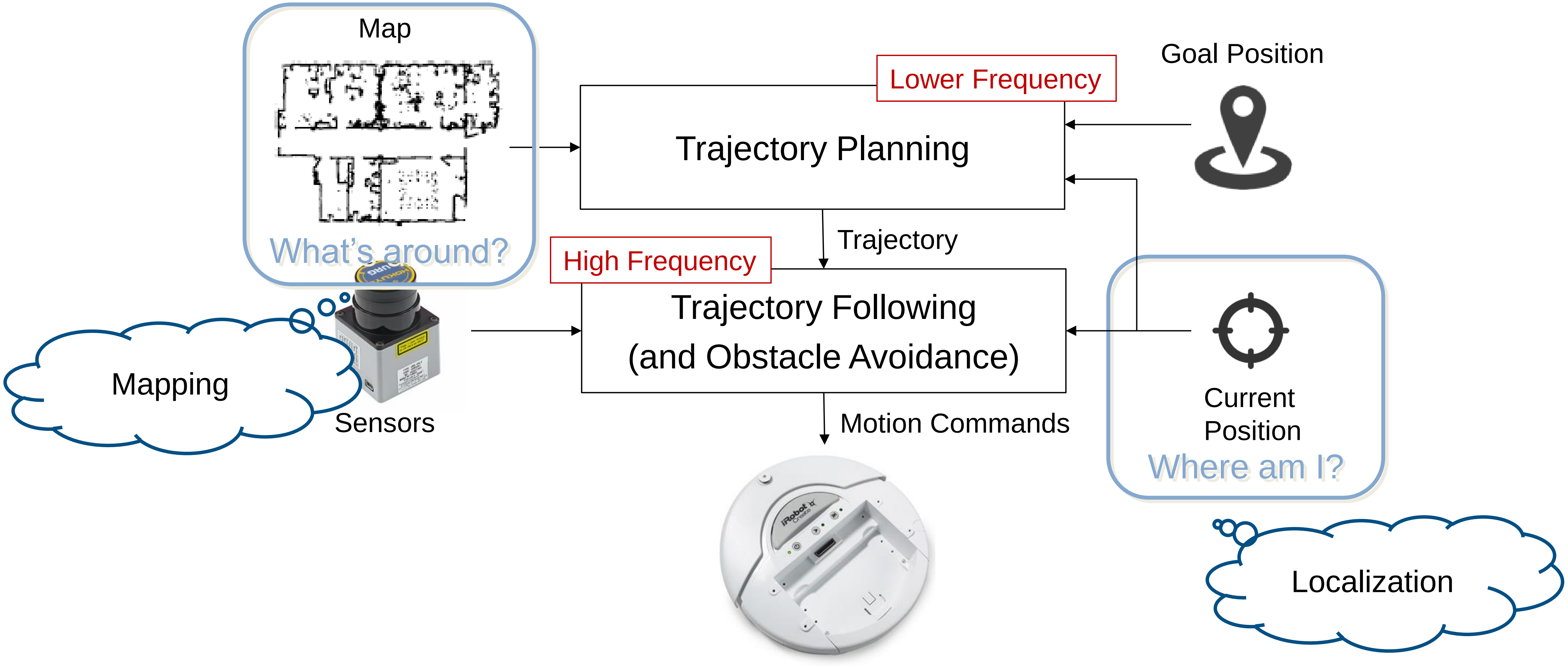
Robotics

Robot Localization – Wheels Odometry

Matteo Matteucci
matteo.matteucci@polimi.it

Artificial Intelligence and Robotics Lab - Politecnico di Milano

A Simplified Sense-Plan-Act Architecture



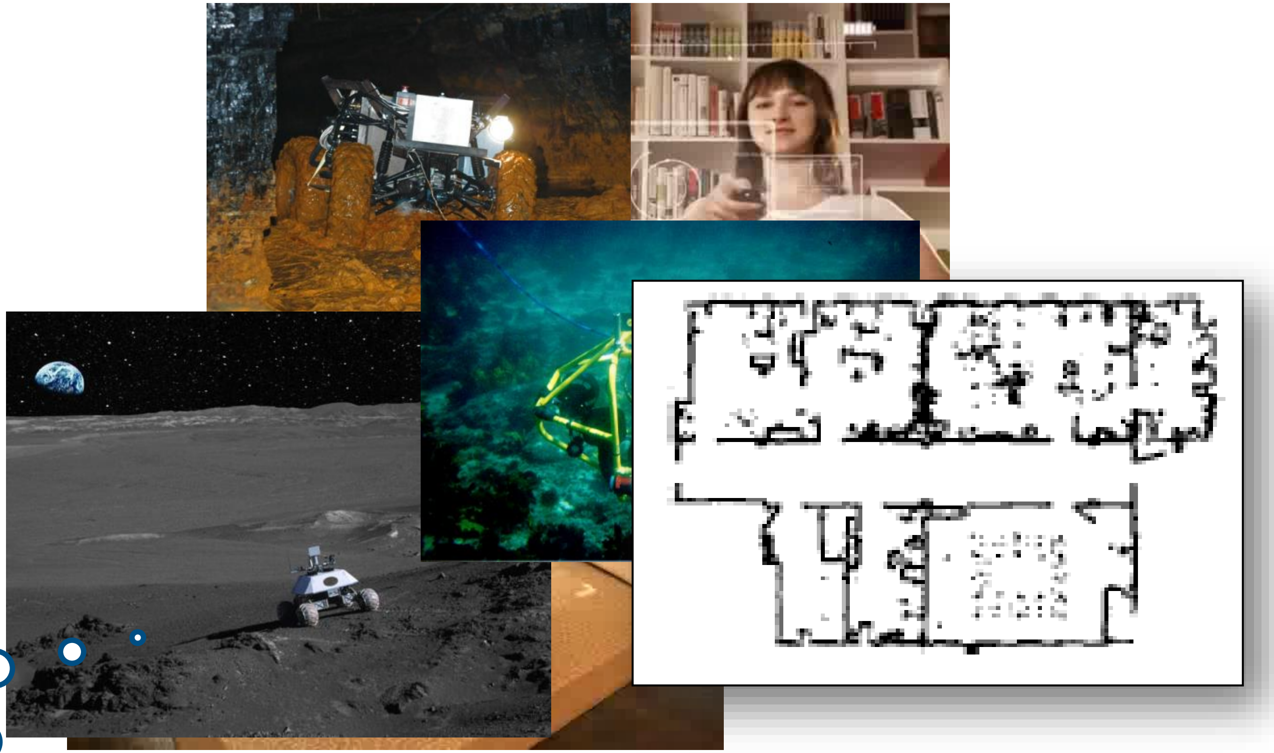
Where Am I?

