

Symulacja dookólnej bazy mobilnej

Radosław Świątkiewicz

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

25 marca 2017

Spis treści

1 Wprowadzenie

2 Cel

3 Platforma

4 ROS

5 Implementacja

6 Komunikacja

7 Problemy

8 Przyszłość

Autor Radosław Świątkiewicz

Promotor dr hab. inż. Wojciech Szynkiewicz

Zespół Programowania Robotów i Systemów
Rozpoznających

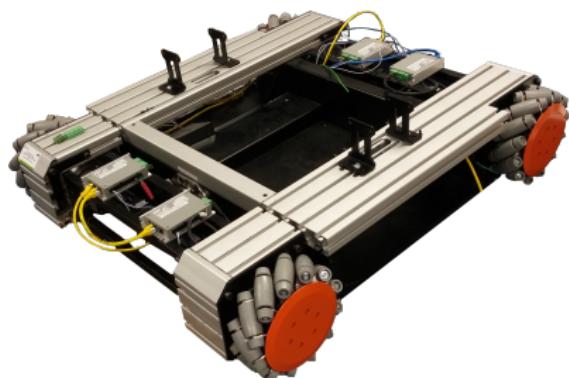
Instytut Automatyki i Informatyki Stosowanej

Projekt <https://github.com/Antyradek/omnivelma>

Co kryje się w P109



Robot manipulacyjny Velma



Platforma na kołach Mecanum

Co to jest model

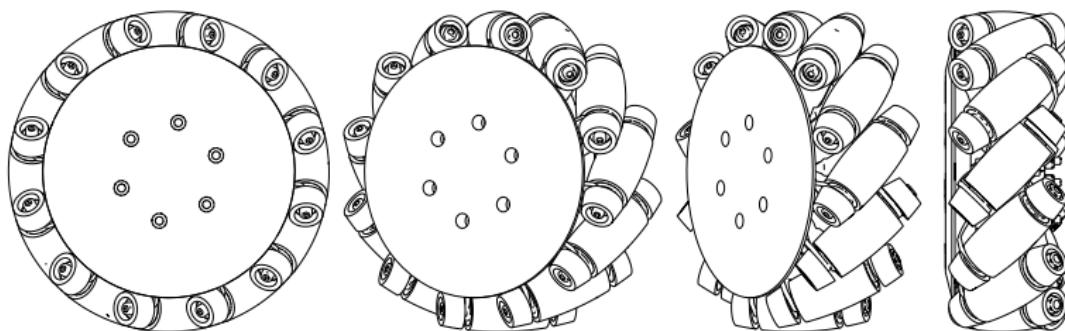
Po co potrzebny jest model:

- Pozwala bezpiecznie testować nowe oprogramowanie.
- Przyspiesza budowanie algorytmów sterowania.
- Ułatwia przeprowadzanie skomplikowanych testów.
- Daje możliwość implementacji nieistniejących czujników.

Wymagania:

- Reaguje na siły w podobny sposób, co robot.
- Przyjmuje takie samo sterowanie.
- Generuje odpowiednie dane z wirtualnych czujników.

Koła Szwedzkie (Mecanum)



Na co pozwalają

Wynalezione w 1973

Przez szwedzkiego inżyniera Beng Ilona zatrudnionego w firmie Mecanum.¹

- Rolka obrócona względem osi o 45° .
- Każde koło ma 12 pasywnych rolek.
- Koła posiadają po 3 stopnie swobody.
- Punkt kontaktu powinien płynnie przechodzić z rolki na rolkę.

¹Wikipedia

Podobne koła



Koło wielokierunkowe.²

²<http://www.robotshop.com>

³"Mechanical development of an automated guided vehicle", Master of Science Thesis MMK 2016:153 MKN 171

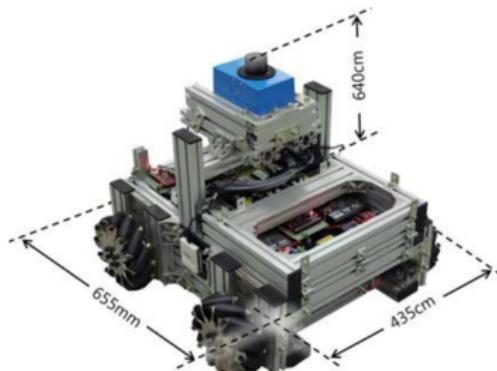


Bardziej wytrzymałe koło.³

Podobne mechanizmy



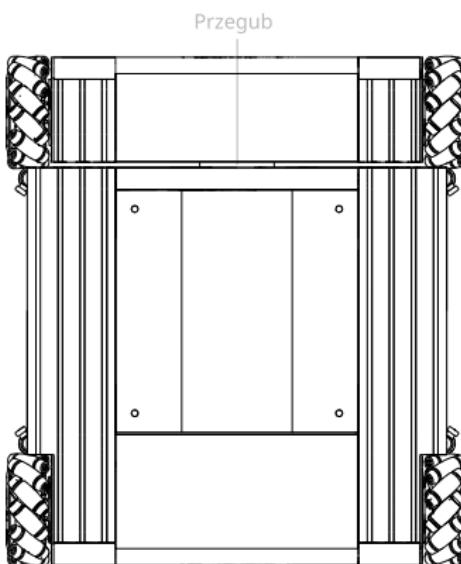
Fragment robota Kuka Youbot.



