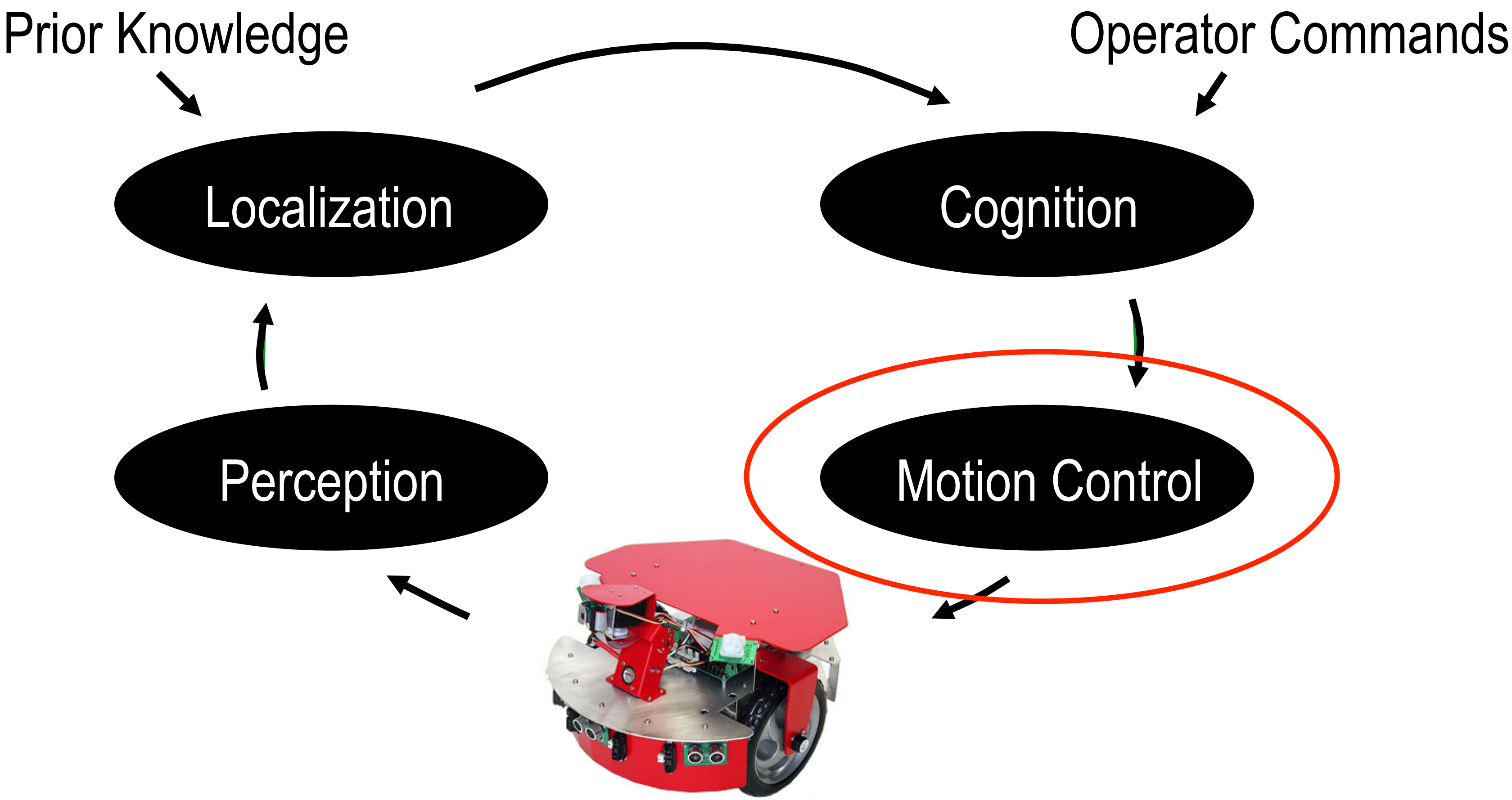
COS 495 - Lecture 3
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Fall 2011

Figures courtesy of Siegwart & Nourbakhsh

Control Structure



Locomotion & Robot Representations

1. Locomotion
    1. Legged Locomotion
    2. Snake Locomotion
    3. Free-Floating Motion
    4. Wheeled Locomotion
2. Continuous Representations
3. Forward Kinematics

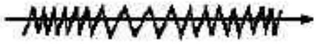
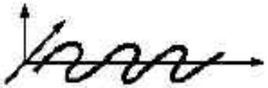
Locomotion

- Locomotion is the act of moving from place to place.
- Locomotion relies on the physical interaction between the vehicle and its environment.
- Locomotion is concerned with the interaction forces, along with the mechanisms and actuators that generate them.

Locomotion - Issues

- Stability
  - Number of contact points
  - Center of gravity
  - Static versus Dynamic stabilization
  - Inclination of terrain
- Contact
  - Contact point or area
  - Angle of contact
  - Friction
- Environment
  - Structure
  - Medium

Locomotion in Nature

Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel 	Hydrodynamic forces	Eddies 
Crawl 	Friction forces	 Longitudinal vibration
Sliding 	Friction forces	 Transverse vibration
Running 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Jumping 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Walking 	Gravitational forces	Rolling of a polygon (see figure 2.2) 

Locomotion in Robots

- Many locomotion concepts are inspired by nature
- Most natural locomotion concepts are difficult to imitate technically
- Rolling, which is NOT found in nature, is most efficient

