

# Symulacja dookólnej bazy mobilnej

Radosław Świątkiewicz

Promotor: dr hab. inż. Wojciech Szynkiewicz  
Instytut Automatyki i Informatyki Stosowanej  
Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

7 czerwca 2017

# Spis treści

1 Wprowadzenie

2 Problem

3 Platforma

4 Narzędzia

5 Implementacja

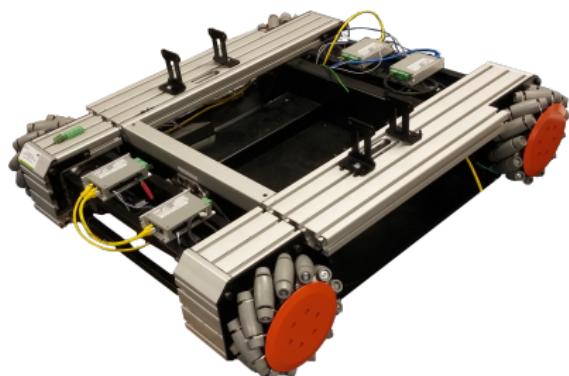
6 Pomiary

7 Podsumowanie

# Platforma z P109



Robot manipulacyjny Velma



Platforma na kołach Mecanum

# Cel projektu — zamodelowanie

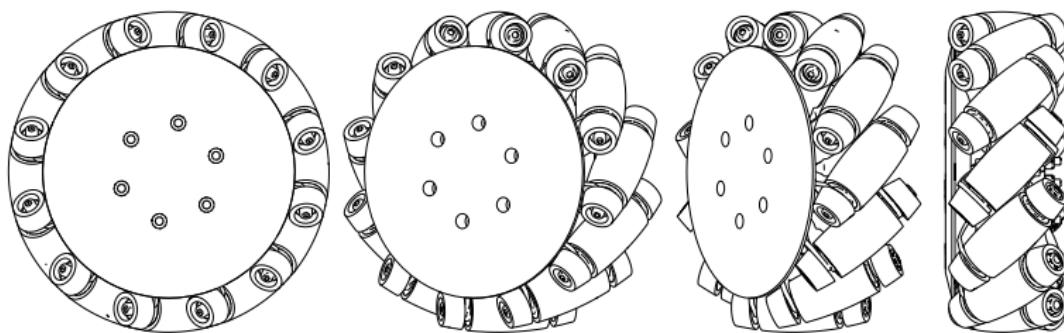
Po co potrzebny jest model:

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

Wymagania:

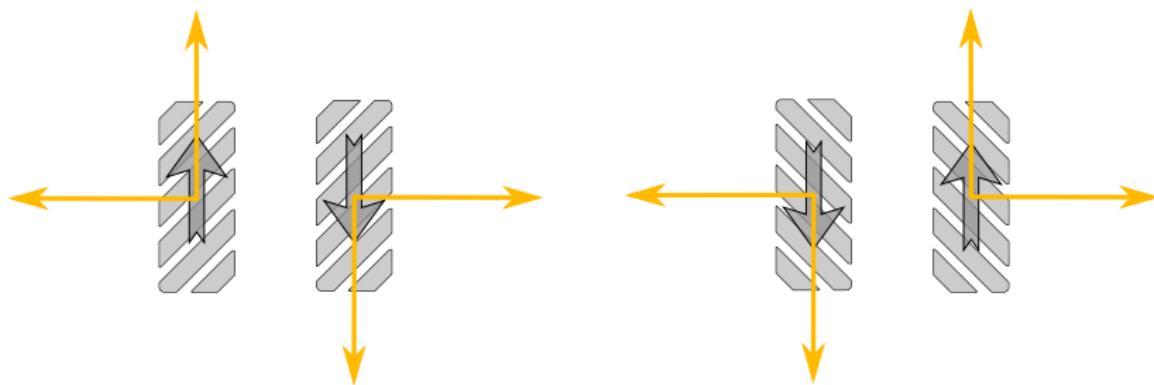
- Reaguje na siły w podobny sposób, co robot.
- Przyjmuje takie samo sterowanie.
- Generuje odpowiednie dane z wirtualnych czujników.

# Koła Szwedzkie (Mecanum)



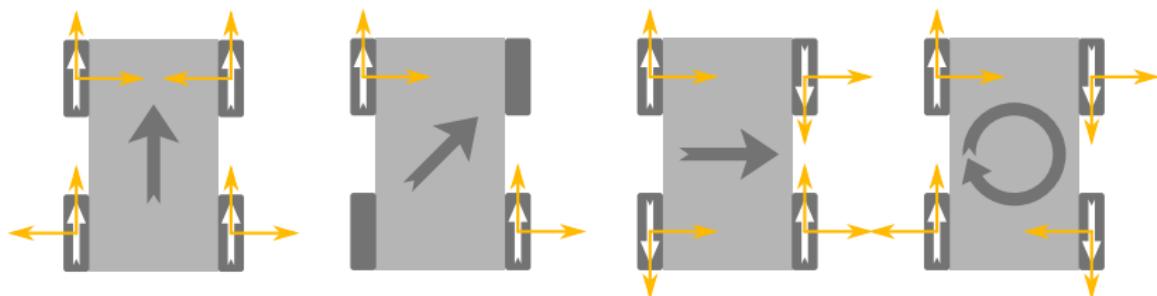
- Każde koło ma 12 pasywnych rolek.
- Rolka obrócona względem osi o  $45^\circ$ .
- Punkt kontaktu powinien płynnie przechodzić z rolki na rolkę.
- Koło pod wpływem momentu siły i tarcia generuje dwa wektory zamiast jednego.