Inny robot AGV²

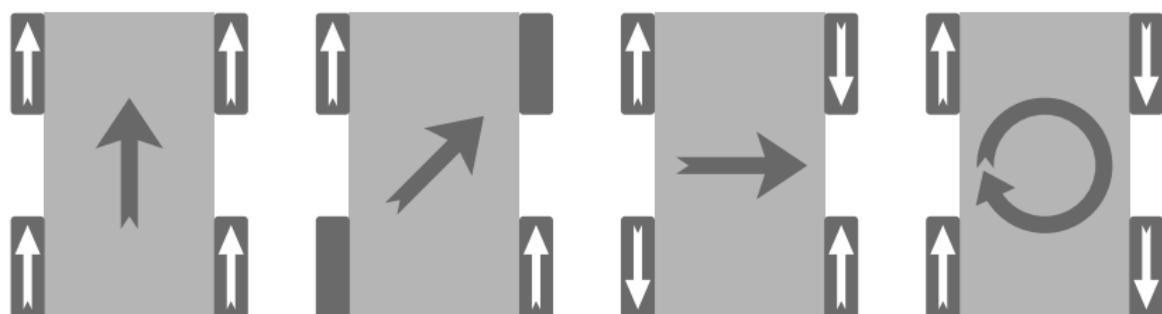
²International Journal Of Precision Engineering And Manufacturing Vol. 13, No. 3, pp. 379-386, DOI: 10.1007/s12541-012-0048-9

Model CAD



Podstawa jazdna posiada przegub w przedniej części.

Kierunki ruchu



Kinematyka

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \frac{r}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{2}{a+b} & \frac{-2}{a+b} & \frac{-2}{a+b} & \frac{2}{a+b} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

Oznaczenie	Opis
r	Promień koła w najszerzym miejscu.
a	Szerokość platformy.
b	Długość platformy.
ω_i	Prędkość kątowa kół.
v_x	Prędkość transwersalna w osi X.
v_y	Prędkość transwersalna w osi Y.
ω_z	Prędkość kątowa w osi Z (w góre).

Robot Operating System

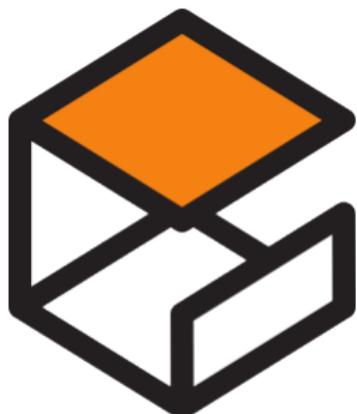


Logo ROS³

- Wbrew nazwie wcale nie jest systemem operacyjnym, a frameworkm.
- Umożliwia zarządzanie i uruchamianie osobnych modułów i dba o komunikację między nimi.
- Dostarcza szereg gotowych algorytmów i komunikatów.
- Moduł może być biblioteką, danymi, definicjami, programem wykonywalnym, skryptem. Niekoniecznie na tej samej maszynie.
- Komunikacja poprzez kolejki wiadomości i asynchroniczne wywołania, łatwe do przesyłania przez sieć.

