



UNIVERSITÀ  
di **VERONA**

Dipartimento  
di **INFORMATICA**

Laurea magistrale in Ingegneria e scienze informatiche

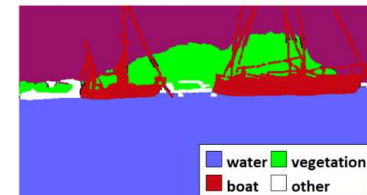
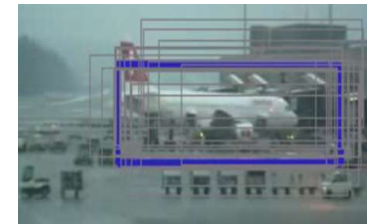
# Cinematica

*modelli e vincoli*

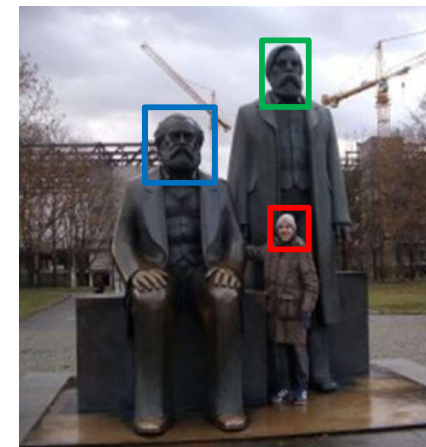
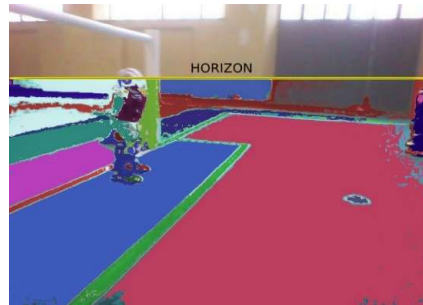


*Corso di Robotica*  
*Parte di Laboratorio*

Docente:  
**Domenico Daniele Bloisi**



Ottobre 2017



# Obiettivo

---

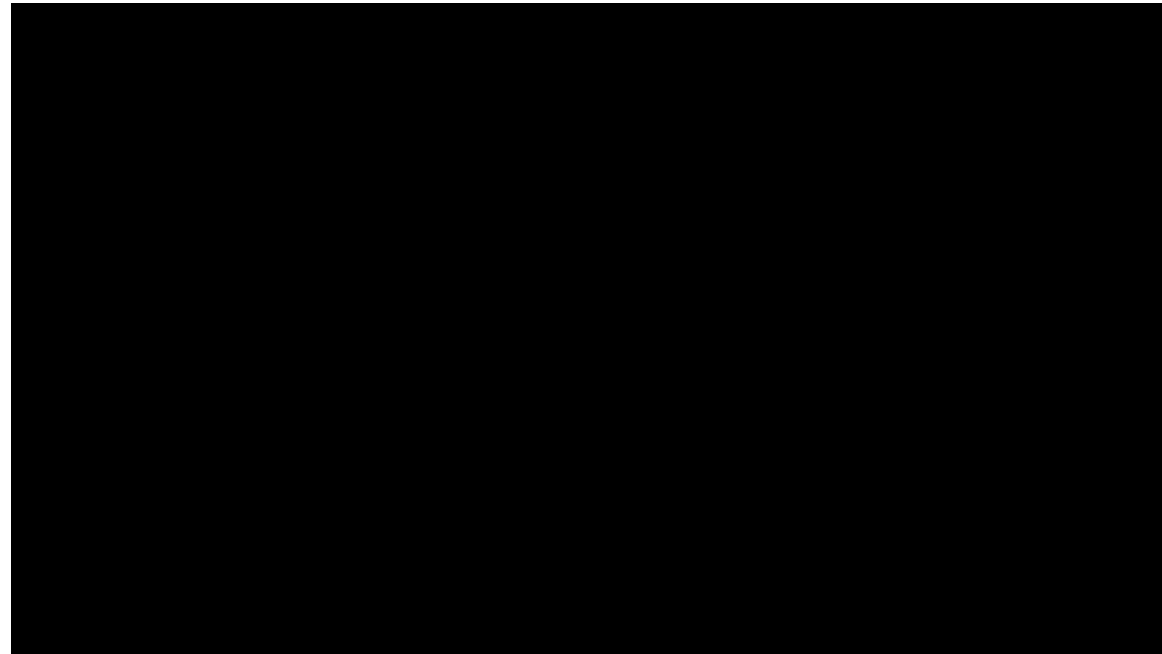
Capire il comportamento meccanico del robot per:

1. progettare in modo appropriato il robot per i task di interesse
2. creare al meglio il software di controllo per l'hardware a disposizione

# Manipolatori vs robot mobili

---

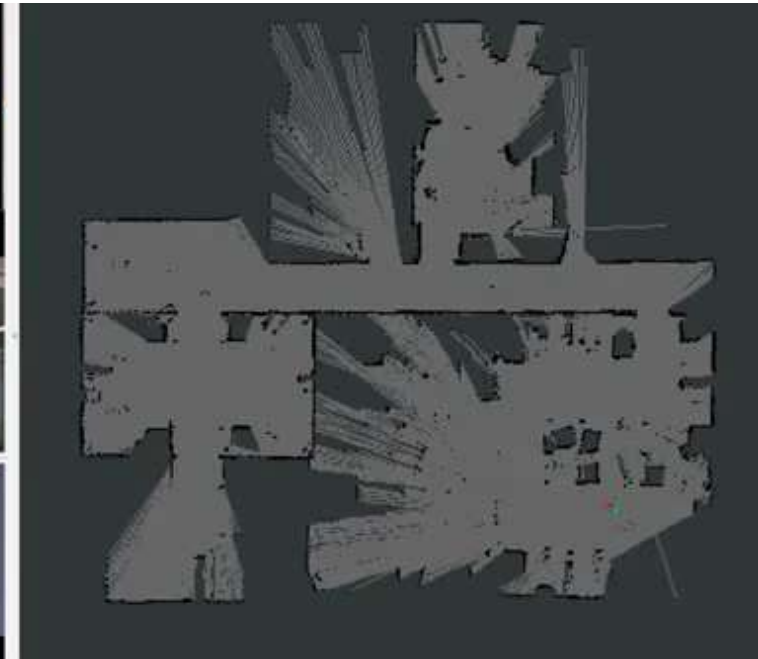
- I bracci robotici sono ancorati al terreno e hanno, di solito, un'unica catena di giunti
- Il workspace di un manipolatore definisce il range (relativamente al punto di ancoraggio) delle possibili posizioni che possono essere raggiunte dagli end-effector del robot



<https://www.youtube.com/watch?v=sWgvIAkfqXQ>

# Manipolatori vs robot mobili

- Il movimento di un robot mobile può essere definito attraverso i vincoli di **rotolamento** e **scivolamento** che agiscono al punto di contatto tra ruota e terreno
- Il workspace di un robot mobile definisce il range delle possibili **pose** che il robot può raggiungere nell'ambiente operativo



<https://www.youtube.com/watch?v=E8OKp31eMpE>

# Limitazioni

---

- Il movimento di un robot mobile è limitato dalla dinamica
- Per esempio, ad alte velocità, un centro di massa molto alto limita il raggio di curvatura (può esserci pericolo di cappottamento)



<https://www.youtube.com/watch?v=0iui1ACWw-c>

# Position estimation - Manipolatore

---

- Un manipolatore ha un'estremità ancorata ad un punto dell'ambiente
- Misurare la posizione dell'end-effector di un braccio richiede unicamente di conoscere la cinematica del robot e di misurare la posizione dei giunti intermedi
- La posizione di un manipolatore è sempre calcolabile avendo a disposizione i dati dei sensori

# Position estimation - Robot mobile

---

- Un robot mobile è un sistema auto-contenuto che si muove interamente rispetto all'ambiente (non ci sono punti fissi di contatto)
- Non c'è un modo diretto di misurare la posizione del robot mobile istantaneamente
- E' possibile integrare il movimento del robot al passare del tempo, ottenendo una *stima* del movimento

# Processo bottom-up

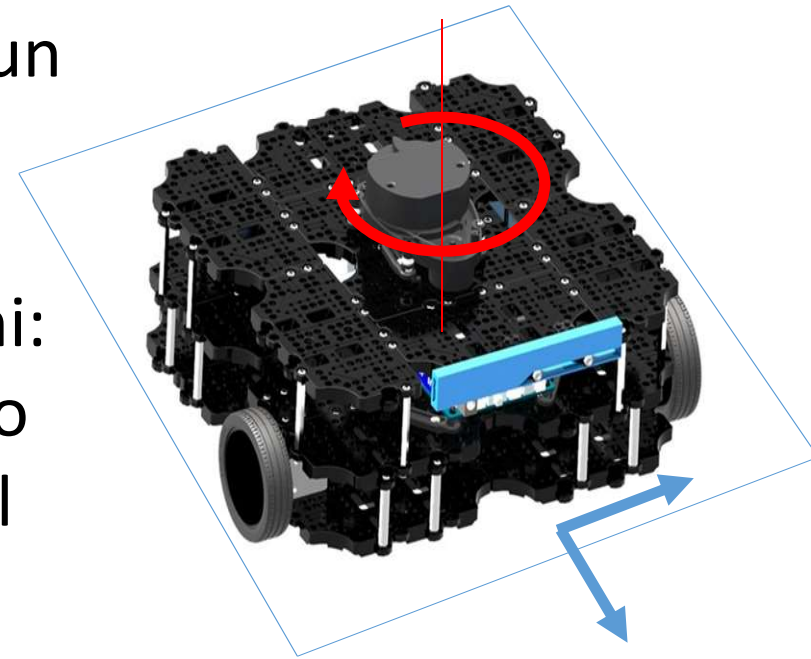
---

- Derivare il modello cinematico per un robot mobile è un processo bottom-up
- Ogni ruota contribuisce individualmente al movimento del robot e, al tempo stesso, impone dei vincoli al movimento
- Poiché le ruote sono collegate tra loro in base alla geometria della scocca, i vincoli posti dalla singola ruota si combinano per formare vincoli che si applicano all'intero sistema