To perform their tasks autonomous robots and unmanned vehicles need

- To know where they are (e.g., Global Positioning System)
- To know the environment map (e.g., Geographical Institutes Maps)

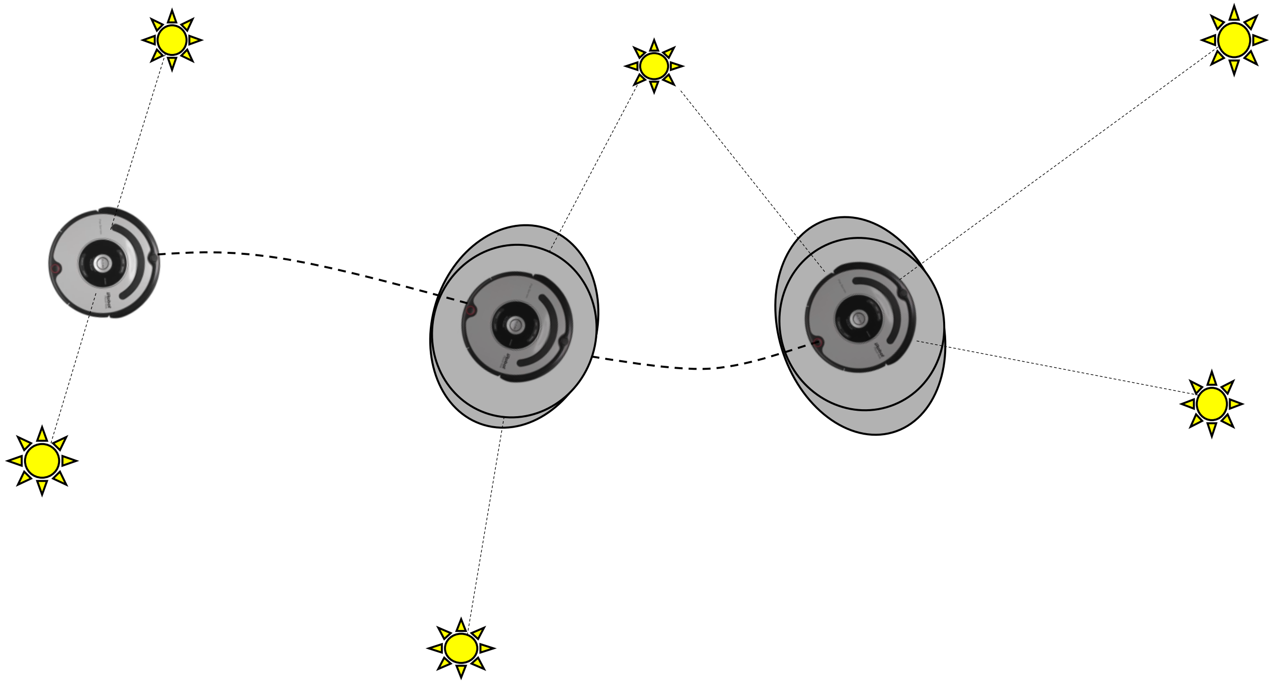
These are not always possible or reliable

- GNSS are not always reliable/available
- Not all places have been mapped
- Environment changes dynamically
- Maps need to be updated