# Dodatkowy wektor



Standardowa opona pod wpływem tarcia przekształca prędkość kątową na liniową w płaszczyźnie obrotu. Koło Mecanum ma dodatkowy wektor równoległy do osi obrotu, zatem prędkość wypadkowa jest obrócona o  $45^\circ$ .

# Kierunki ruchu



Poprzez znoszenie się składowych prędkości, pojazd może poruszać się w kierunkach nieosiągalnych dla urządzeń o zwykłych kołach.

# Kinematyka

Jak matematycznie przekształcić dane poszczególnych prędkości kół na wypadkową prędkość całości w danej chwili.

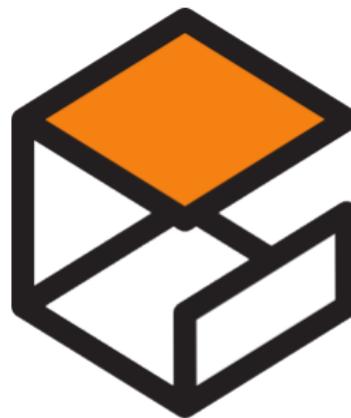
$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \frac{r}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{2}{a+b} & \frac{-2}{a+b} & \frac{-2}{a+b} & \frac{2}{a+b} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

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

# Główne komponenty



Logo ROS<sup>1</sup>



GAZEBO  
Logo Gazebo<sup>2</sup>



Logo V-Rep<sup>3</sup>

---

<sup>1</sup><http://www.ros.org/>

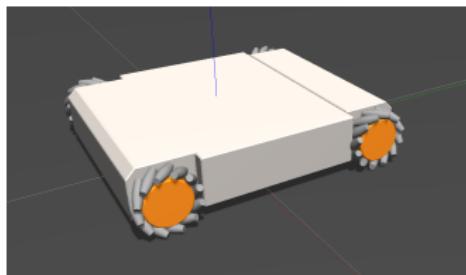
<sup>2</sup><http://gazebosim.org/>

<sup>3</sup><http://www.v-rep.eu>

# Struktura projektu

- Symulator, który wczytuje model platformy i wystawia interfejs kolejek wiadomości ROS-a.
- Zbiór pakietów i narzędzi ROS-a pozwala na sprawne zarządzanie kolejkami.
- Program do manualnego sterowania robotem lub algorytm wyznaczania trasy.
- Programy zbierające dane do wyświetlenia.

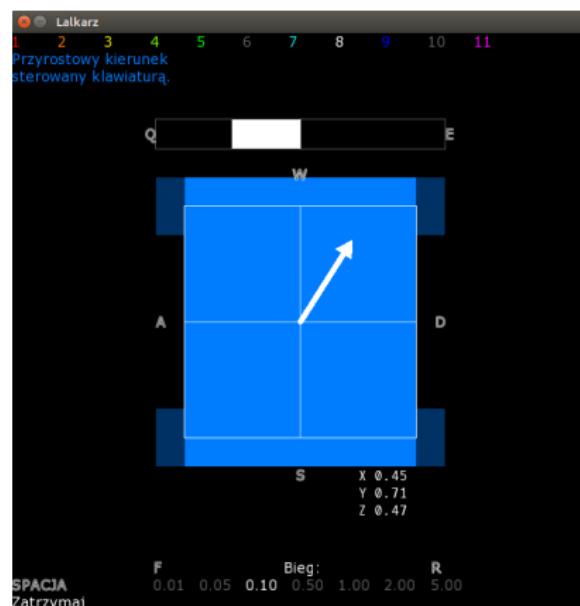
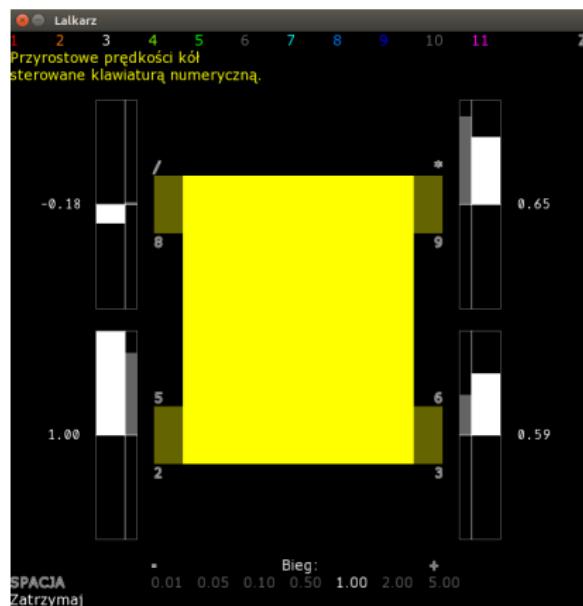
# Modele



Dynamiczny model platformy.