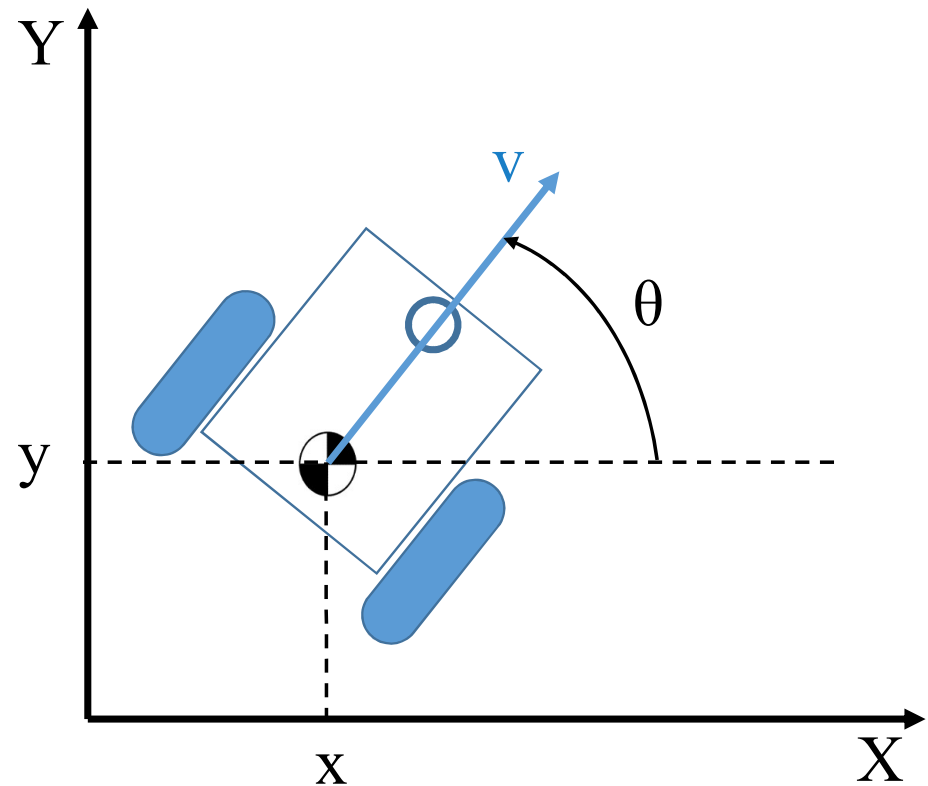
# Modello del robot mobile

- Il robot verrà modellato come un corpo rigido su ruote, in grado di muoversi su un piano orizzontale
- Il modello semplificato avrà 3 dimensioni:
  - 2 per descrivere la posizione nel piano
  - 1 per rappresentare l'orientazione del robot lungo l'asse verticale (che è ortogonale al piano su cui avviene il movimento)



# Robot pose

- La *robot pose* è definita come la posizione del robot e la sua orientazione in un dato sistema di riferimento
- Per un robot mobile che si muove su un piano, la *pose* è definita dalla tripla  $[x, y, \theta]$



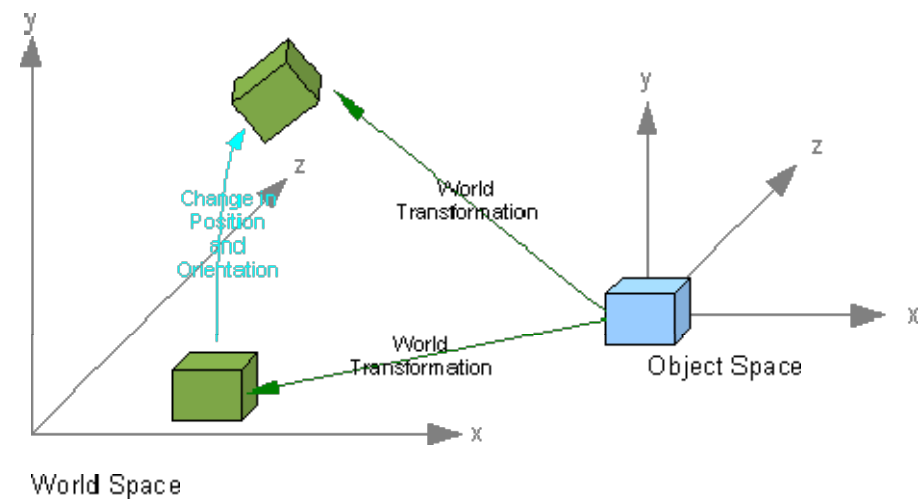
# Localizzazione

---

- Il termine localizzazione indica l'attività di determinare la robot pose
- “Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities”  
[I.J. Cox. Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. IEEE Transactions on Robotics and Automation, 7(2):193–204, 1991]

# Sistemi di riferimento

- Forze e vincoli relativi ad ogni ruota devono essere espressi rispetto ad un *chiaro e coerente* sistema di riferimento
- Poiché il robot si muove all'interno dell'ambiente, è necessario avere a disposizione un *mapping* tra il sistema di riferimento **locale** e quello **globale**



# Combinare i sistemi di riferimento

- Esistono molteplici modi di combinare i diversi sistemi di riferimento
- Questo può generare ambiguità

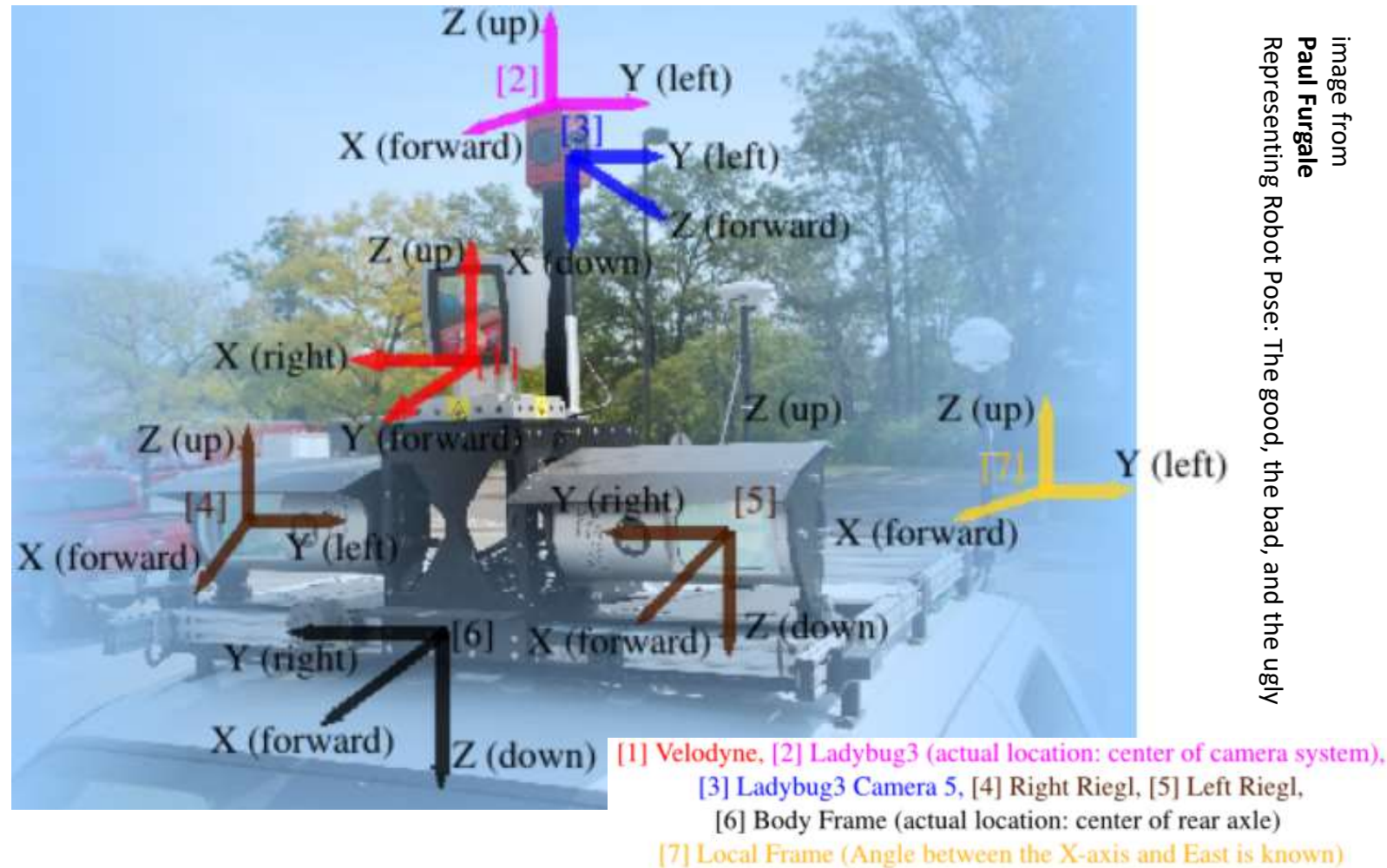


image from  
Paul Furgale  
Representing Robot Pose: The good, the bad, and the ugly