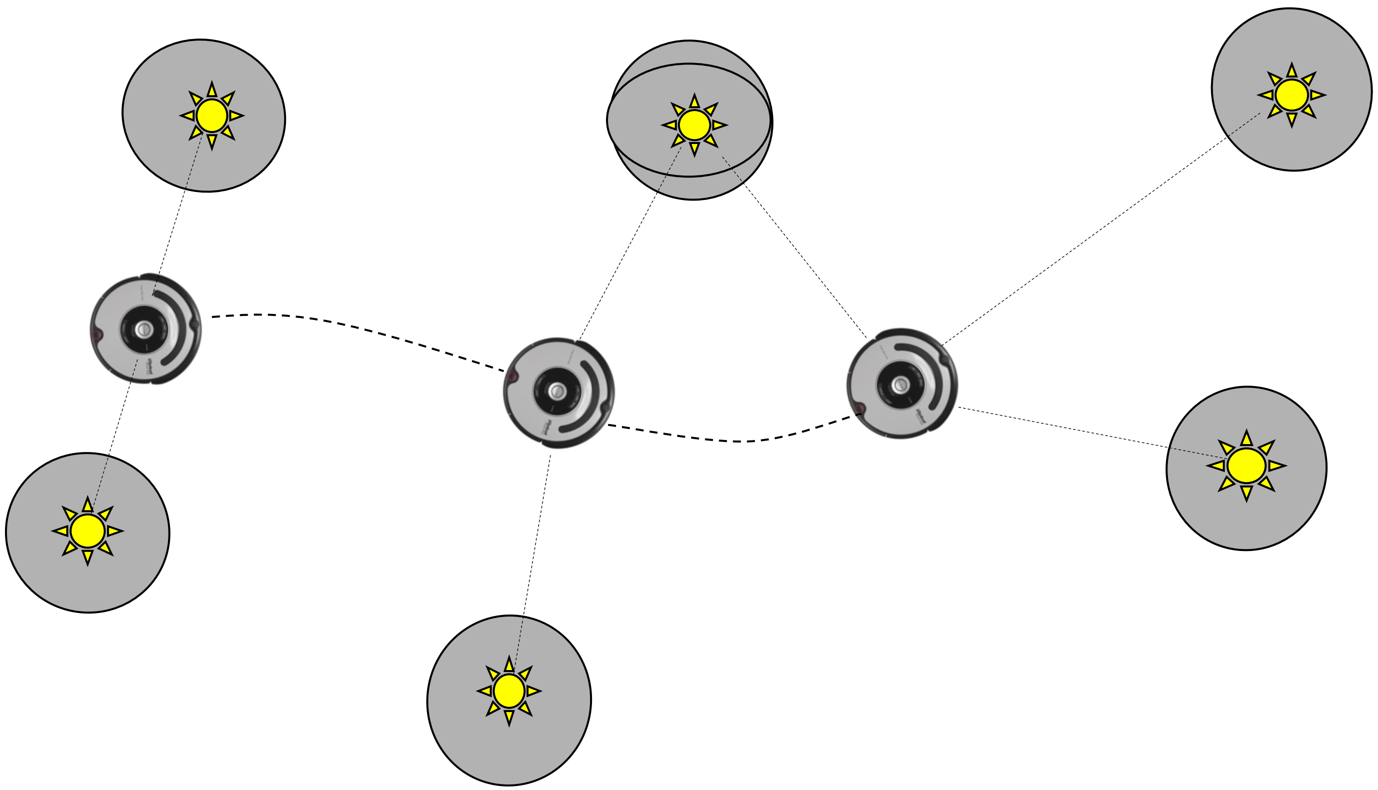


How does this robot's map look like?