³<http://www.ros.org/>

Gazebo

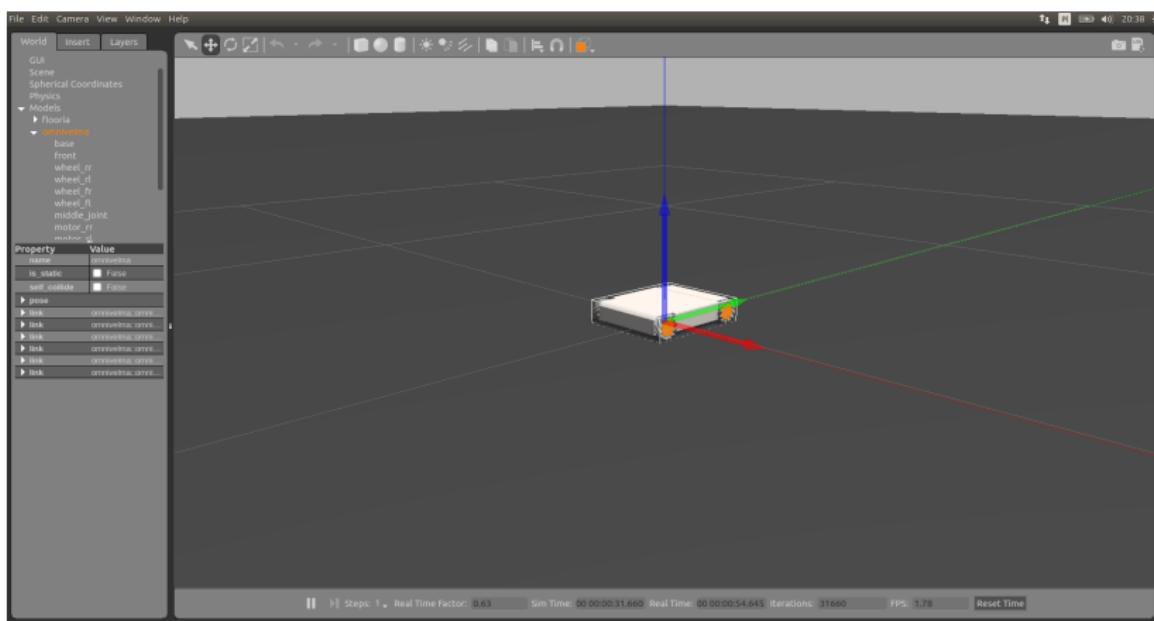


GAZEBO
Logo Gazebo⁴

- Symulator obiektów w przestrzeni wirtualnej.
- Używa silnika fizycznego do symulacji różnych części robota.
- Wtyczki są bibliotekami w C++ ładowanymi w trakcie wykonania.
- Obiekty symulacyjne zapisywane w XML-owym formacie SDF.

⁴<http://gazebosim.org/>

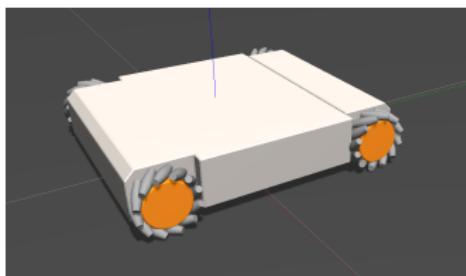
Edytor Gazebo



Jak to działa

- ➊ Tworzymy odpowiednie moduły w ROS-ie:
 - Model platformy i skrypty sterujące, oraz definicje komunikatów.
 - Plik definiujący przestrzeń symulacji.
 - Programy pomocnicze.
- ➋ Uruchamiamy symulację i komunikujemy z programem sterującym.
- ➌ Zbieramy dane, badamy dokładność programu i modelu, wprowadzamy poprawki.
- ➍ Podłączamy program sterujący do rzeczywistego robota.

Stworzenie modelu



Dynamiczny model platformy.

- ① Korzystając z modelu CAD wygenerować siatki o odpowiednich wymiarach.
- ② W pliku SDF zdefiniować składowe modelu, wygląd i przeguby.
- ③ Zapisać wtyczkę sterującą ruchem kół.
- ④ Umieścić wszystko w symulatorze i uruchomić.

Próba odwzorowania

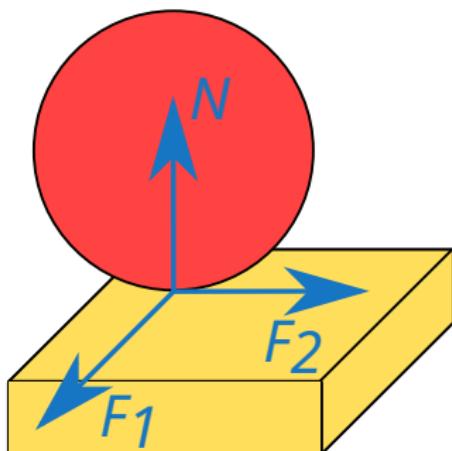
- Każde koło ma 12 rolek.
- Każda rolka ma przegub łączący ją z kołem.
- Każda rolka zawiera się w paraboloidzie, więc nie może być przybliżona wbudowanym kształtem.
- Każda siatka jest kanciasta i powoduje niejednostajne tarcie.
- Każda ściana generuje bardzo dużą ilość obliczeń silnika fizycznego.

Skomplikowanie modelu nie nadaje się do symulacji w czasie rzeczywistym, a także generuje duże błędy.