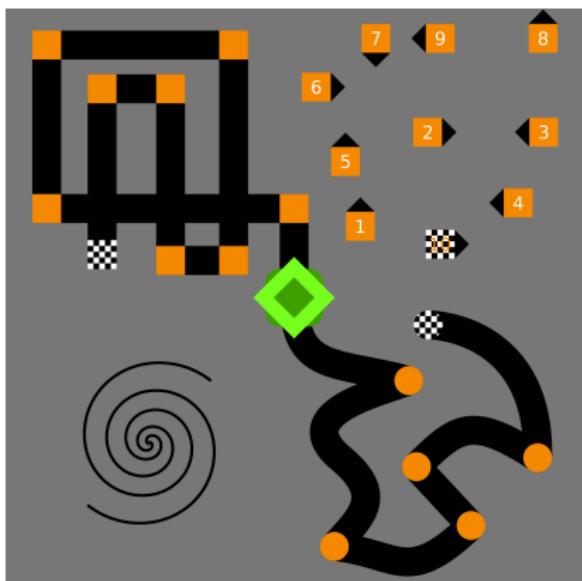
- Model dynamiczny w Gazebo.
- Matematycznie doskonały model kinematyczny.
- Model dynamiczny w V-Repie.

# Manualne sterowanie

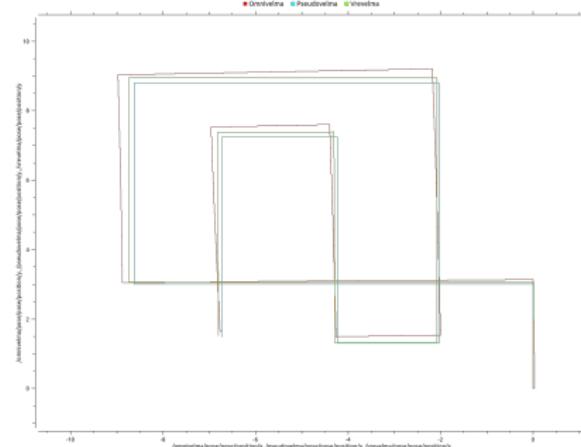


- Generuje bezpośrednie zadane prędkości kół lub kierunek.
- Obsługuje wejście z klawiatury, kontrolera i myszki.
- Wyświetla wartości modelu czujnika enkoderów.

# Obszar testów

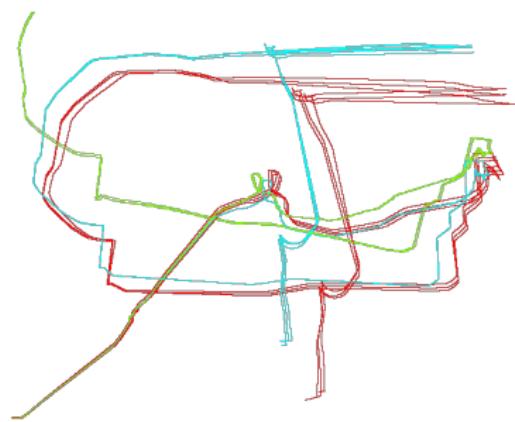


Rysunki na podłożu do łatwego testowania.

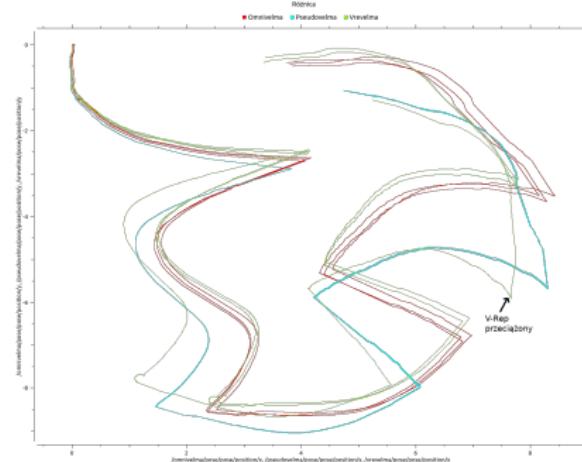


Najprostszy przebieg po pierwszej trasie, bez obrotu wokół osi.

# Powtarzalność



Druga trasa sterowana  
nieostrożnie.



Trzecia trasa sterowana binarnie.

# Pokaz?

Pokaz na żywo, bądź film.

# Aktualny stan

- Eksperymenty są powtarzalne.
- Pomimo różnej budowy symulatorów i modeli, trasy są podobne.
- Istnieje wiele aspektów w których można próbować ulepszyć model.

# Ku przyszłości

- Zdjęcie sterowania i przybliżonej trasy robota, a następnie ulepszanie modelu, aby jak najbardziej zbliżyć się do rzeczywistego przebiegu.
- Modyfikacje programu sterującego dla wygody.
- Próba implementacji modelu czujnika laserowego.
- Możliwie jak najbardziej dogłębna analiza modelu i określenie przyczyn rozbieżności.