Locomotion in Robots: Examples

- Locomotion via Climbing

Courtesy of T. Bretl

Locomotion in Robots: Examples

- Locomotion via Hopping

Courtesy of S. Martel

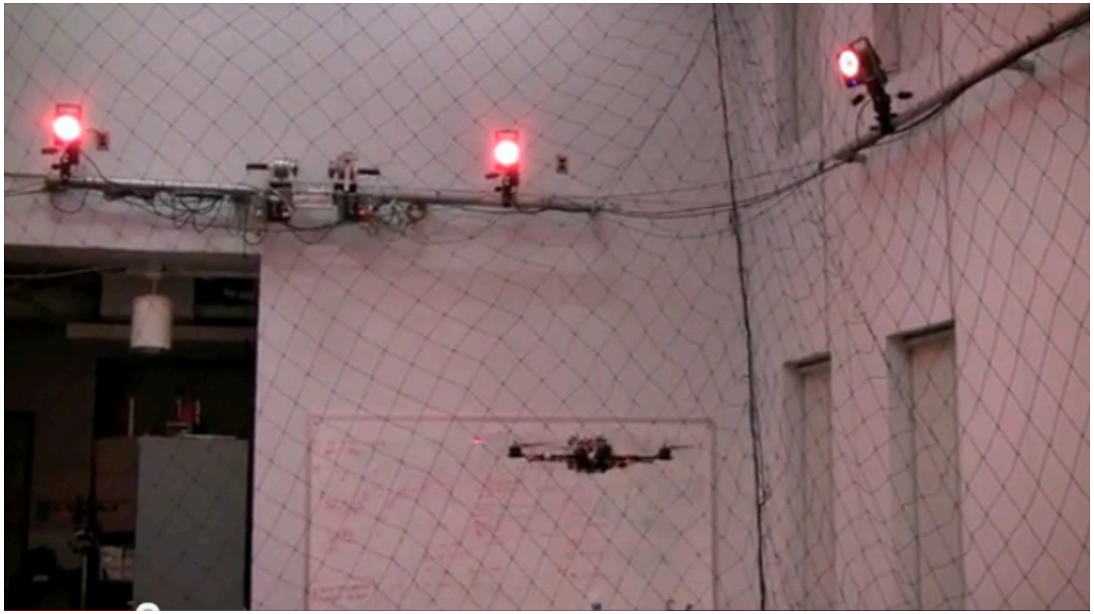
Locomotion in Robots: Examples

- Locomotion via Sliding

Courtesy of G. Miller

Locomotion in Robots: Examples

- Locomotion via Flying



GRASP Lab, Univ. of Pennsylvania

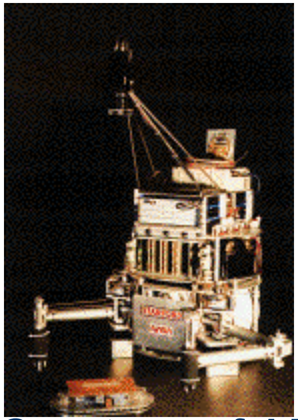
Locomotion in Robots: Examples

- Locomotion via Self Reconfigurable Robots

Courtesy of USC

Locomotion in Robots: Examples

- Other types of motion



Courtesy of ARL, Stanford

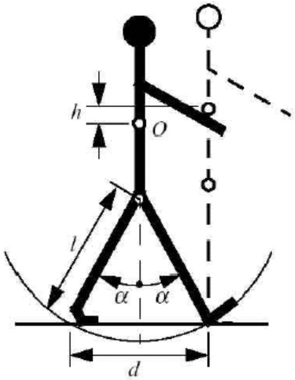
Courtesy of S. Martel

Locomotion in Robots: Examples

- Other types of motion

Legged Locomotion

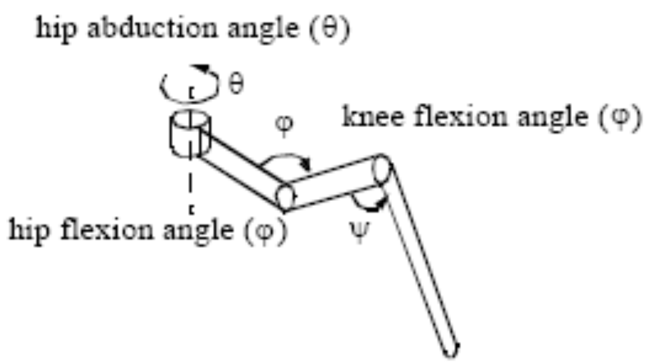
- Nature inspired.
- The movement of walking biped is close to rolling.

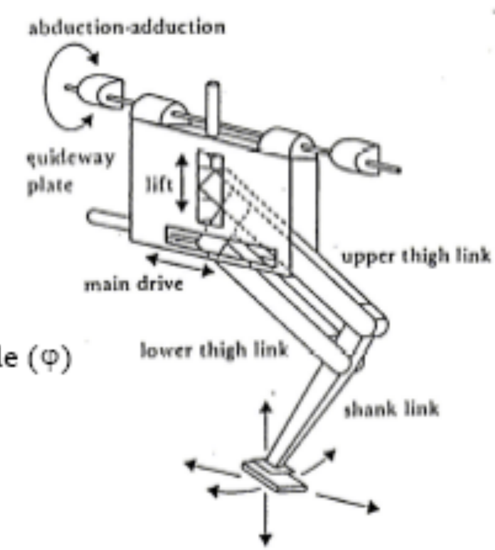


- Number of legs determines stability of locomotion

Legged Locomotion

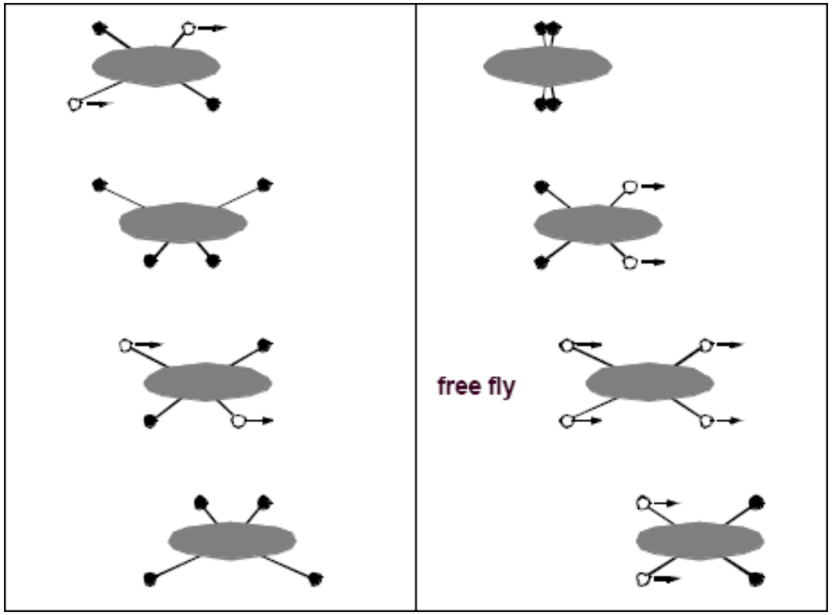
- Degrees of freedom per leg
  - Trade-off exists between complexity and stability
- Degrees of freedom per system
  - Too many, needed gaited motion





Legged Locomotion

- Walking gaits
  - The gait is the repetitive sequence of leg movements to allow locomotion
  - The gait is characterized by the sequence of lift and release events of individual legs.