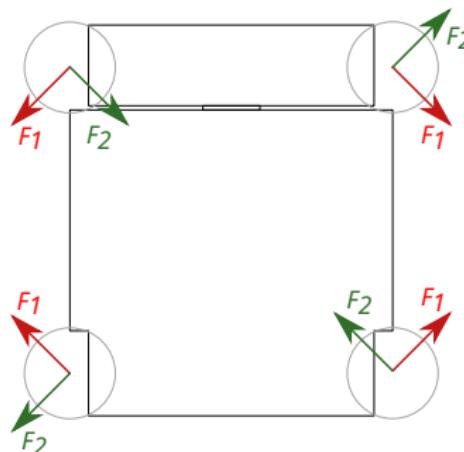
Próba resetowania pozycji rolek

- Do korpusu podłączone jest wirtualne koło, do tego koła podłączamy jedną obróconą o 45° rolkę, do której dodajemy kolizję z podłożem.
- W każdej klatce symulacji resetujemy pozycję rolki, aby nie obróciła się razem z kołem wirtualnym. Działa to tak, jakby koło zawsze dotykało podłożą jedną rolką.
- Gazebo nie ma zaimplementowanego drzewa pozycji obiektów względem siebie. Informacja o tym nie znajduje się w dokumentacji, tylko w notatkach o błędach, w kontroli wersji.
- Podobne rozwiązanie to resetowanie osi rolki w każdej klatce. Powoduje niekontrolowane zachowanie i skoki.

Modyfikacja wektorów tarcia



Wektory tarcia w symulatorach fizyki.⁵



Ustawienie wektorów tarcia w modelu. Ruch w kierunku F_1 jest ograniczony, w F_2 nieograniczony.

⁵ Na podstawie obrazka z dokumentacji ODE: http://www.ode-wiki.org/wiki/index.php?title=Manual:_Joint_Types_and_Functions

Wszystkie składowe

Omnivelma Model dynamiczny reagujący na siły.

Pseudovelma Model kinematyczny, sterowany wzorem.

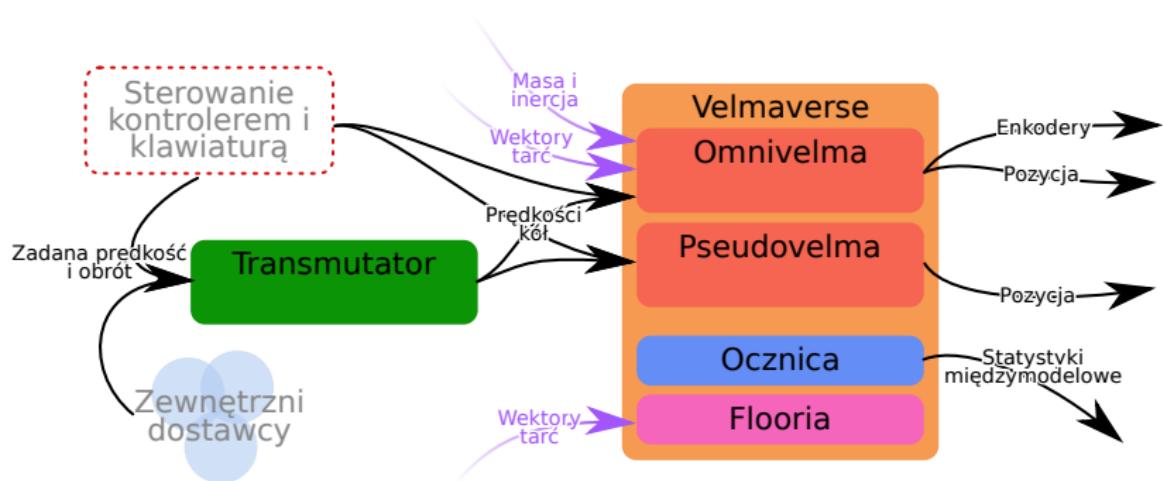
Velmaverse Skrypt uruchamiający Gazebo z definicją symulacji.

Transmutator Program tłumaczący zadany ruch na prędkości kół.

Ocznica Skrypt obserwujący scenę i produkujący informacje.

Flooria Podłoże ze zmiennym współczynnikiem tarcia.

Komunikacja wewnętrzna i zewnętrzna



Największe przeszkody w trakcie tworzenia

- ① Nieodpowiednie ustawienie zależności w CMakeLists.txt.
- ② Niezaimplementowane, a udokumentowane funkcjonalności.
- ③ Niewystarczająco dokładnie opisane API.
- ④ Niewielkie zainteresowanie tematem na forach i martwe linki.
- ⑤ Nieprzewidywalne zachowanie maszyny symulacyjnej.
- ⑥ Brak wsparcia twórców podobnych modeli.
- ⑦ Zależności od dokładnych wersji bibliotek i niemożność uruchomienia na nie-Ubuntu.

Ku przyszłości

- Moduł do łatwego sterowania z klawiatury, lub kontrolera.
- Ustawienie odpowiednich współczynników modelu w celu uzyskania zadowalającej jakości symulacji.
- Testowanie tego samego sterowania, co w symulacji na rzeczywistym robocie i zebranie wyników.
- Implementacja modelu w V-REPie.