# Frame inerziale e frame del robot

## Sistema di riferimento globale

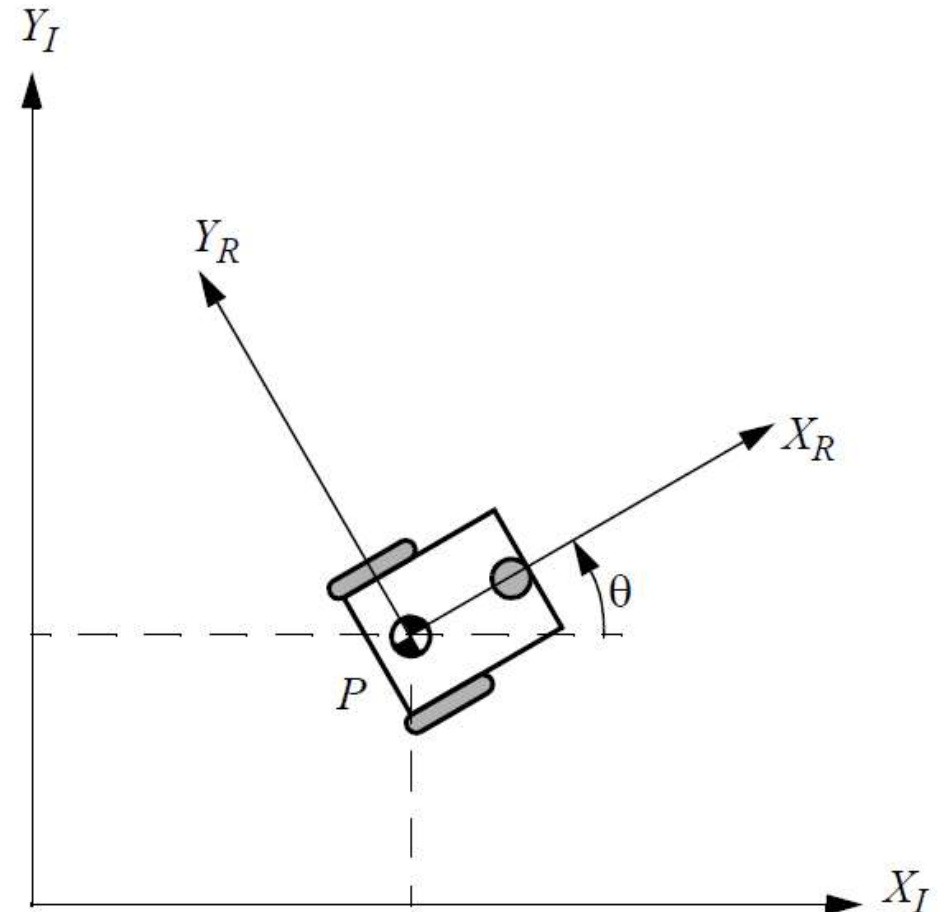
Gli assi  $X_I$  e  $Y_I$  definiscono una arbitraria base inerziale sul piano avente origine  $O: \{X_I, Y_I\}$

## Punto di riferimento per la posizione

Il punto  $P$  rappresenta la posizione del robot

## Sistema di riferimento locale

La base  $\{X_R, Y_R\}$  definisce il sistema di riferimento locale del robot (*body frame*)



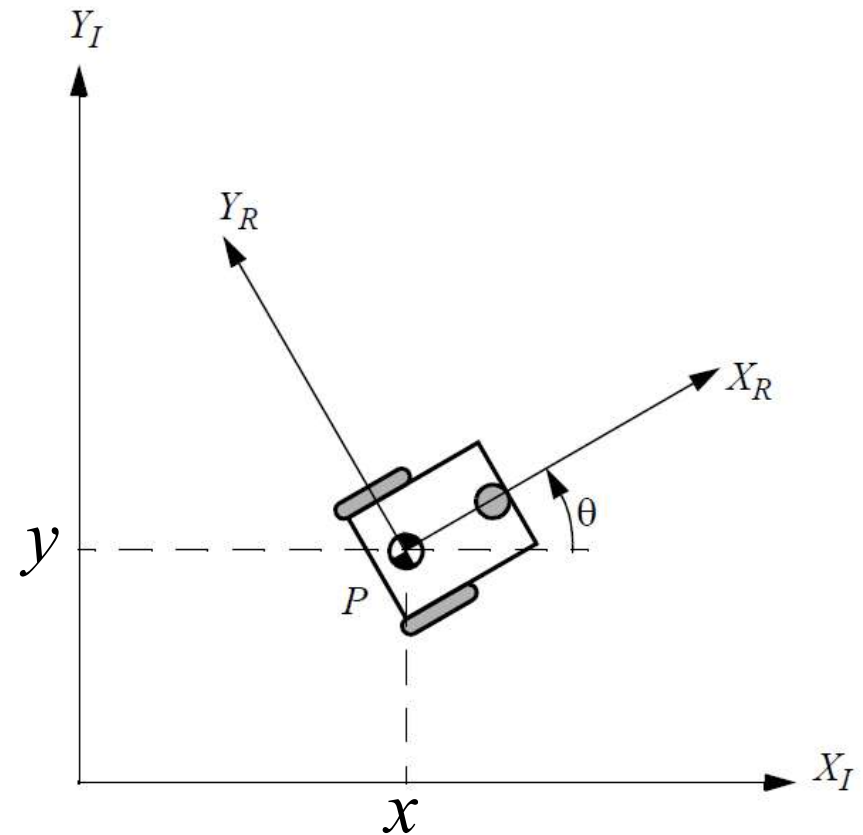
# Posizione $P$ nel frame globale

- Il punto  $P$  è rappresentato nel frame globale dalle coordinate  $x$  e  $y$
- la differenza angolare tra i frame locale e globale è data da  $\theta$

- Robot pose:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

espressa nel frame  
di riferimento globale



# Descrizione del movimento

---

- Vogliamo descrivere il movimento del robot in base al movimento delle sue componenti
- Per farlo è necessario trovare una trasformazione  $T$  che leghi il movimento che si osserva nel sistema di riferimento globale agli assi del sistema locale

$$\begin{array}{l} \text{movimento espresso} \\ \text{nel frame} \\ \text{di riferimento locale} \end{array} = T \chi \begin{array}{l} \text{movimento espresso} \\ \text{nel frame} \\ \text{di riferimento globale} \end{array}$$

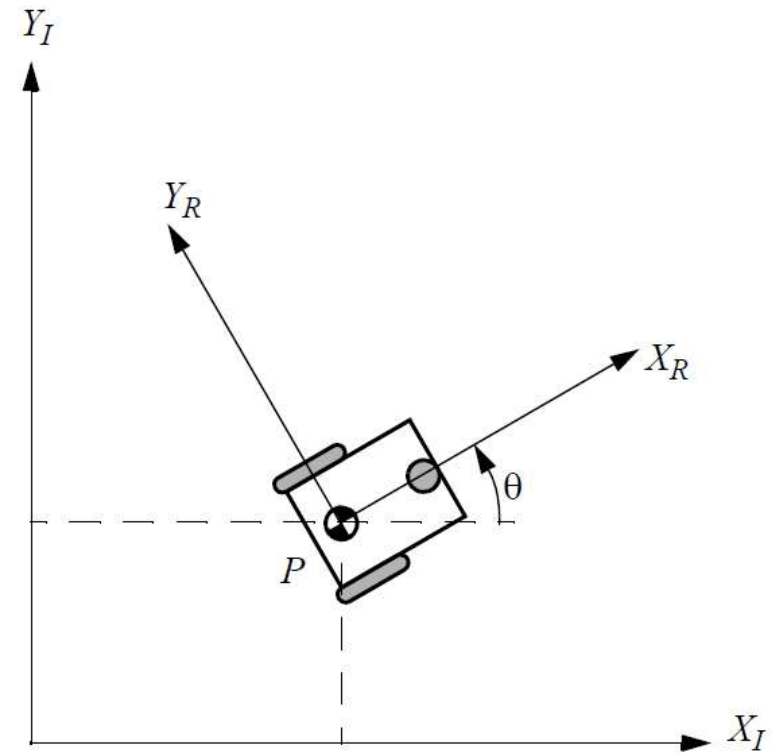
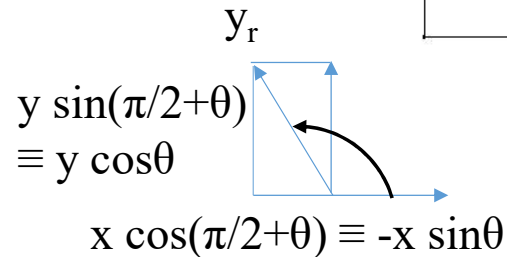
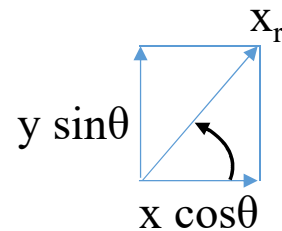


# Matrice di rotazione ortogonale

- La matrice di rotazione ortogonale  $R(\theta)$  serve a mappare nel frame di riferimento del robot  $\{X_R, Y_R\}$  il movimento calcolato nel frame di riferimento globale  $\{X_I, Y_I\}$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- otteniamo  $\dot{\xi}_R = R(\theta)\dot{\xi}_I$

