejście to zakłada istnienie następujących modułów:

1. moduł zawierający informacje o świecie,
2. moduł umożliwiający ocenę sytuacji na planszy (pozwalający na ocenę atrakcyjności zachowań, punktów docelowych etc),
3. moduł sztucznej inteligencji,
4. moduł nawigacji robota; moduł ten powinien być odpowiedzialny za tworzenie bezkolizyjnej ścieżki prowadzącej do zadanego celu,
5. moduł sterowania ruchem robota; moduł ten powinien wyznaczyć optymalne prędkości prowadzące do zadanego punktu (problem bezwładności robota, wyhamowanie przed punktem docelowym etc.),
6. moduł bezpośrednio odpowiedzialny za sterowanie warstwą fizyczną robota (zadawanie prędkości liniowej kątowej, uruchamianie urządzenia do prowadzenia piłki(*dribbler-a*), kopnięcie piłki).

Zostanie opisana rola każdego z modułów, ze szczególnym uwzględnieniem modułu sztucznej inteligencji. Architektura STP zakłada planowanie działań drużyny na 3 poziomach.

1. Poziom położony najwyżej w hierachii: Play.
2. Tactics
3. Skills

- doprowadzenie piłki do celu piłki,
- przemieszczenie robota do celu,
- podążanie za innym robotem,

Rozdział 6

Sterowanie modelem robota z *Small-size League*

Streszczenie

W rozdziale zostanie zaprezentowana trójkołowa holonomiczna baza jezdna. Wyjaśniony zostanie sposób w jaki osiągnięto omnikierunkowość robota. Dodatkowo zostanie omówiony algorytm wyznaczania prędkości kół robota przy zadanej prędkości liniowej i kątowej robota. Zaprezentowany zostanie także problem poruszania się robota z piłką i ograniczenia z niego wynikające.

6.1 Omówienie omnikierunkowej bazy jezdnej

6.1.1 Opis położenia kół

6.1.2 Opis kinematyki oraz dynamiki bazy

6.2 Dryblowanie z piłką

- ograniczenia wynikające z budowy dribblera,
- wyznaczanie dopuszczalnych prędkości.

Rozdział 7

Algorytmy unikania kolizji

Streszczenie

W rozdziale zostanie szerzej zaprezentowany jeden z algorytmów unikania kolizji jakim jest RRT(**R**apidly-**E**xploring **R**andom **T**ree). Poruszona zostanie kwestia powodów, dla których wybrano tę, a nie inną metodę. Uzasadnione zostanie odejście od algorytmu opracowanego w ramach pracy inżynierskiej(CVM **C**urvature **V**elocity **M**ethod). Ponadto opisane zostaną inne metody planowania ścieżki, omówione zostaną ich właściwości. Na tej podstawie zostanie uzasadniony wybór algorytmu RRT. Omówione zostaną także szczegóły implementacji algorytmu.

7.1 Krótki przegląd algorytmów unikania kolizji

Krótki przegląd metod unikania kolizji z odsyłaczami do literatury szerzej opisującej problem. Wstępne odrzucenie globalnych metod planowania ruchu, ze względu na dużą złożoność obliczeniową, a co z tym idzie niską efektywność w dynamicznym środowisku.

- algorytm BUG
- algorytm VFH
- technika dynamicznego okna
- algorytm CVM
- metoda pól potencjałowych
- algorytm RRT

7.2 Zalety algorytmu RRT w stosunku do CVM

- Pokazanie zalet algorytmu oraz prostoty jego wykorzystania w przypadku robotów o napędzie holonomicznym.
- Szybka możliwość sterowania czasem trwania algorytmu, poprzez sterowanie liczbą węzłów drzewa.
- Proste modyfikacje algorytmu uwzględniające ograniczenia wynikające z dynamiki oraz kinematyki robota.

7.3 Szczegółowa zasada algorytmu RRT

- opis algorytmu budującego drzewo
- opis sposobu określania przestrzeni osiągalnych stanów robota w zależności rozpatrywanego węzła w drzewie
 - uwzględnianie kolizji na drodze od położenia bierzącego punktu docelowego
 - przeszkody opisywane za pomocą okręgów(promień okręgu przeszkody podczas planowania ścieżki jest większy niż podczas wykrywania kolizji)
- modyfikacje wprowadzone do podstawowej wersji algorytmu
 - waypoints
 - ograniczenie na maksymalną liczbę węzłów
 - uwzględnienie prędkości przeszkód

7.4 Szczegóły implementacji zastosowanego algorytmu

- opis wykorzystywanych struktur danych
- omówienie funkcji losującej stan docelowy

7.5 Wyznaczanie prędkości prowadzącej do punktu docelowego

Wyznaczenie punktu znajdującego się na bezkolizyjnej ścieżce prowadzącej do celu nie rozwiązuje problemu nawigacji robota, kolejnym krokiem jest wyznaczenie prędkości prowadzących robota do celu. W paragrafie zostanie opisany mechanizm zastosowany do wyznaczania prędkości robota w dwóch przypadkach:

- dojazd do celu,
- poruszanie się do celu z piłką.