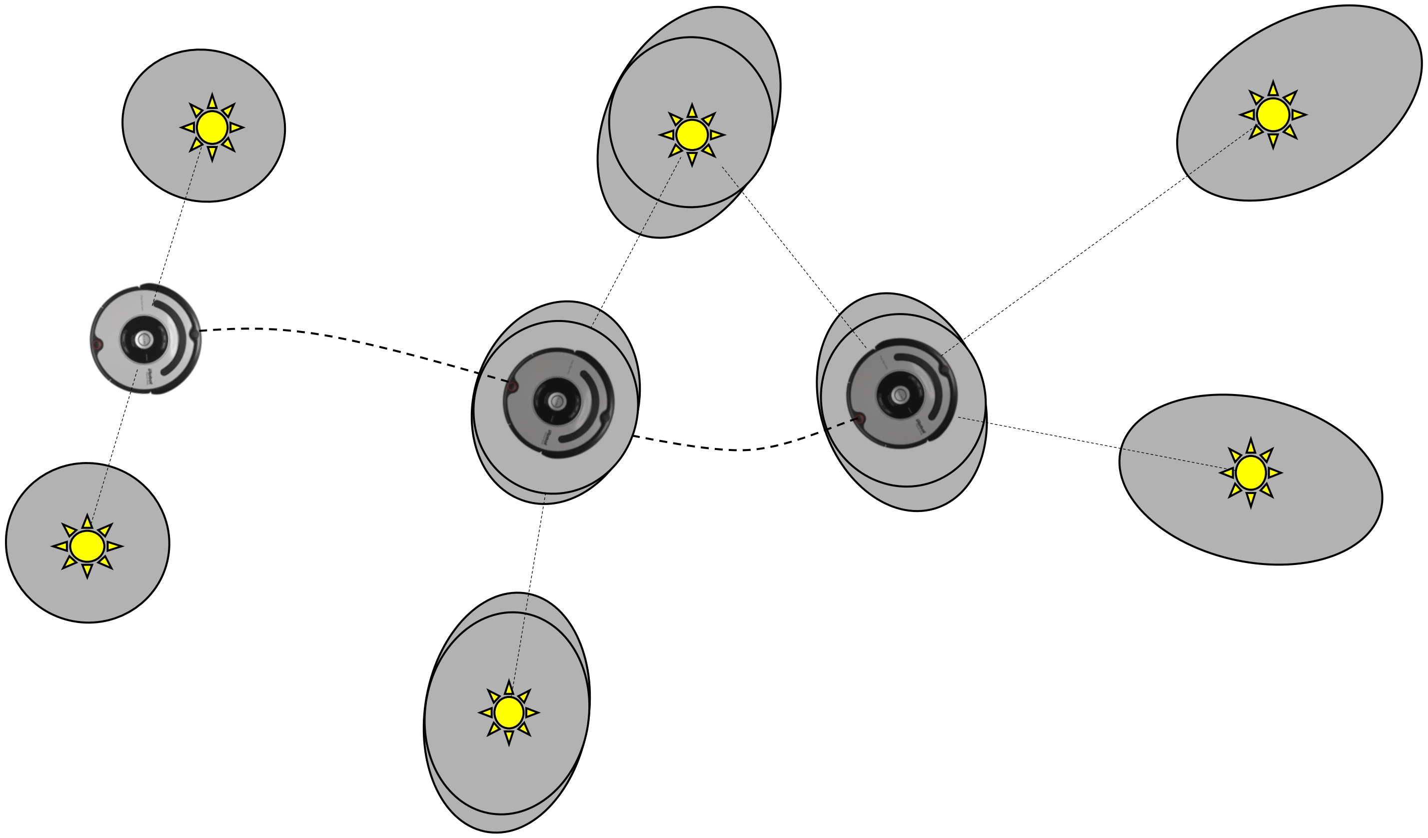
Localization with Known Map



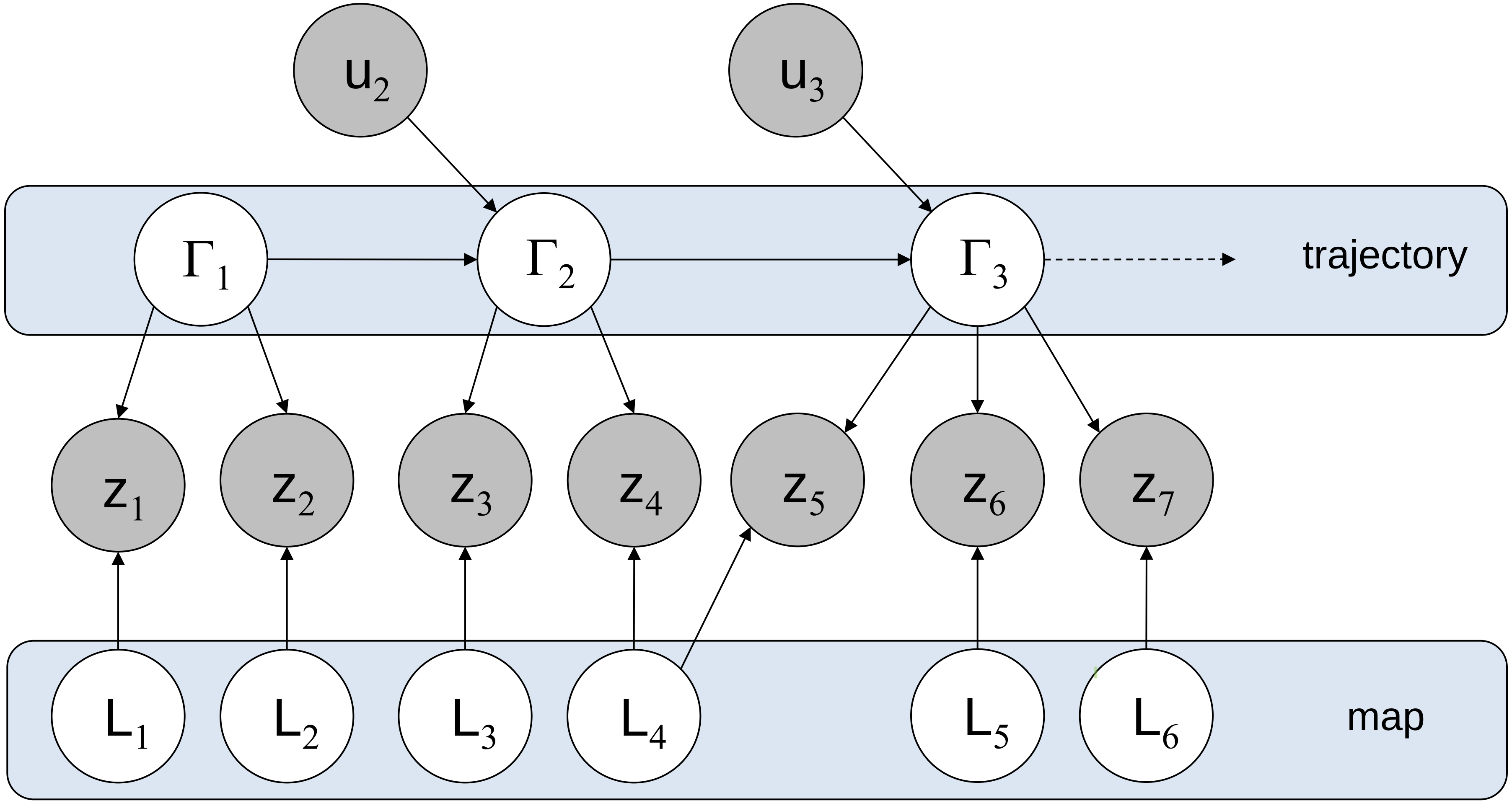
Mapping with Known Poses



Simultaneous Localization and Mapping