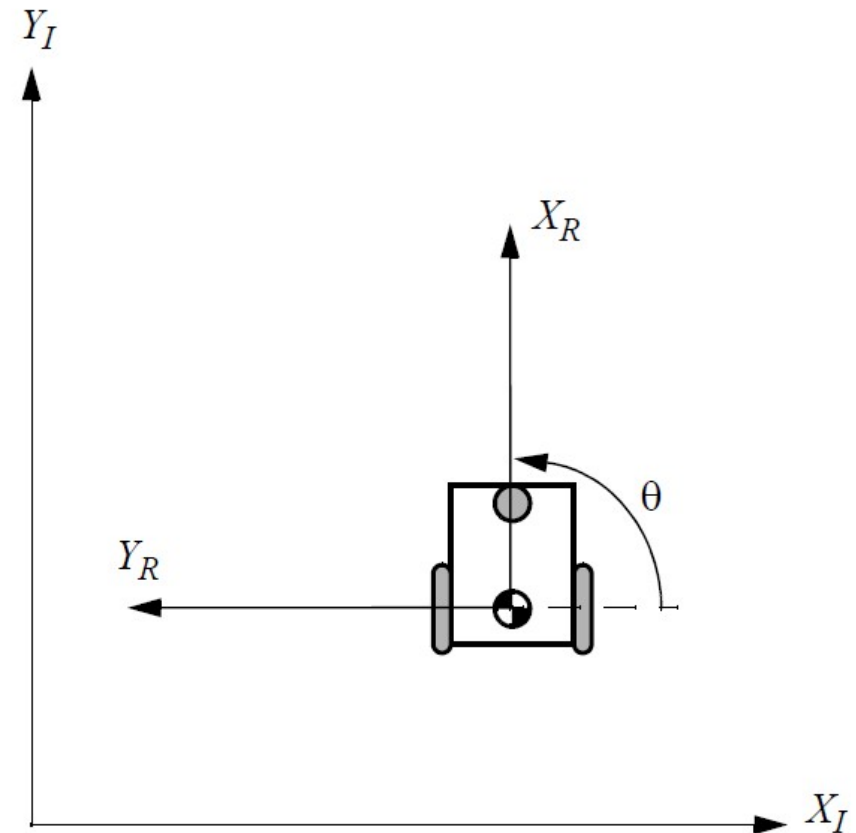


# Esempio: allineamento con un asse globale

- Calcolo della matrice di rotazione per il robot in figura

$$\theta = \frac{\pi}{2} \quad \dot{\xi}_R = R\left(\frac{\pi}{2}\right)\dot{\xi}_I$$

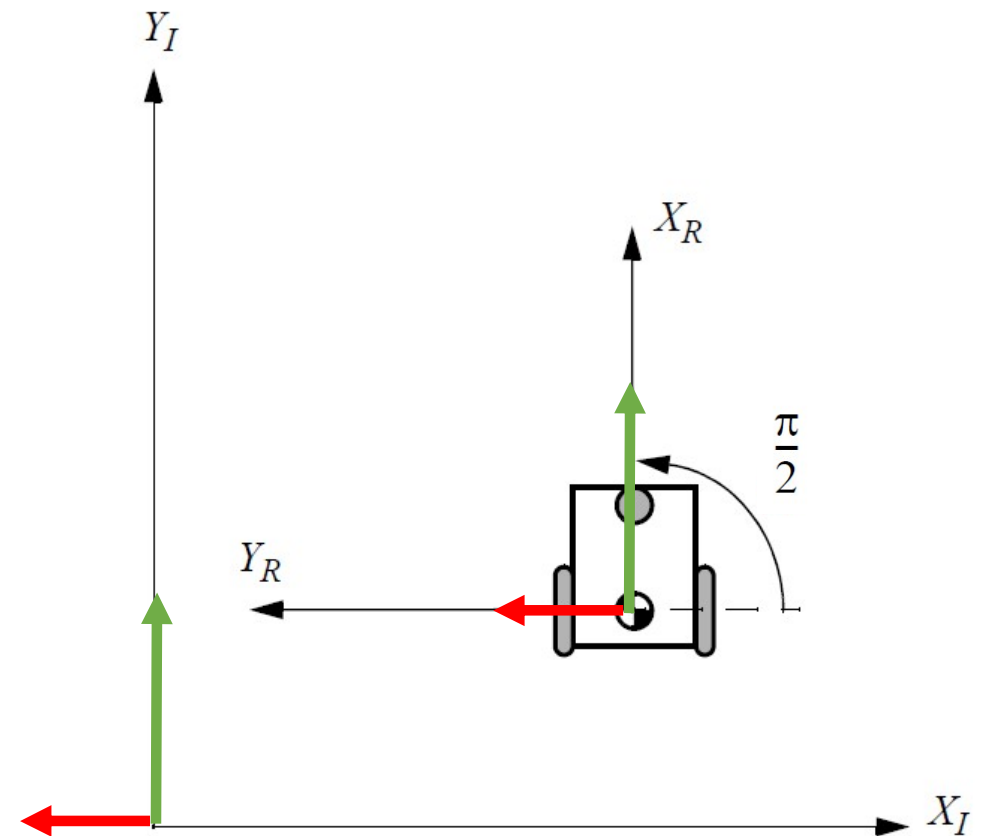
$$R\left(\frac{\pi}{2}\right) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



# Esempio: allineamento con un asse globale

- Data una certa velocità  $(\dot{x}, \dot{y}, \dot{\theta})$  calcolata nel frame globale, possiamo ricavare le componenti del movimento nel sistema di riferimento del robot come

$$\dot{\xi}_R = R\left(\frac{\pi}{2}\right)\dot{\xi}_I = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix}$$

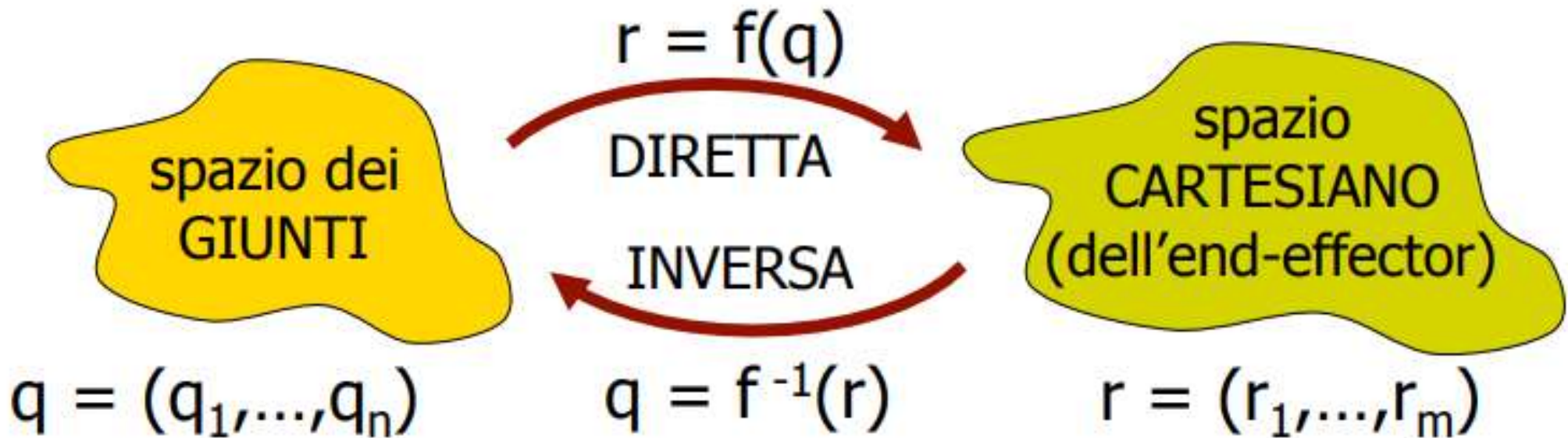


# Cinematica diretta e inversa

---

- La cinematica studia gli aspetti geometrici e temporali del moto delle strutture robotiche, senza riferimento alle cause che lo provocano
- La *cinematica diretta* è una trasformazione dallo spazio dei giunti allo spazio fisico
- La *cinematica inversa* è una trasformazione dallo spazio fisico allo spazio dei giunti. E' necessaria per controllare il movimento del robot

# Cinematica diretta e inversa



$n$  = numero di gradi di libertà (DoF)

# Esempio: differential drive robot

Ruote con diametro  $r$

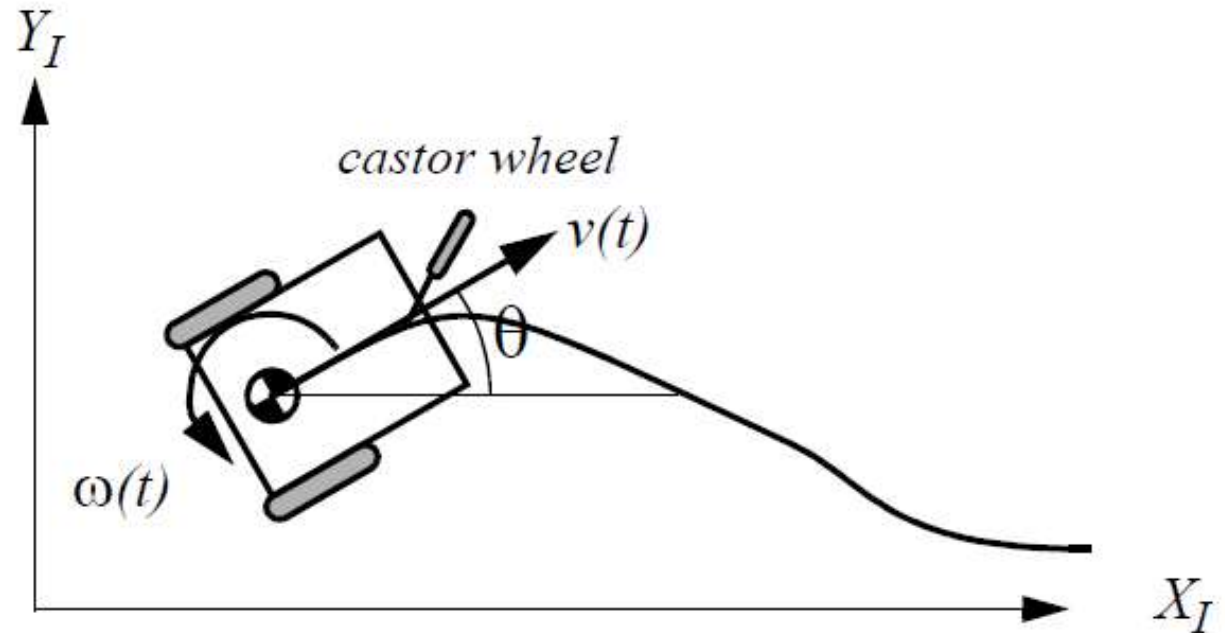
 punto  $P$  (centro di massa)