Changeover Walking

Galloping

Legged Locomotion

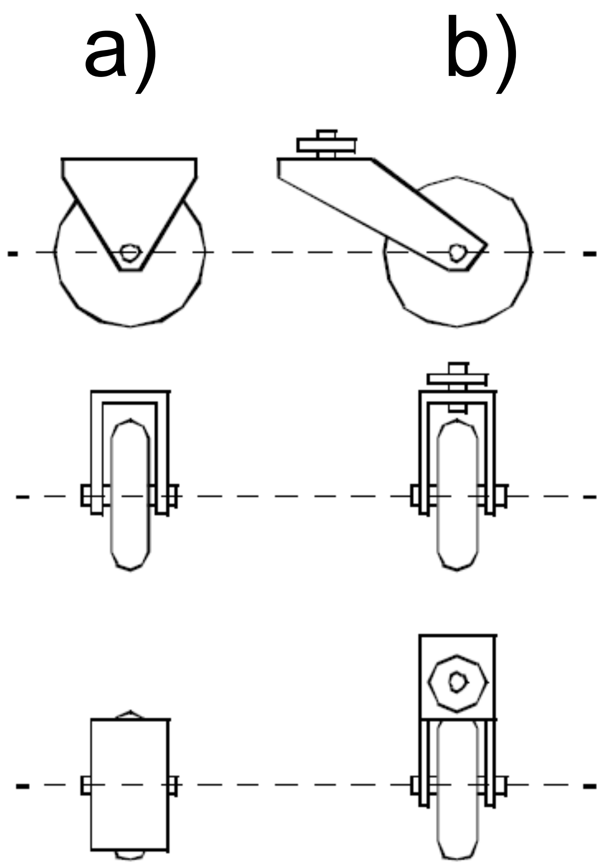
Courtesy of Pulstech

Wheeled Locomotion

- Wheel types

  a) Standard Wheel
  - 2 DOF

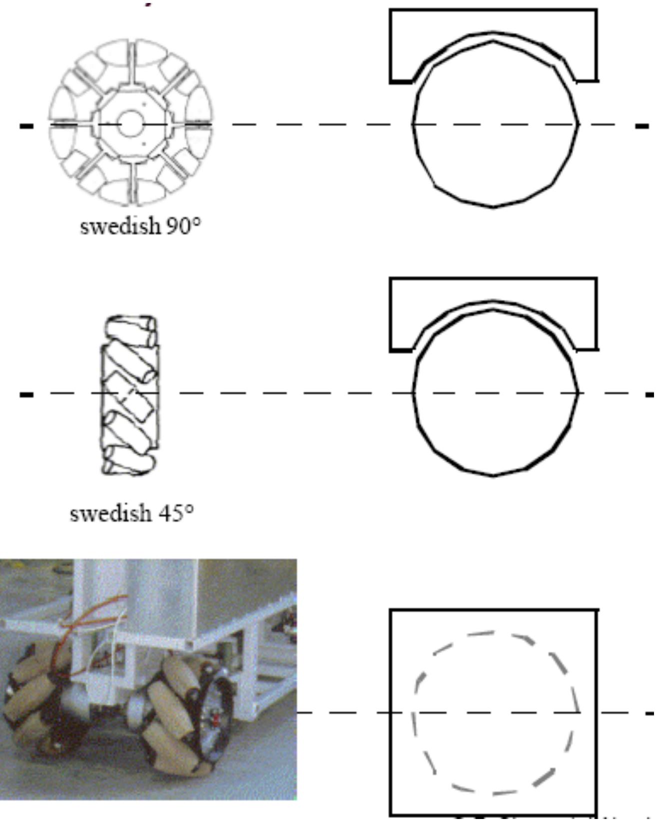
  b) Castor Wheel
  - 3 DOF



Wheeled Locomotion

- Wheel types

c) d)



  c) Swedish Wheel
  - 3 DOF

  d) Spherical Wheel
  - Technically difficult

Wheeled Locomotion

- Wheel Arrangements
  - Three issues: *Stability*, *Maneuverability* and *Controllability*
  - Stability is guaranteed with 3 wheels, improved with four.
  - Tradeoff between Maneuverability and Controllability
    - Combining actuation and steering on one wheel increases complexity and adds positioning errors

Locomotion & Robot Representations

1. Locomotion
2. Continuous Representations
    1. Global Coordinate Frames
    2. Local Coordinate Frames
    3. Transformations
3. Forward Kinematics

Continuous Representations

- To control a robot we need to represent the robot's state with some quantifiable variables.
- Given the state description, we model the motion of the robot with differential equations:

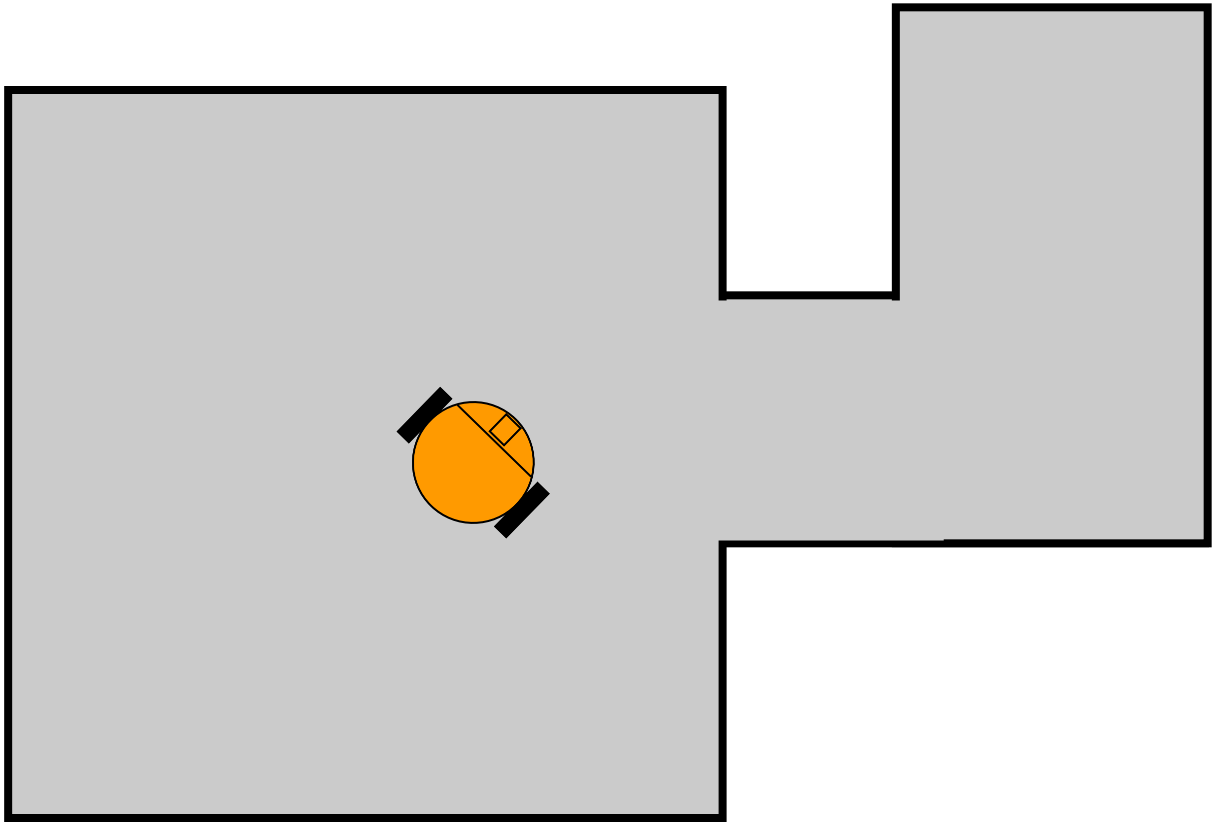
  Kinematics

- Once we have the Kinematics equations, we can develop a control law that will bring a robot to the desired location.

Continuous Representations

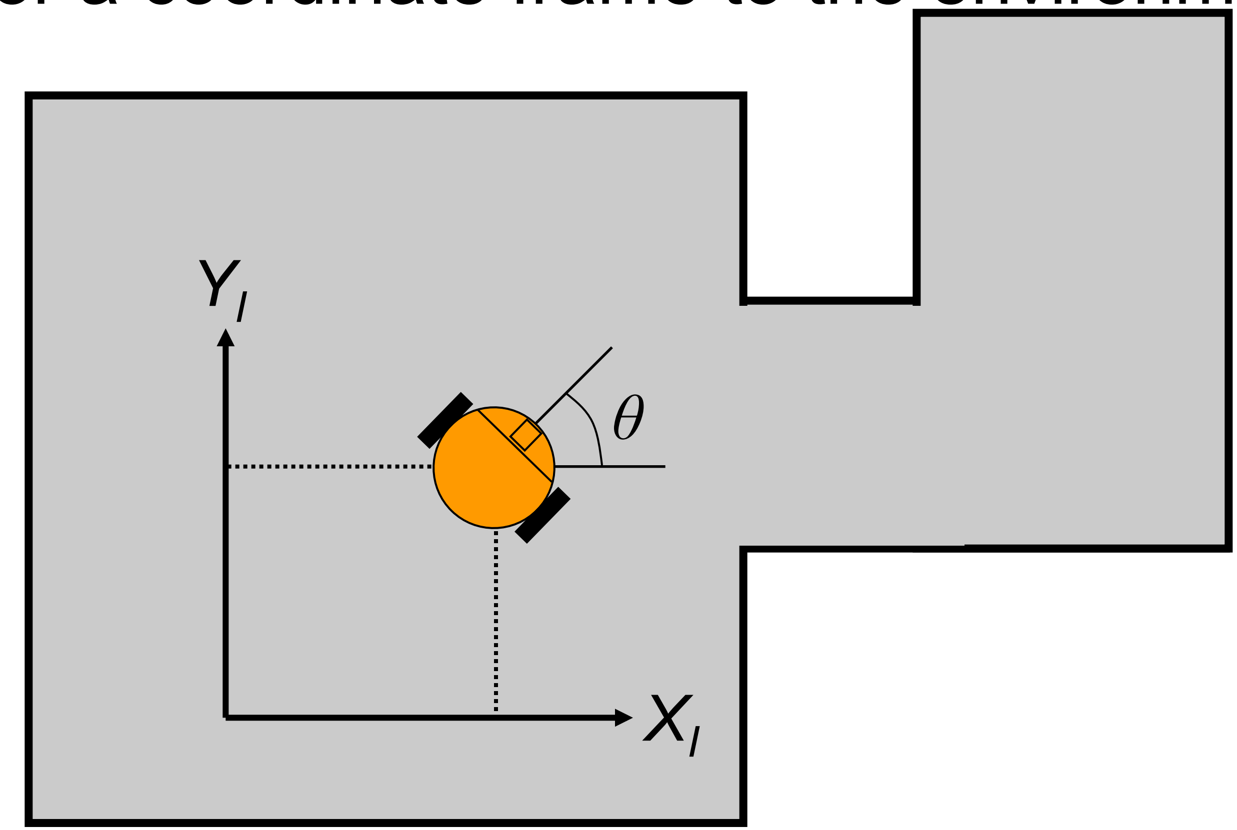
- To control a robot we need to represent the robot's state we use coordinate frames:
  - Global frame
  - Local frame

Global (Inertial) Coordinate frame



Global (Inertial) Coordinate frame

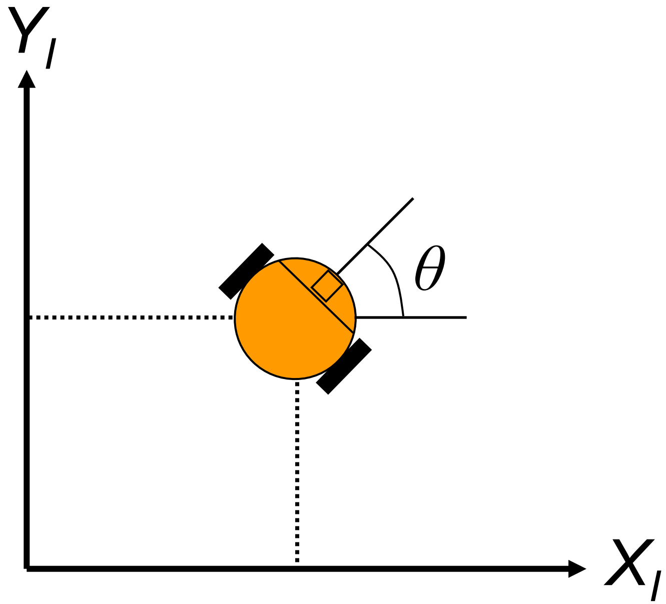
- Anchor a coordinate frame to the environment