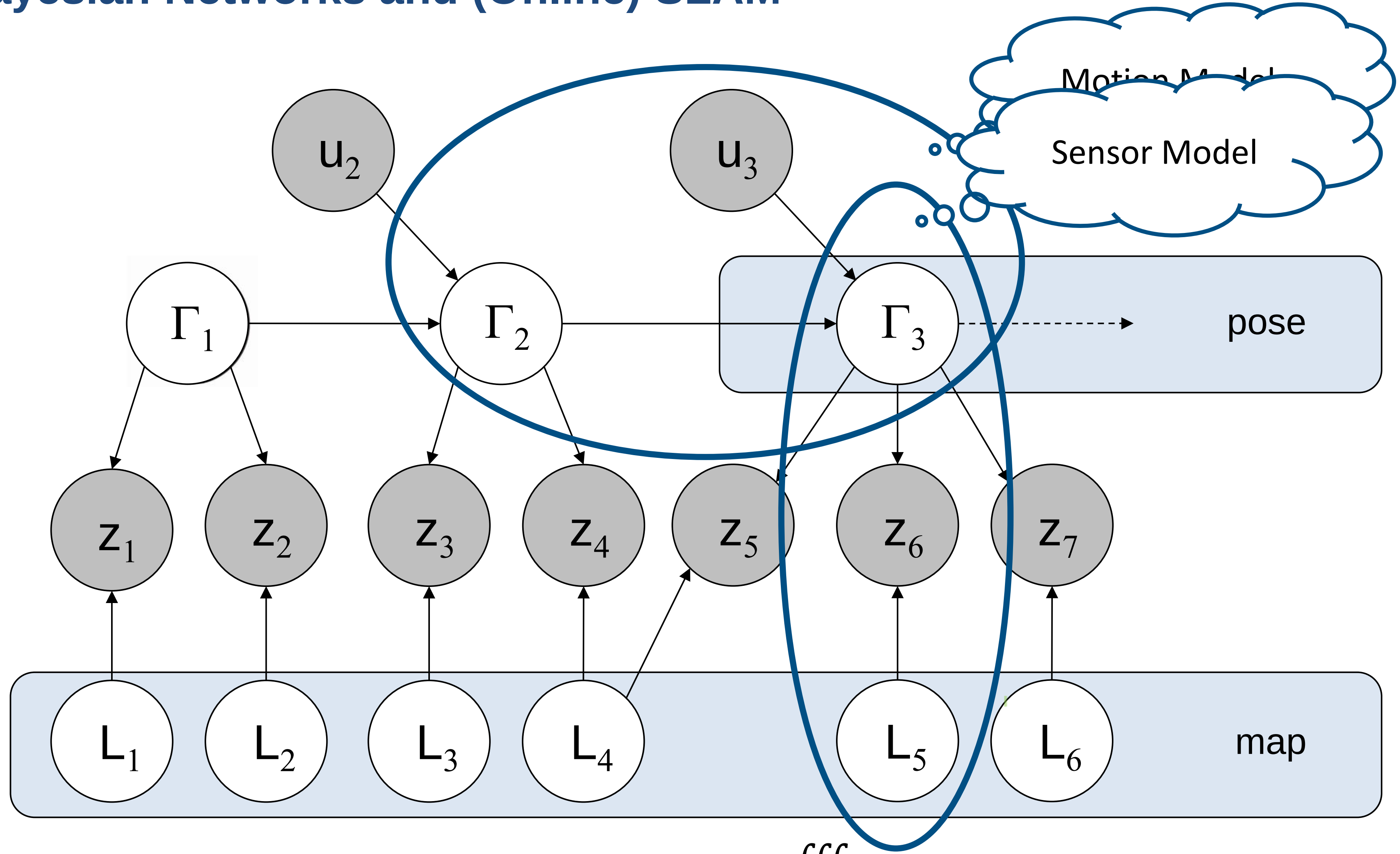


Dynamic Bayesian Networks and (Full) SLAM



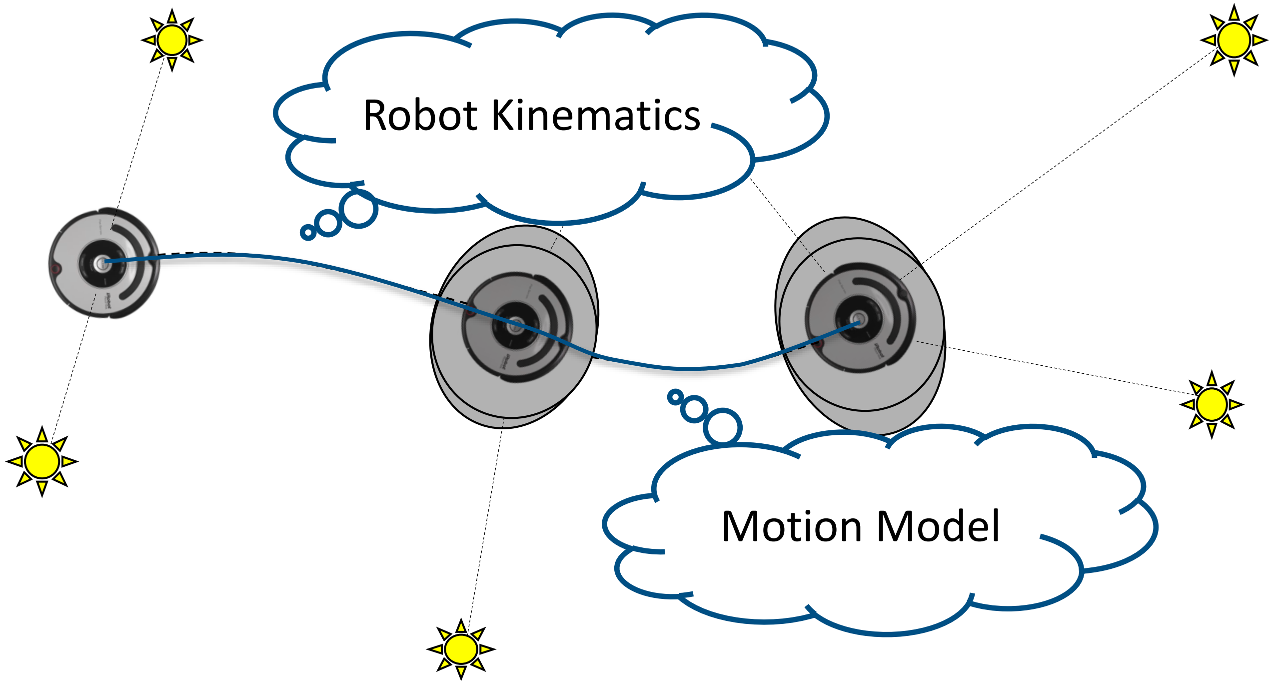
Smoothing: $p(\Gamma_{1:t}, l_1, \dots, l_N | Z_{1:t}, U_{1:t})$