$l$  distanza della ruota dal centro di massa

$\dot{\varphi}_1$  e  $\dot{\varphi}_2$  velocità di rotazione delle due ruote

abbiamo:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2)$$



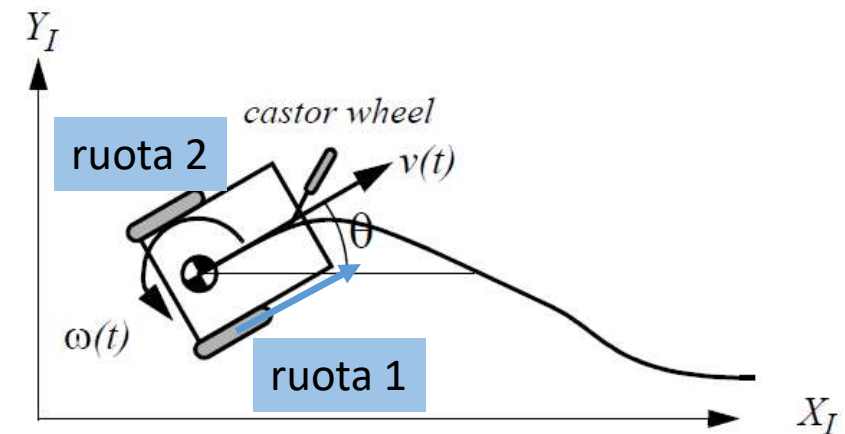
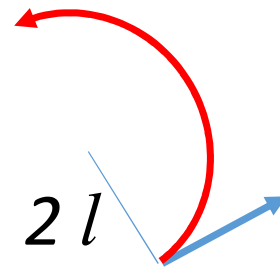
# Esempio: differential drive robot

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I \quad \longrightarrow \quad \dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$$

Strategia: calcoliamo (nel body frame) indipendentemente i contributi di ogni ruota per poi sommarli

Consideriamo solo il movimento della ruota 1 con la ruota 2 ferma  
abbiamo:

$$\dot{x}_{r1} = (1/2)r\dot{\phi}_1 \quad \text{e} \quad \omega_1 = \frac{r\dot{\phi}_1}{2l} \quad \text{con} \quad \dot{y}_{r1} = 0$$



# Esempio: differential drive robot

Consideriamo ora il movimento della ruota 2 con la ruota 1 ferma

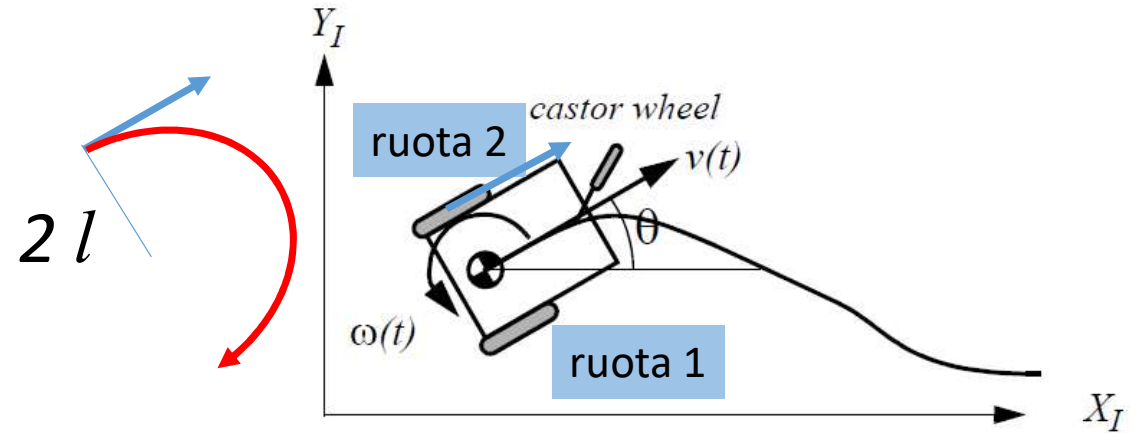
abbiamo:

$$\dot{x}_{r2} = (1/2)r\dot{\phi}_2 \quad \text{e} \quad \omega_2 = \frac{-r\dot{\phi}_2}{2l}$$

con  $\dot{y}_{r2} = 0$

Combinando le due ruote abbiamo:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix}$$





# Inversa di $R(\theta)$

---

$$\text{L'inversa di } R(\theta) \text{ è } R(\theta)^{-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pertanto otteniamo

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix}$$

# Vincoli cinematici

---

- Il primo step per ottenere il modello cinematico di un robot mobile consiste nell'esprimere i vincoli al movimento imposti dalle singole ruote
- Il movimento complessivo del robot viene calcolato combinando i movimenti delle singole ruote

# Ipotesi di funzionamento

---

- Le parti costituenti sono rigide
- Il robot è composto da una base su cui sono assemblate una o più ruote ideali
- Le ruote possono essere di vario tipo, attive o passive
- Le ruote possono essere sterzanti o non sterzanti, oppure costituite da cingoli
- L'asse di sterzata è sempre perpendicolare al suolo e vi è un unico punto di contatto tra ruota e terreno
- Il raggio delle ruote è costante
- Il moto delle ruote sul piano è ideale (rotolamento puro, no scivolamento laterale)

# Vincoli dati dalle ruote

---

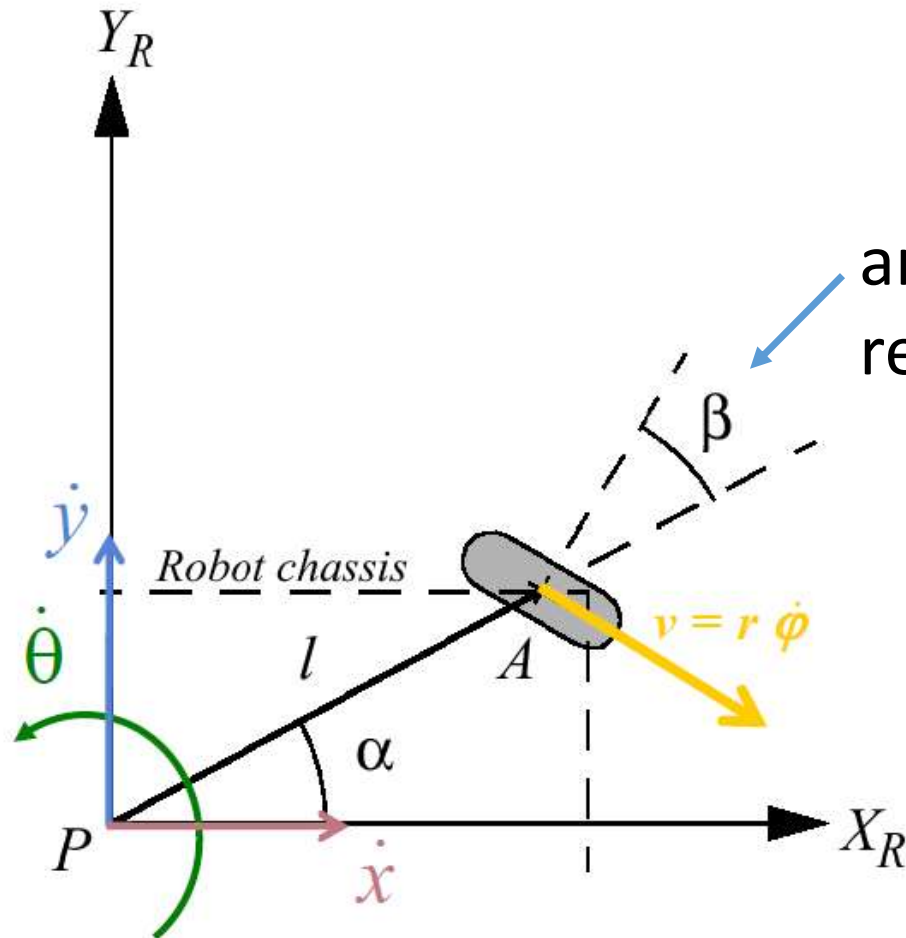
1. Vincolo sul rotolamento:

La ruota deve girare quando il movimento viene attuato nella giusta direzione

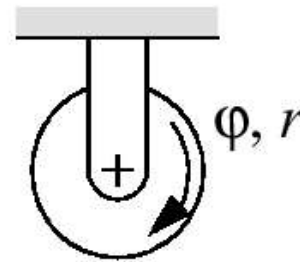
2. Vincolo sullo scivolamento:

La ruota non deve scivolare ortogonalmente al proprio piano

# Ruota semplice fissa



angolo del piano della ruota  
relativo alla scocca (fisso)