Global (Inertial) Coordinate frame

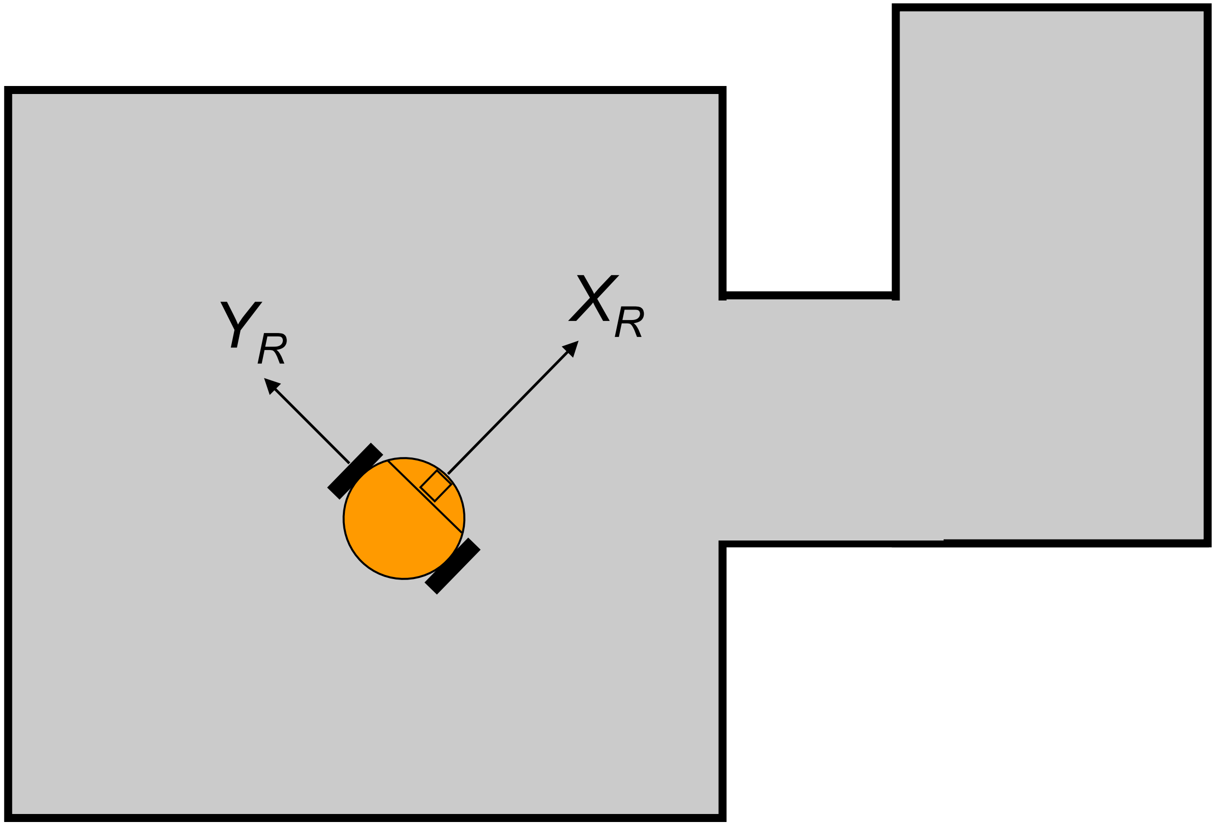
- With this coordinate frame, we describe the robot state as:

$$\xi_I = [x\ y\ \theta]_I$$



Local Coordinate frame

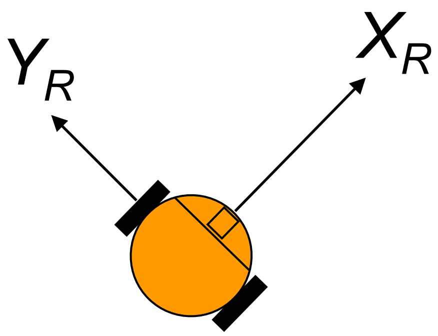
- Anchor a coordinate frame to the robot



Local Coordinate frame

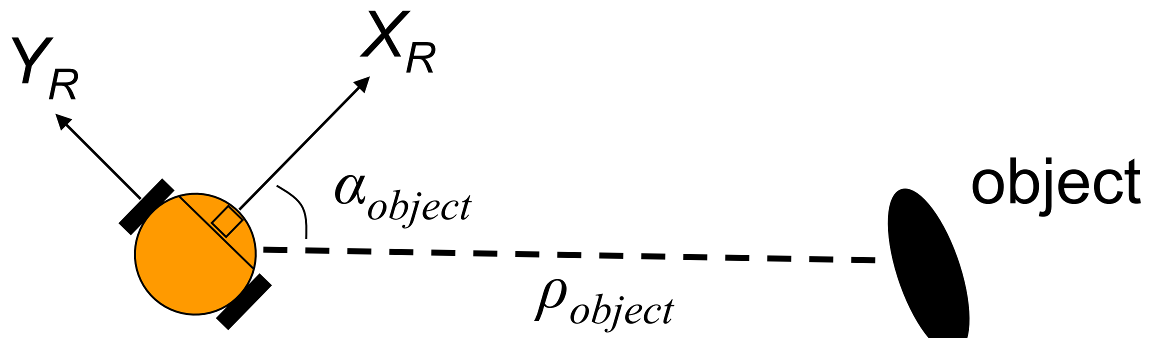
- With this coordinate frame, we describe the robot state as:

$$\xi_R = [x\ y\ \theta]_R = [0\ 0\ 0]$$



Local Coordinate frame

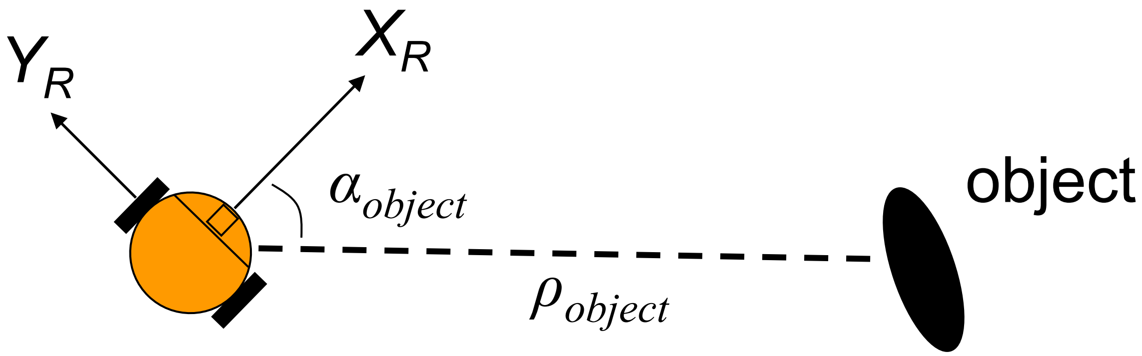
- The local frame is useful when considering taking measurements of environment objects.
  - Consider the detection of an wall using a range finder:



Local Coordinate frame

- The measurement is taken relative to the robot's local coordinate frame $(\rho_{object}, \alpha_{object})$
- We can calculate the position of the measurement in local coordinate frames:

$$x_{object, R} = \rho_{object} \cos(\alpha_{object})$$
$$y_{object, R} = \rho_{object} \sin(\alpha_{object})$$



Local Coordinate frame

- The local frame is also useful when considering velocity states:

$$d\xi_R/dt = [dx/dt\ dy/dt\ d\theta/dt]_R$$

$$= [\ \dot{x}\ \ \dot{y}\ \ \dot{\theta}\ ]_R$$

$$= \dot{\xi}_R$$

Local Coordinate frame

- Often we know the velocities of the robot in the local coordinate frame:

$$\dot{x} = v$$

$$\dot{y} = 0$$

$$\dot{\theta} = w$$

Transformations

- We are also interested in the robot's velocities with respect to the global frame.
- To calculate these, we need to consider the transformation *R* between the two frames:

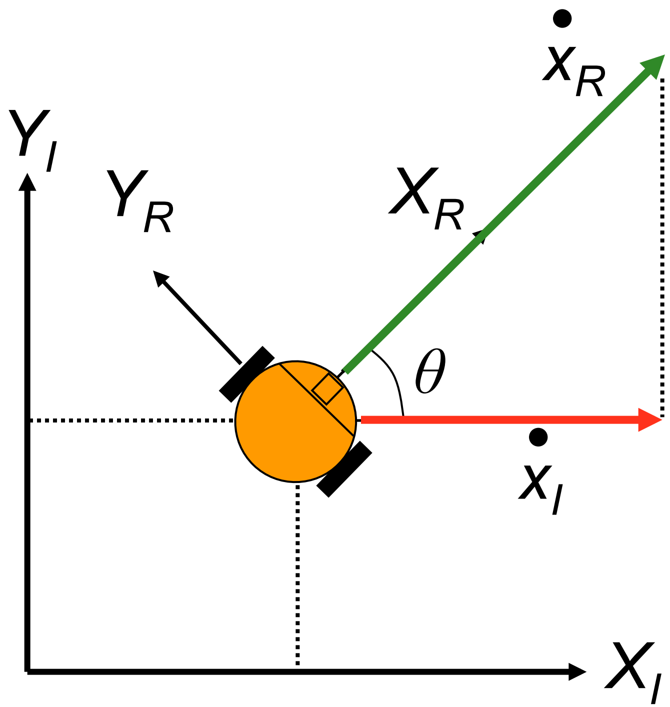
$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

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

- Note that *R* is a function of theta, the relative angle between the two frames.

Transformations

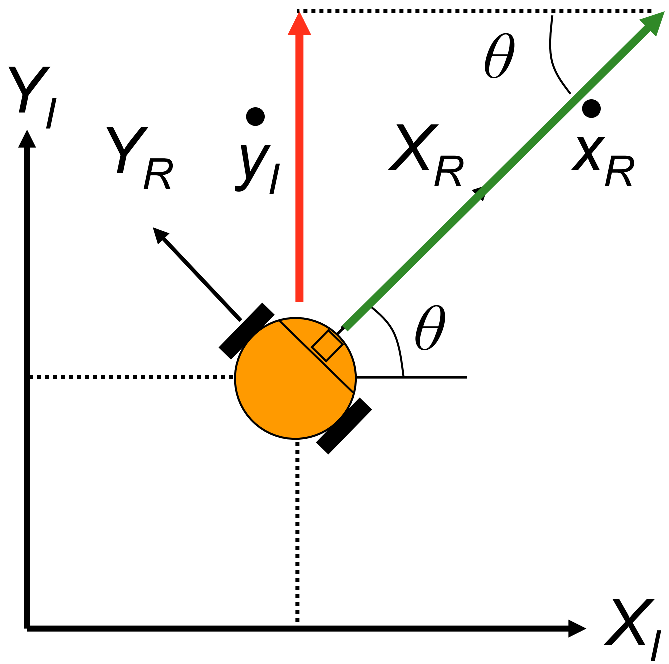
- Let's obtain the transformation matrix, starting with the X_I direction:



$$\dot{x}_I = \dot{x}_R \cos(\theta)$$

Transformations

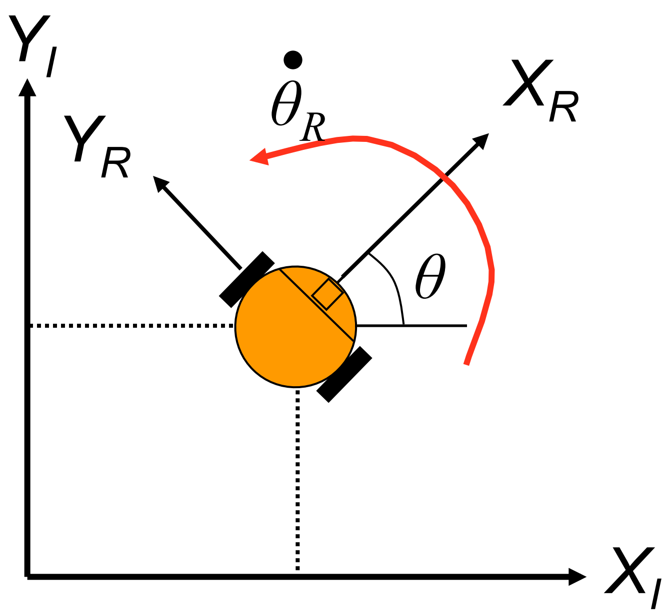
- Now the Y_I direction:



$$\dot{y}_I = \dot{x}_R \sin(\theta)$$

Transformations

- What about rotational velocity?



$$\dot{\theta}_I = \dot{\theta}_R$$

Transformations

- Lets put our equations in matrix form:

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix}$$

Transformations

- Lets put our equations in matrix form:

$$\underbrace{\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix}}_{\dot{\xi}_I} = \underbrace{\begin{pmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R(\theta)^{-1}} \underbrace{\begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix}}_{\dot{\xi}_R}$$