Dynamic Bayesian Networks and (Online) SLAM



Filtering: $p(\Gamma_t, l_1, \dots, l_N | Z_{1:t}, U_{1:t}) = \iiint_{1:t-1} p(\Gamma_{1:t}, l_1, \dots, l_N | Z_{1:t}, U_{1:t})$

Localization with Known Map



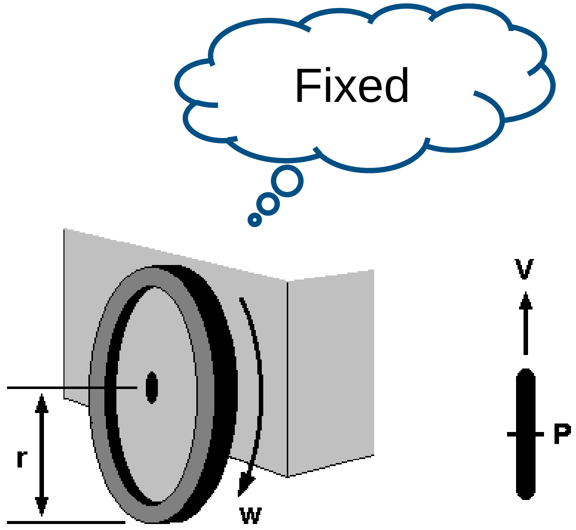
Wheeled Mobile Robots

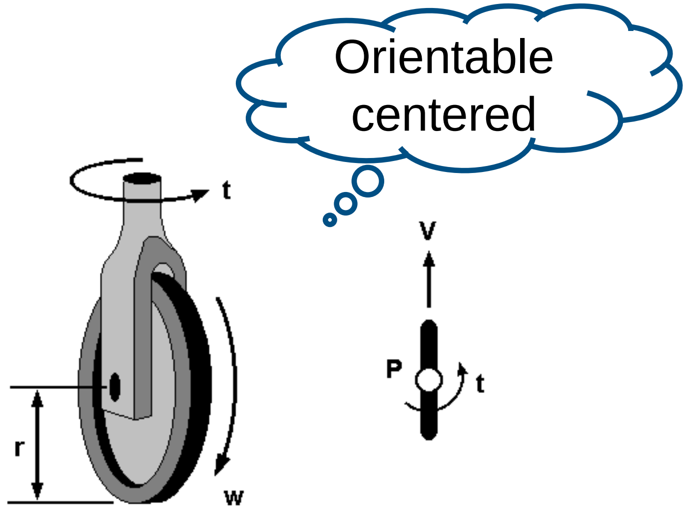
A robot capable of locomotion on a surface **solely through the actuation of wheel assemblies** mounted on the robot and in contact with the surface. A wheel assembly is a device which provides or allows motion between its mount and surface on which it is intended to have **a single point of rolling contact.**

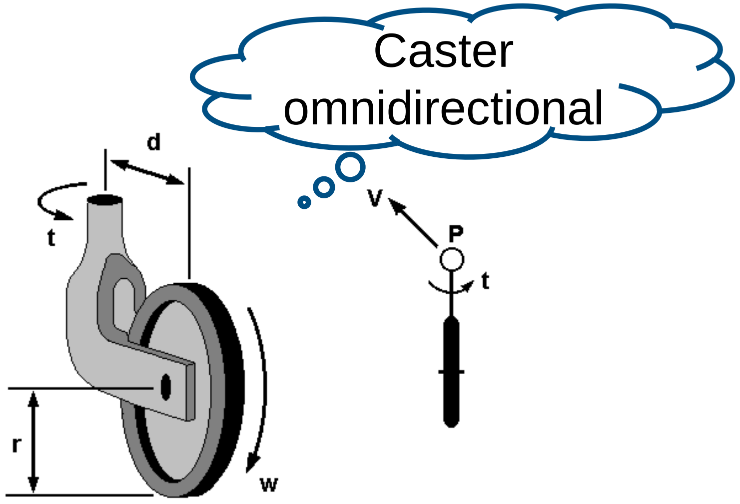
(Muir and Newman, 1986)