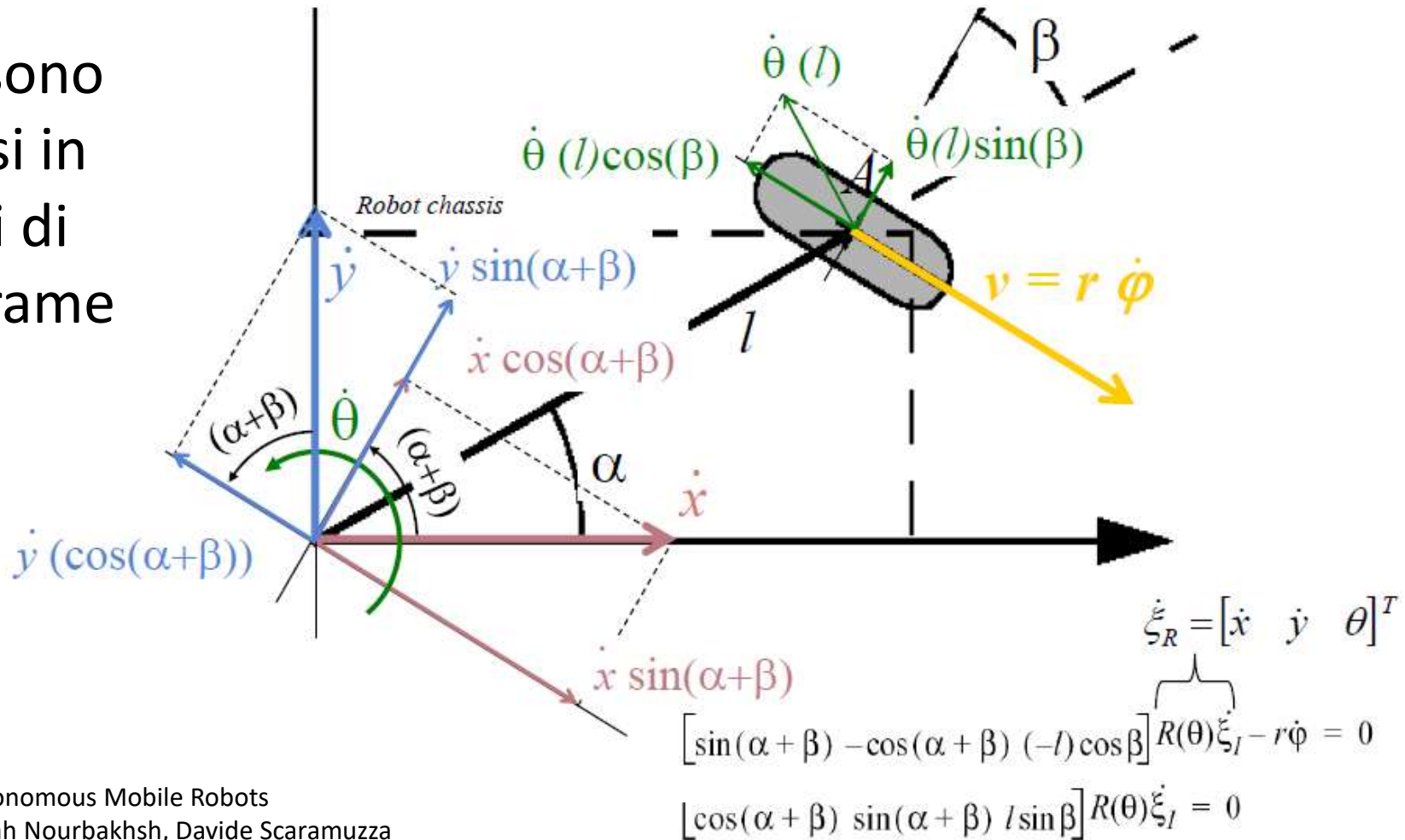


Punto  $A$  espresso  
in coordinate  
polari  $(l, \alpha)$

- Nessun asse di rotazione per la sterzata

# Ruota semplice fissa

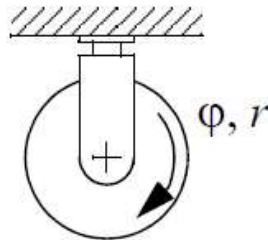
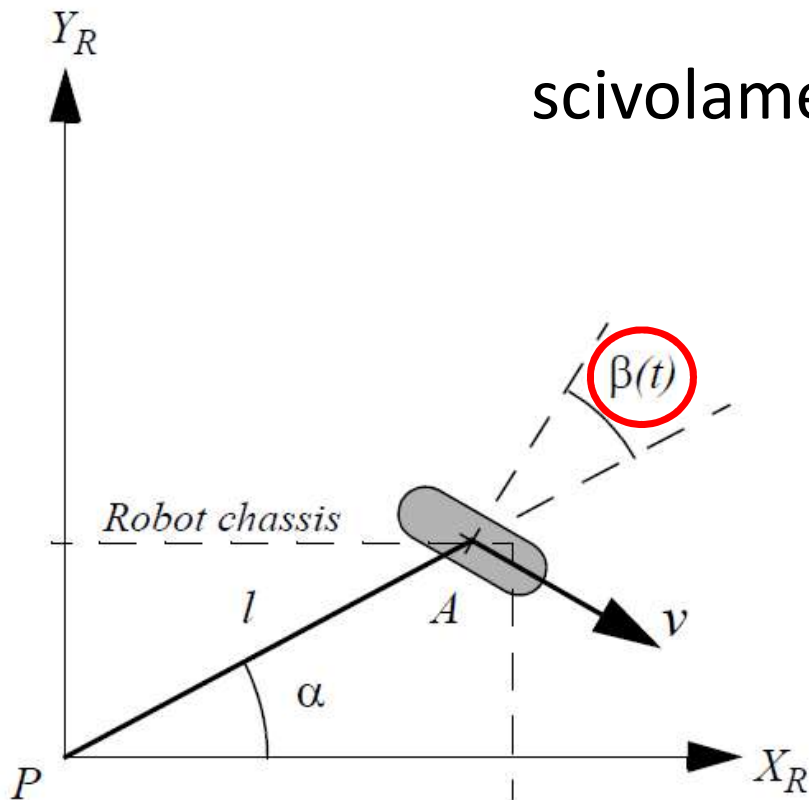
$\alpha$ ,  $\beta$ ,  $l$  sono espressi in termini di body frame



# Ruota semplice sterzante

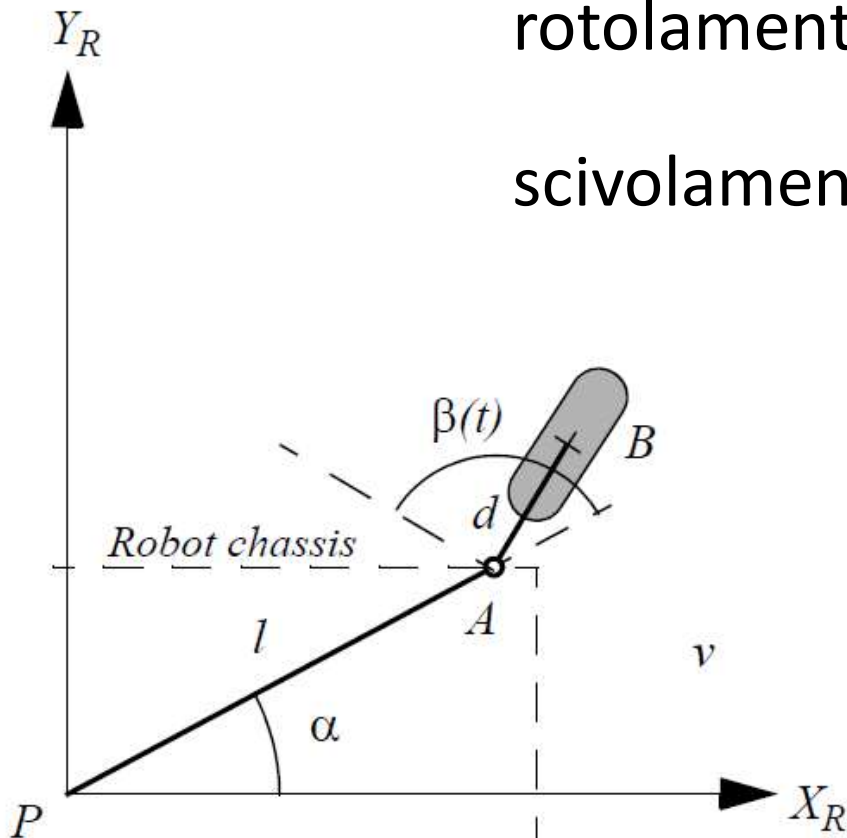
rotolamento  $\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l)\cos\beta \end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\phi} = 0$

scivolamento  $\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l\sin\beta \end{bmatrix} R(\theta)\dot{\xi}_I = 0$



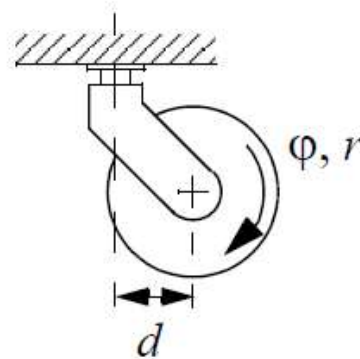
- vincoli identici a quelli per la ruota fissa
- $\dot{\beta}$  non ha impatto diretto quando si considera il movimento istantaneo

# Ruota castor



rotolamento  $\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l)\cos\beta \end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$

scivolamento  $\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & d + l\sin\beta \end{bmatrix} R(\theta)\dot{\xi}_I + d\dot{\beta} = 0$



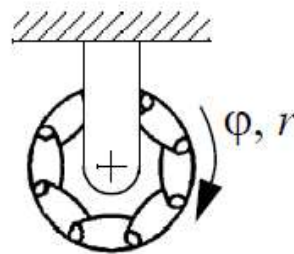
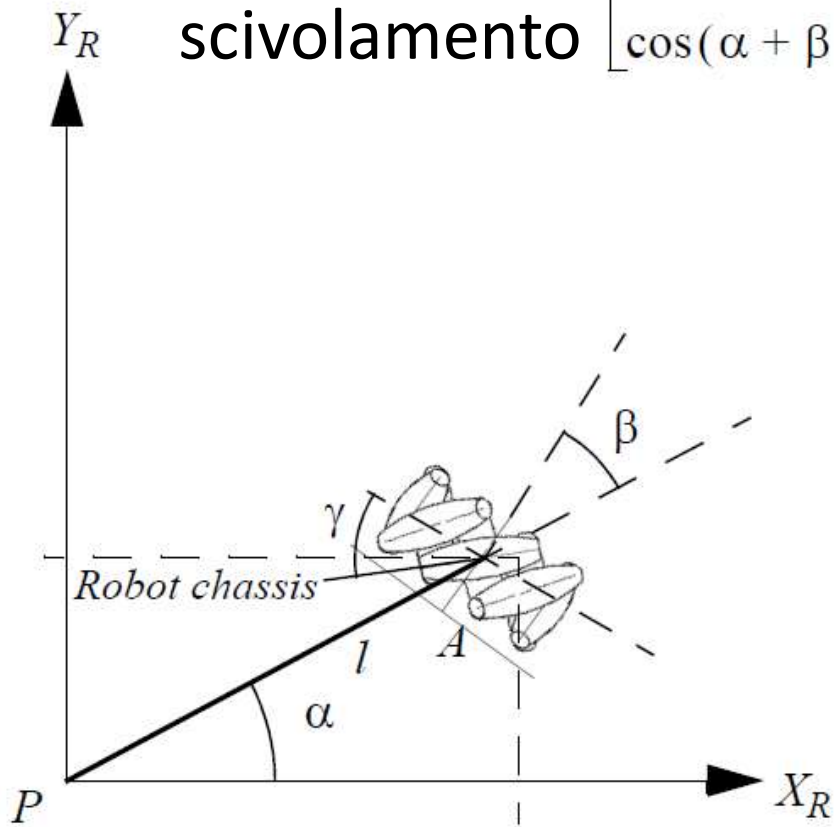
ogni movimento ortogonale al piano della ruota deve essere bilanciato da un opposto ed equivalente movimento di sterzata del castor