Transformations

- Or we can rewrite:

$$\dot{\xi}_I = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}$$

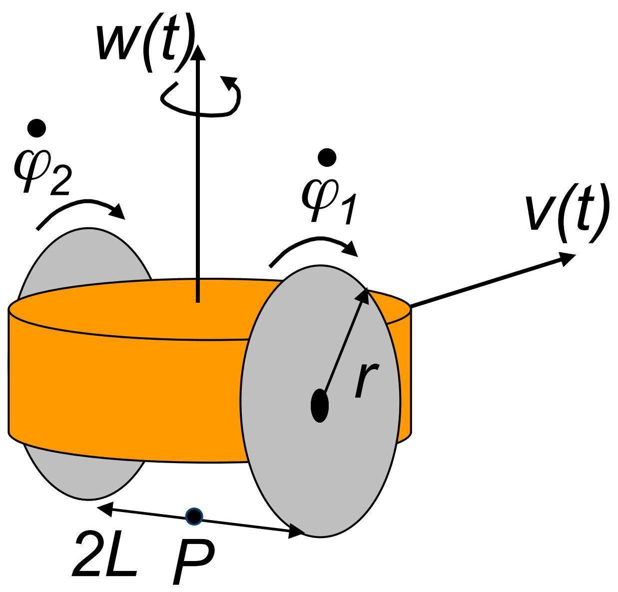
Locomotion & Robot Representations

1. Locomotion
2. Continuous Representations
3. Forward Kinematics

Kinematics

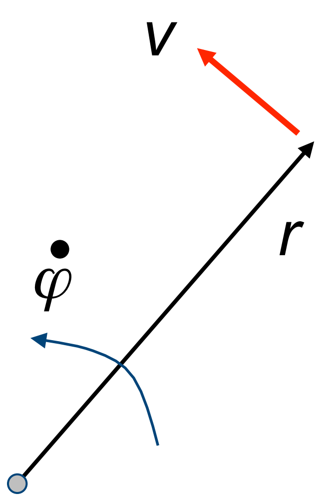
- The transformations we just defined form the basis of our forward Kinematics
  - The Kinematics equations should model how velocities in the global frame - $\dot{\xi}_I$, are a function of wheel speed inputs – $\dot{\varphi}_1$ and $\dot{\varphi}_2$.

Forward Kinematics



Forward Kinematics

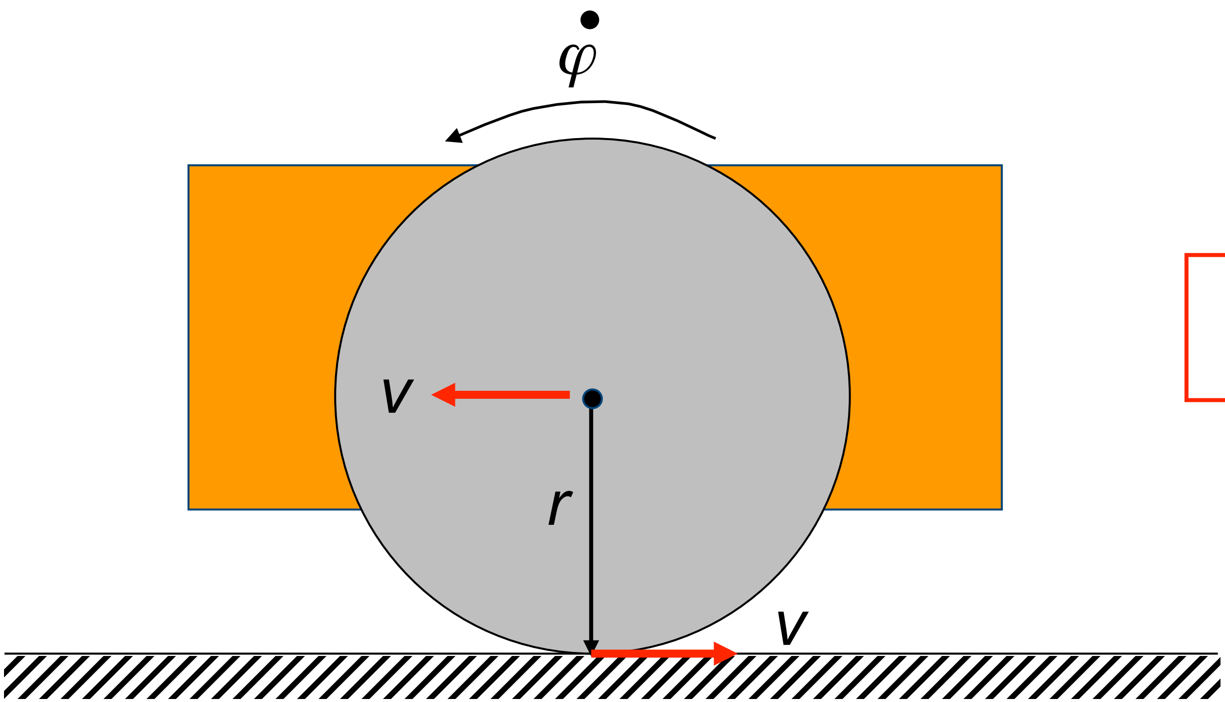
- Before we continue, we need to understand the relation between rotational velocity and forward velocity.



$$r\dot{\varphi} = v$$

Forward Kinematics

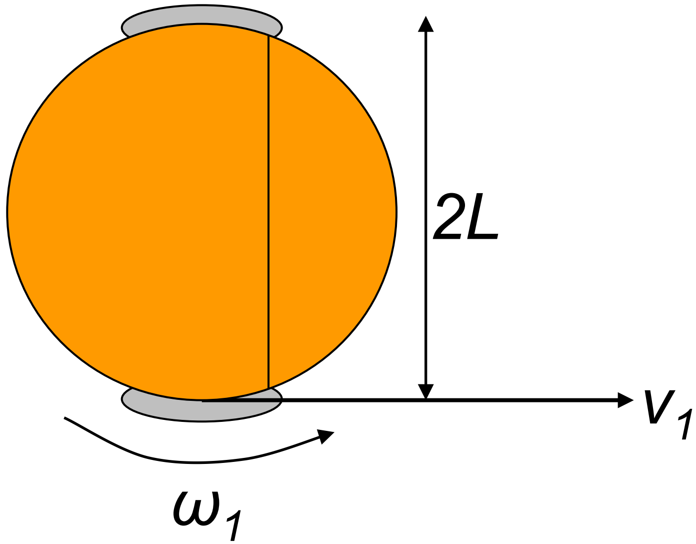
- Apply this to a wheel on the robot.