Rozdział 8

Testy algorytmu nawigacji robota

W rozdziale zostaną zaprezentowane testy warstwy odpowiadającej za nawigację robotem, a więc sprawdzony zostanie algorytm RRT oraz algorytm wyznaczania prędkości prowadzącej do zadanego punktu. Testy zostaną przeprowadzone zarówno w środowisku statycznym jak i dynamicznym.

8.1 Zachowanie algorytmu RRT w środowisku statycznym

W paragrafie zostaną opisane eksperymenty dowodzące poprawność działania algorytmu RRT. Ponadto eksperyment ma pomóc w dobraniu odpowiedniej wagi parametru p odpowiadającego za kierowanie robota do punktu docelowego.

Do wizualizacji ścieżki utworzonej przez robota w każdym kroku algorytmu RRT zostanie użyta osobna aplikacja napisana w języku JAVA. **Plan eksperymentów:**

- w jednej linii ustawionych kilka robotów a za nimi piłka, jeden z robotów ma za zadanie dotrzeć do piłki,
- przedmiotem badań będzie czas wykonania algorytmu na który bezpośrednio przekłada się maksymalna dopuszczalna liczba węzłów drzewa oraz parametr p określający prawdopodobieństwo, że w kolejnym kroku algorytmu robot będzie poruszał się do celu,

Na podstawie eksperymentów zostaną sporządzone rysunki wyznaczające trajektorie po jakiej poruszał się robot na drodze do celu oraz średni czas dojazdu dla wybranych parametrów.

8.2 Środowisko dynamiczne

W paragrafie zostaną opisane eksperymenty mające na celu sprawdzenie efektywności algorytmu RRT w środowisku zmiennym w czasie. Sprawdzona zostanie efektywność modyfikacji algorytmu RRT pod kątem pracy w takim środowisku (m.in. predykcja położeni ruchomych przeszkód), **Plan eksperymentów:**

1. pierwsze eksperymenty mają na celu wykazanie poprawności działania algorytmu RRT w środowisku dynamicznym, a więc za pomocą algorytmu sterowany będzie jeden robot, reszta natomiast poruszać się będzie po stałych trajektoriach ze stałą prędkością,
2. wszyscy zawodnicy obu drużyn sterowani za pomocą algorytmu RRT kierowani są do losowych punktów docelowych, przedmiotem eksperymentu jest liczba wszystkich kolizji w przeciągu całego czasu trwania eksperymentu.

8.3 Prowadzenie piłki

W paragrafie zostaną opisane testy sprawdzające poruszanie się robota w kierunku zadanego punktu docelowego w sytuacji kiedy prowadzi on piłkę.

1. opisanie dodatkowych ograniczeń wynikających z prowadzenia piłki,
2. porównanie z wynikami otrzymanymi w poprzednich paragrafach.

Rozdział 9

Testy algorytmu planowania i koordynacji działań

- 9.1 Testy planowania działań na poziomie pojedynczego robota (Tactics)
- 9.2 Testy planowania działań na poziomie drużyny
- 9.3 Środowisko dynamiczne

Rozdział 10

Podsumowanie

Rozdział 11

Dodatek

11.1 Instrukcja konfiguracji Gazebo

11.2 Kod źródłowy modeli

Bibliografia

- [1] Oficjalna strona Ligi *RoboCup* dostępna pod adresem:
www.robocup.org
- [2] J. Bruce, M. Veloso: *Real-Time Randomized Path Planning for Robot Navigation*. Carnegie Mellon University
- [3] I. Duleba: *Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych*. Warszawa, Akademicka Oficyna Wydawnicza EXIT, 2001.
- [4] T. Quasn, L. Pyeatt, J. Moore: *Curvature-Velocity Method for Differentially Steered Robots*. AI Robotics Lab Computer Science Department, Texas Tech University, 2003.
- [5] R. Simmons: *The Curvature-Velocity Method for Local Obstacle Avoidance*. School of Computer Science, Carnegie Mellon University, 1996.
- [6] M. Majchrowski: *Algorytm unikania kolizji przez robota mobilnego bazujący na przeszukiwaniu przestrzeni prędkości*. Praca Magisterska, Politechnika Warszawska, 2006.
- [7] J. Borenstein, Y. Koren: *The vector field histogram – fast obstacle avoidance for mobile robots*. IEEE Transaction on Robotics and Automation, 1991.
- [8] J. Borenstein, Y. Koren: *Histogramic in-motion mapping for mobile robot obstacle avoidance*. Department of Mechanical Engineering and Applied Mechanics, The University of Michigan, 1991.
- [9] D. Fox, W. Burgard, S. Thrun: *The Dynamic Window Approach to Collision Avoidance*. Department of Computer Science, University of Bonn; Department of Computer Science, Carnegie Mellon University, Pittsburgh, 1991.

- [10] B.Browning, J.Bruce, M.Bowling, M.Veloso: *STP: Skills, tactics and plays for multi-robot control in adversarial environments*. Carnegie Mellon University, 2004.