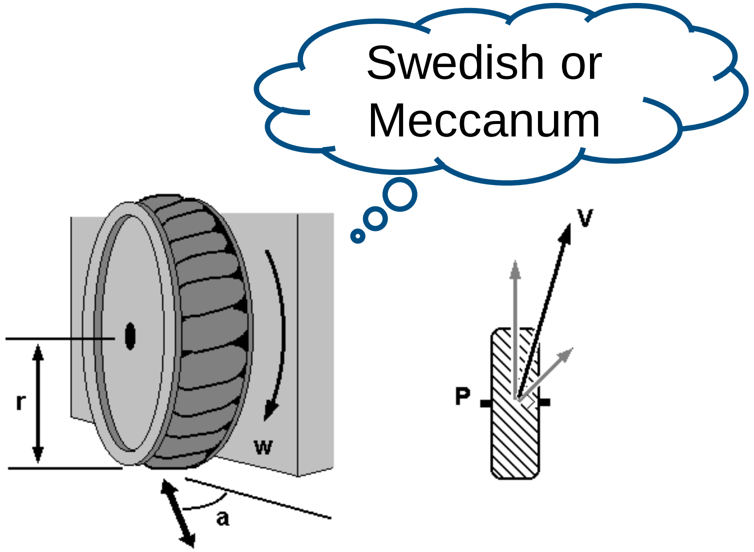
Robot Mobile | AGV | Unmanned vehicle

Wheels Types









Mobile Robots Types/Kinematics (some)

Two wheels (differential drive)

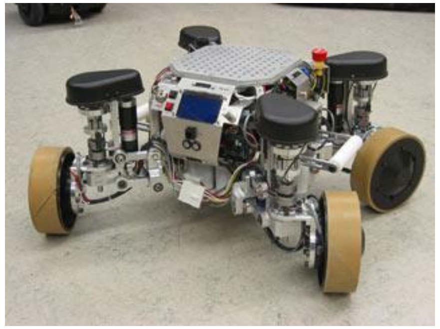
- Simple model
- Suffers terrain irregularities
- Cannot translate laterally

Tracks

- Suited for outdoor terrains
- Not accurate movements (with rotations)
- Complex model
- Cannot translate laterally

Omnidirectional (synchro drive)

- Can exploit all degrees of freedom (3DoF)
- Complex model
- Complex structure



Differential Drive (MRT – Politecnico di Milano)

Differential Drive (MRT – Politecnico di Milano)

Omnidirectional (Swedish wheels)

Omnidirectional (Syncro drive)

Some Definitions ...

<u>Locomotion</u>: the process of causing an autonomous robot to move

- To produce motion, forces must be applied to the vehicle

<u>Dynamics</u>: the study of motion in which forces are modeled

- Includes the energies and speeds associated with these motions

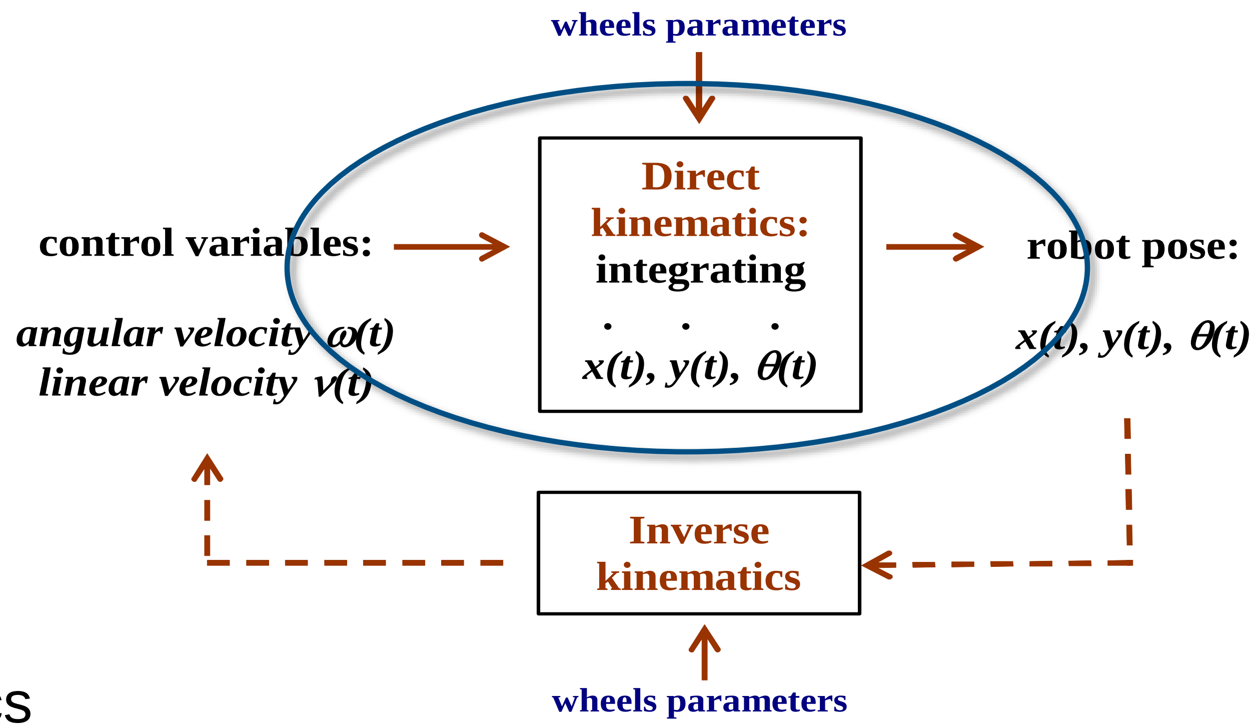
<u>Kinematics</u>: study of motion without considering forces that affect if

- Deals with the geometric relationships that govern the system
- Deals with the relationship between control parameters and the behavior of a system in state space

We'll stick on the plane!