$$r\dot{\varphi} = v$$

Forward Kinematics

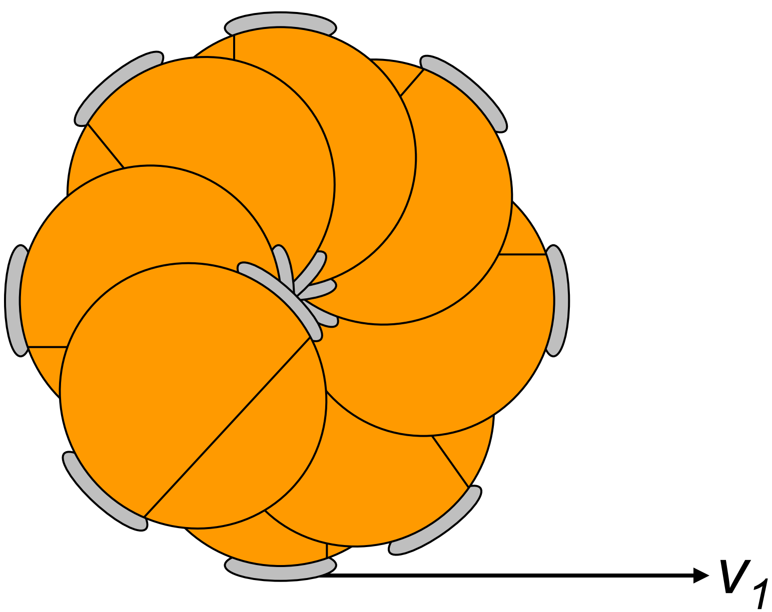
- Apply the same equation to a top view of the robot, assuming only wheel 1 is rotating.



$$v_1 = 2L\omega_1$$

Forward Kinematics

- Lets look in more detail:
  - If the left wheel has velocity 0, and right wheel has velocity *v*, the robot will spin with the left wheel acting as the center of rotation.

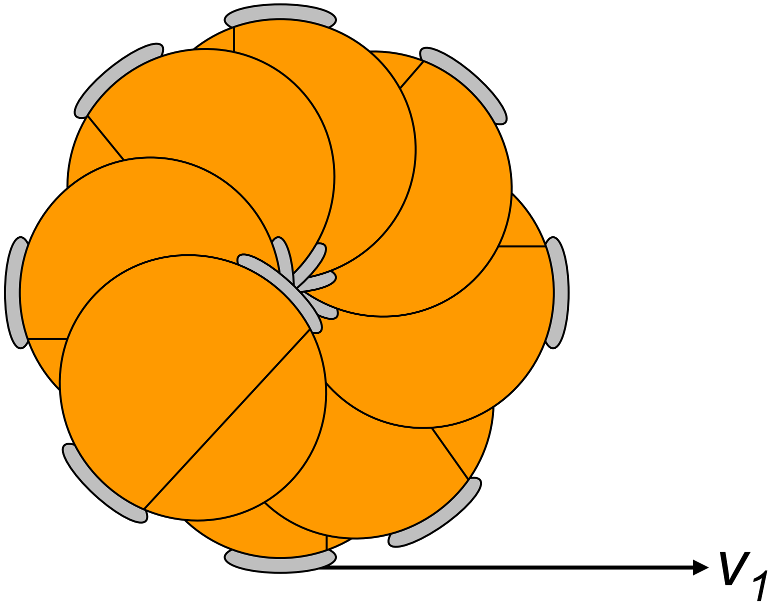


- There is no doubt that the wheel velocity induces a rotational velocity ω_1.
- The right wheel travels a distance $2\pi(2L)$ in *1* rotation.
- To make 1 full circle, it takes $2\pi(2L)/v_1$ seconds.
- The rotational velocity is then $(2\pi\ rad)$ / $(2\pi(2L)/v_1$ seconds)

Forward Kinematics

- So the rotational velocity induced by the right wheel is:

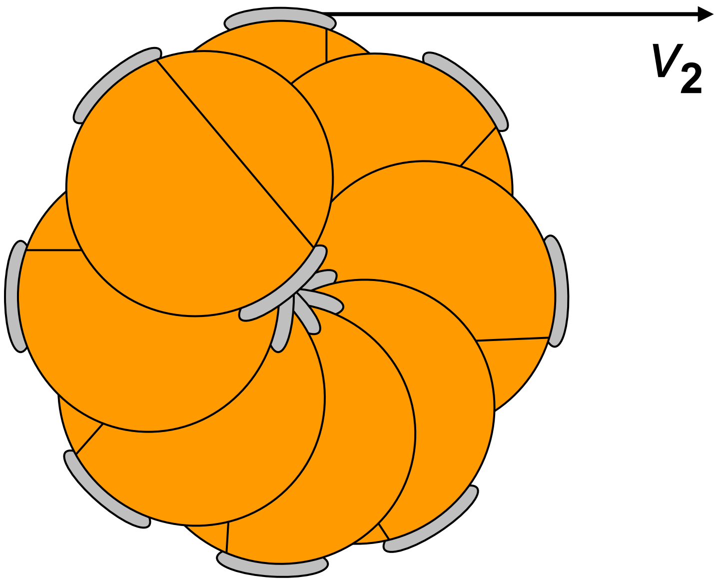
$$\omega_1 = v_1 / 2L \ rad/s$$



Forward Kinematics

- Similarly, the rotational velocity induced by the left wheel is:

$$\omega_2 = -v_2 / 2L \; rad/s$$



- Note the negative sign because forward wheel velocity induces a negative rotational velocity on the robot.

Forward Kinematics

- Now, substitute velocities v_1 and v_2 calculated from wheel speeds (slide 43) into the rotational velocity equations (slides 46, 47).

$$\omega_1 = \frac{r\dot{\varphi}_1}{2L}$$

$$\omega_2 = \frac{-r\dot{\varphi}_2}{2L}$$

Forward Kinematics

- Now, the rotational velocities can be calculated by summing the components of velocities from each wheel:

$$w(t) = \omega_1 + \omega_2$$

- The forward velocity is the sum of the two components, (i.e. average of 2 velocities) again using the same equation from slide 44:

$$v(t) = L(\omega_1 - \omega_2)$$

Transformations

- Recall:

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ 0 \\ w \end{pmatrix}$$

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

Forward Kinematics

- The resulting kinematics equation is:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{pmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2L} - \frac{r\dot{\varphi}_2}{2L} \end{pmatrix}$$

Forward Kinematics

- We now know how to calculate how wheel speeds affect the robot velocities in the global coordinate frame.
- This will be useful when we want to control the robot to track points (i.e. move to desired locations in the global coordinate frame by controlling wheel speeds).