# Swedish wheel

rotolamento  $\left[ \sin(\alpha + \beta + \gamma) \quad -\cos(\alpha + \beta + \gamma) \quad (-l) \cos(\beta + \gamma) \right] R(\theta) \dot{\xi}_I - r \dot{\phi} \cos \gamma = 0$

scivolamento  $\left[ \cos(\alpha + \beta + \gamma) \quad \sin(\alpha + \beta + \gamma) \quad l \sin(\beta + \gamma) \right] R(\theta) \dot{\xi}_I - r \dot{\phi} \sin \gamma - r_{sw} \dot{\phi}_{sw} = 0$



- Non c'è un asse di rotazione verticale
- $\gamma$  è l'angolo tra il piano principale della ruota e gli assi di rotazione dei roller

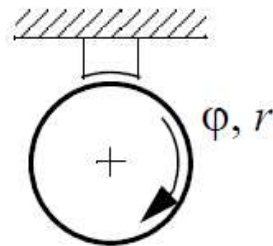
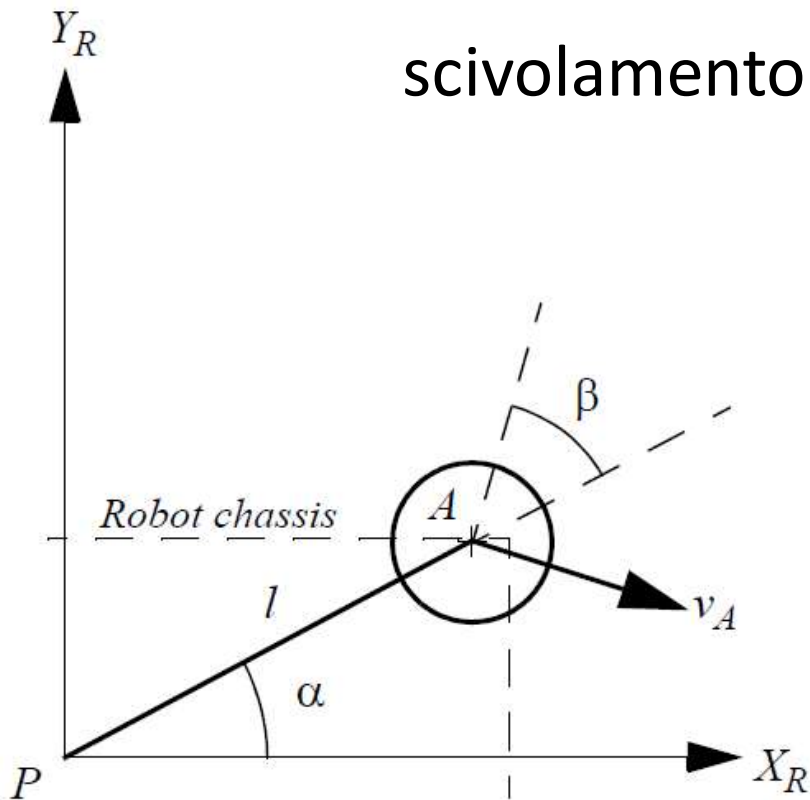
# Ruota sferica

rotolamento

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l) \cos \beta \end{bmatrix} R(\theta) \dot{\xi}_I - r \dot{\phi} = 0$$

scivolamento

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I = 0$$



- Non ci sono vincoli diretti sul movimento
- Non esiste un asse principale di rotazione
- vincoli identici a quelli per la ruota fissa



UNIVERSITÀ  
di **VERONA**

Dipartimento  
di **INFORMATICA**

Laurea magistrale in Ingegneria e scienze informatiche

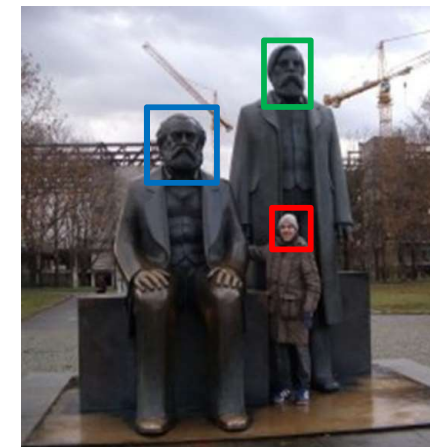
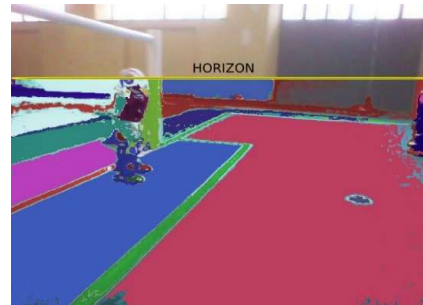
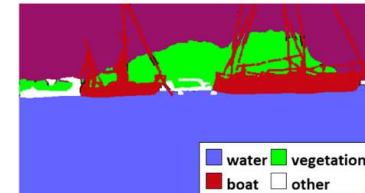
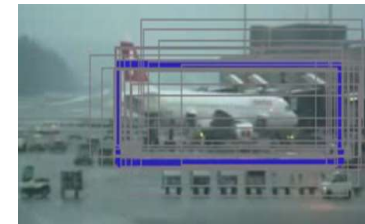
# Cinematica

*modelli e vincoli*



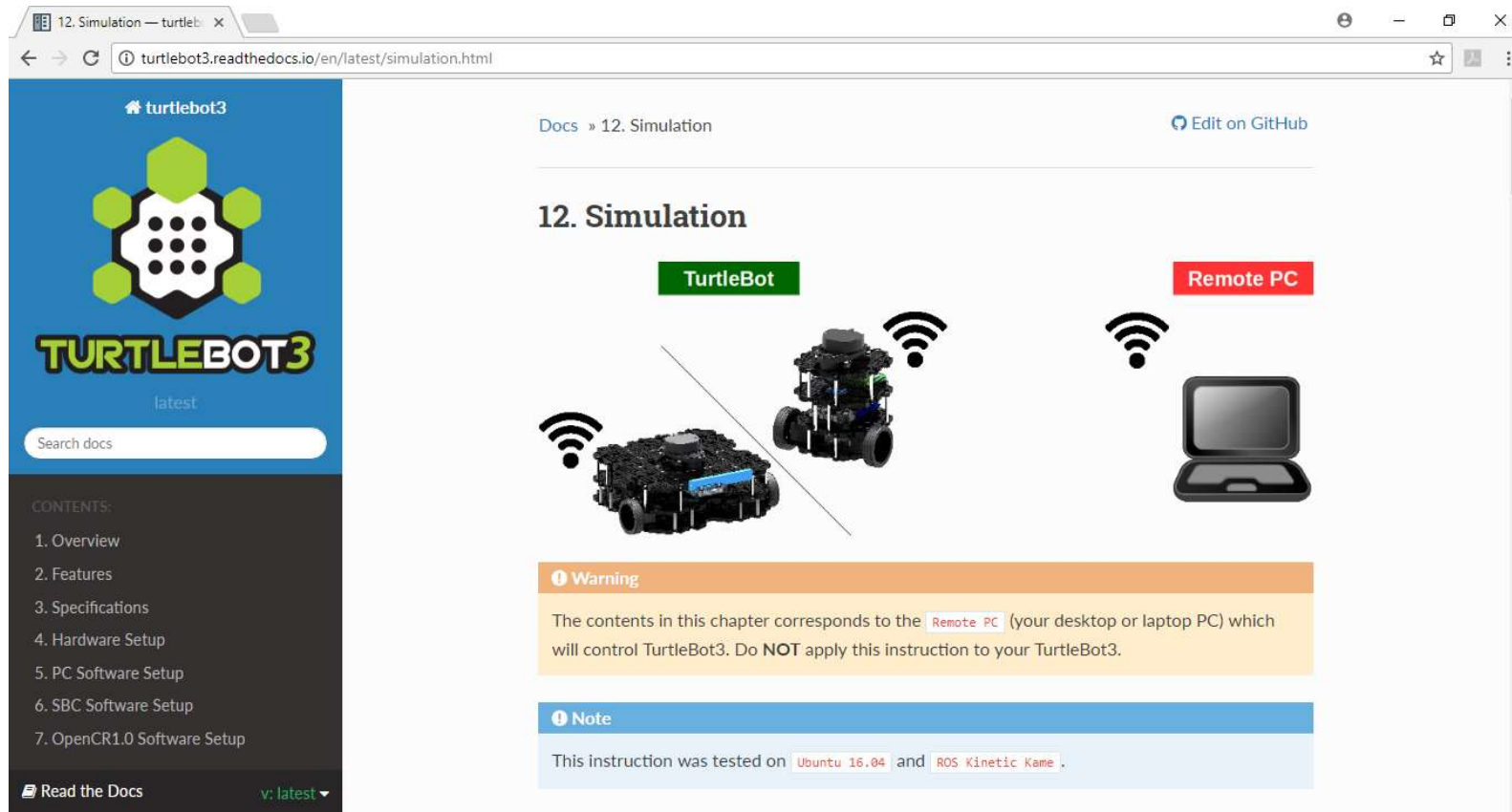
*Corso di Robotica*  
*Parte di Laboratorio*

Docente:  
**Domenico Daniele Bloisi**



Ottobre 2017

# Esercitazione



The screenshot shows a web browser window displaying the TurtleBot3 documentation page for simulation. The page title is "12. Simulation" and the URL is "turtlebot3.readthedocs.io/en/latest/simulation.html". The page features a navigation sidebar on the left with a search bar and a table of contents. The main content area includes a diagram showing a TurtleBot3 robot connected to a Remote PC via Wi-Fi. Below the diagram, there are two callout boxes: a "Warning" box stating that the instructions are for a Remote PC and not for the physical robot, and a "Note" box indicating the instructions were tested on Ubuntu 16.04 and ROS Kinetic Kame.

Docs » 12. Simulation [Edit on GitHub](#)

## 12. Simulation

**TurtleBot** **Remote PC**

**Warning**

The contents in this chapter corresponds to the **Remote PC** (your desktop or laptop PC) which will control TurtleBot3. Do **NOT** apply this instruction to your TurtleBot3.

**Note**

This instruction was tested on **Ubuntu 16.04** and **ROS Kinetic Kame**.

[Read the Docs](#) v: latest

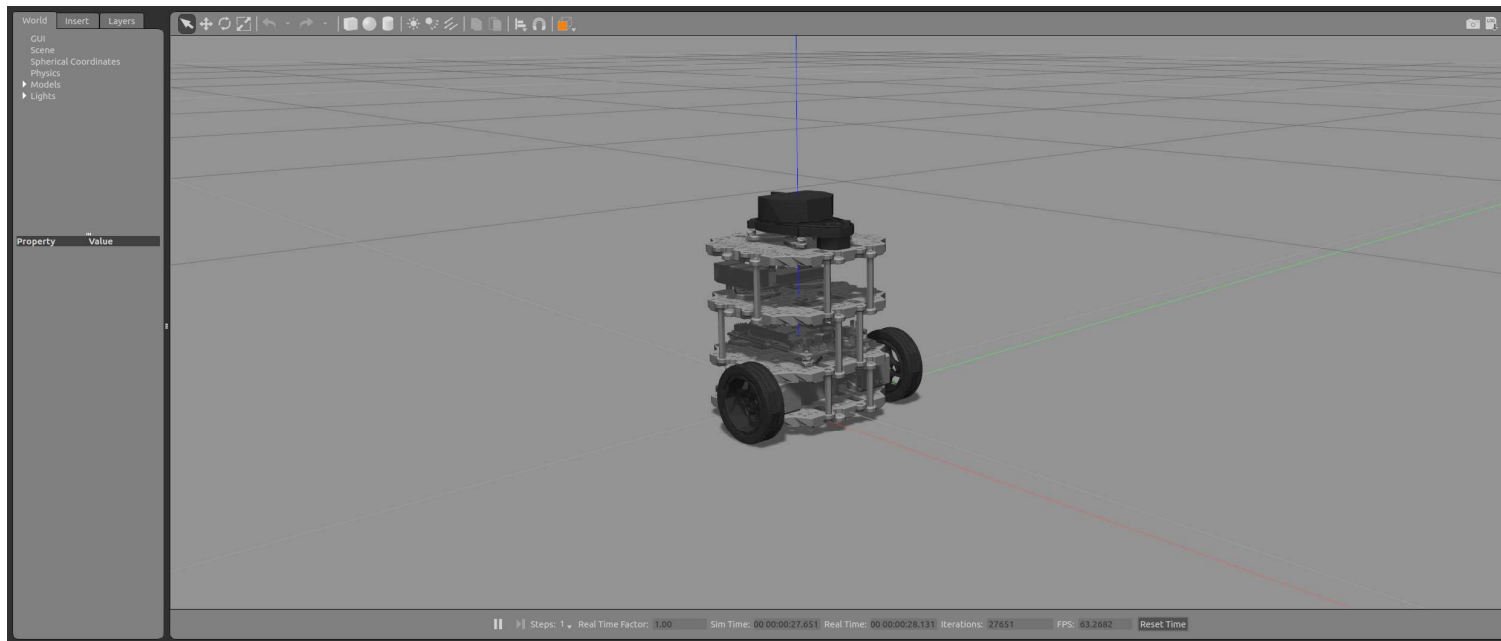
<http://turtlebot3.readthedocs.io/en/latest/simulation.html>

# Esercitazione

---

Testare i comandi per lanciare Gazebo

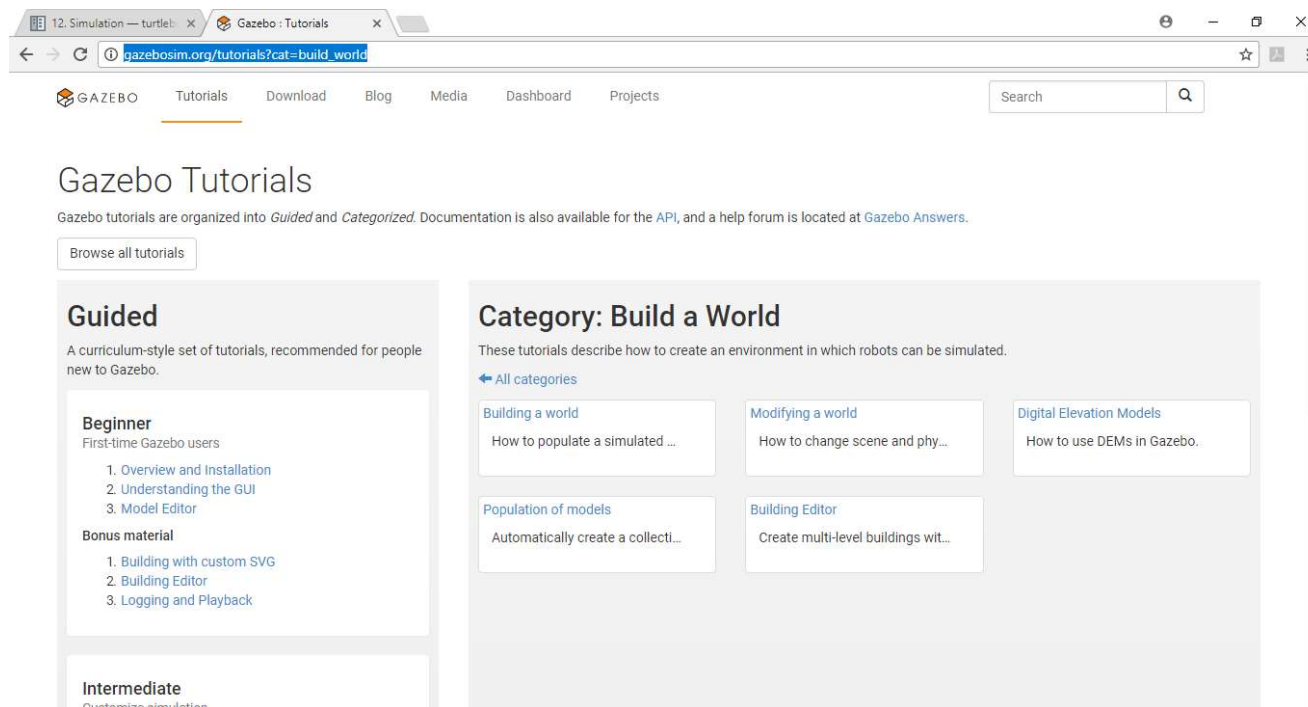
```
export TURTLEBOT3_MODEL=waffle  
roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
```



# Esercitazione

Modificare l'ambiente `empty_world` con l'aggiunta di oggetti (tavoli e sedie) per riprodurre il laboratorio ciberfisico

[http://gazebo.org/tutorials?cat=build\\_world](http://gazebo.org/tutorials?cat=build_world)



The screenshot shows a web browser window with the URL `gazebo.org/tutorials?cat=build_world`. The page title is "Gazebo Tutorials" and it features a navigation menu with links for "Tutorials", "Download", "Blog", "Media", "Dashboard", and "Projects". A search bar is located in the top right corner. The main content area is divided into two columns. The left column is titled "Guided" and contains a "Beginner" section with a list of tutorials: "1. Overview and Installation", "2. Understanding the GUI", and "3. Model Editor". Below this is a "Bonus material" section with "1. Building with custom SVG", "2. Building Editor", and "3. Logging and Playback". The right column is titled "Category: Build a World" and contains a list of tutorials: "Building a world", "Modifying a world", "Digital Elevation Models", "Population of models", and "Building Editor".

Gazebo Tutorials

Gazebo tutorials are organized into *Guided* and *Categorized*. Documentation is also available for the *API*, and a help forum is located at [Gazebo Answers](#).

[Browse all tutorials](#)

### Guided

A curriculum-style set of tutorials, recommended for people new to Gazebo.

#### Beginner

First-time Gazebo users

1. Overview and Installation
2. Understanding the GUI
3. Model Editor

#### Bonus material

1. Building with custom SVG
2. Building Editor
3. Logging and Playback

#### Intermediate

Customize simulation

### Category: Build a World

These tutorials describe how to create an environment in which robots can be simulated.

[← All categories](#)

- [Building a world](#)  
How to populate a simulated ...
- [Modifying a world](#)  
How to change scene and phy...
- [Digital Elevation Models](#)  
How to use DEMs in Gazebo.
- [Population of models](#)  
Automatically create a collecti...
- [Building Editor](#)  
Create multi-level buildings wit...

# Esercitazione

---

## Schema di soluzione

Una volta creato il file del modello (per esempio lab.world) inserirlo in  
`~/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/models`

Poi creare nella cartella

`~/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch`  
un nuovo launch file, in modo che possa essere lanciato il modello del lab con il comando

```
roslaunch turtlebot3_gazebo turtlebot3_lab_world.launch
```

**Fare attenzione a modificare anche i file in**

`/opt/ros/kinetic/share/gazebo_ros/launch/`