We'll focus on this motion model!

Kinematics



Direct kinematics

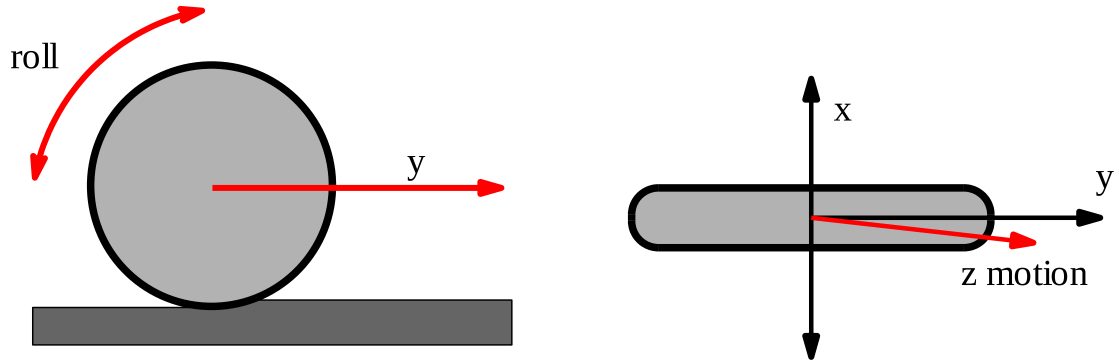
- Given control parameters, e.g., wheels and velocities, and a time of movement *t*, find the pose (x, y, θ) reached by the robot

Inverse kinematics

- Given the final pose *(x, y, θ)* find control parameters to move there in a given time *t*

Wheeled robot assumptions

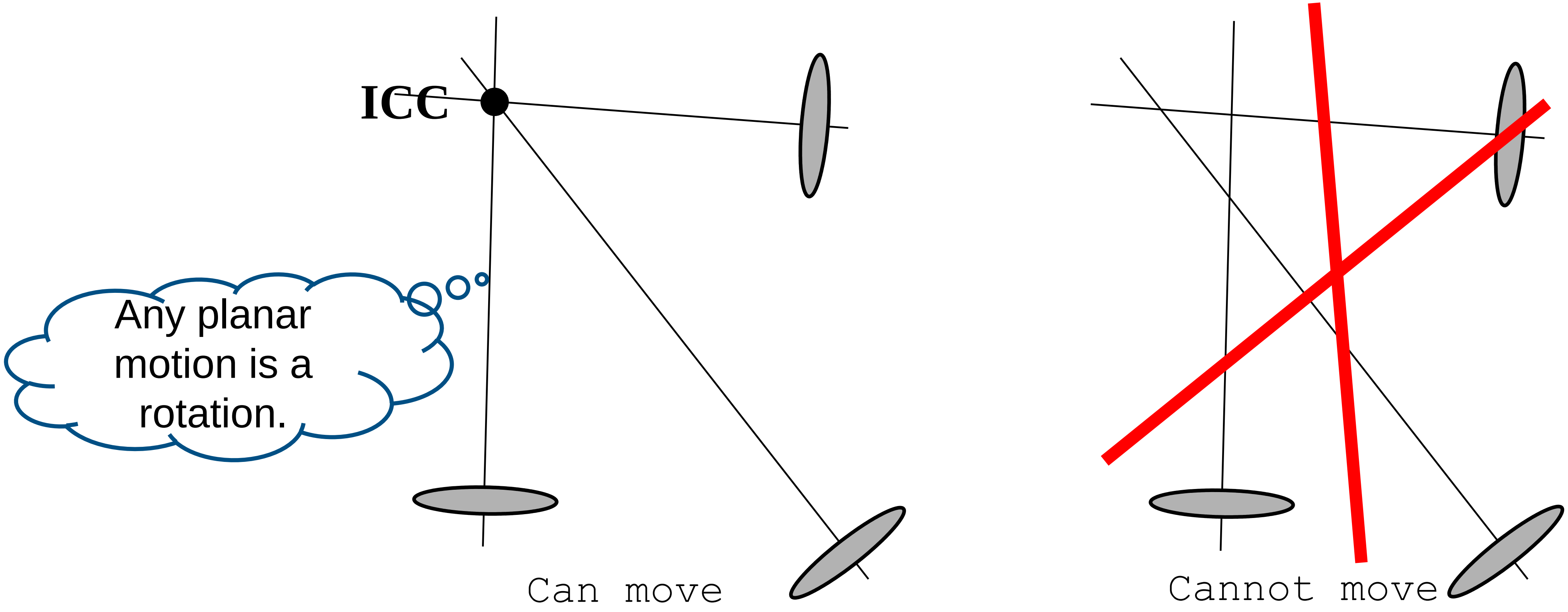
1. Robot made only by rigid parts
2. Each wheel may have a 1 link for steering
3. Steering axes are orthogonal to soil
4. Pure rolling of the wheel about its axis (x axis)
5. No translation of the wheel



Wheel parameters:
r = *radius*
v = *linear velocity*
ω = *angular velocity*

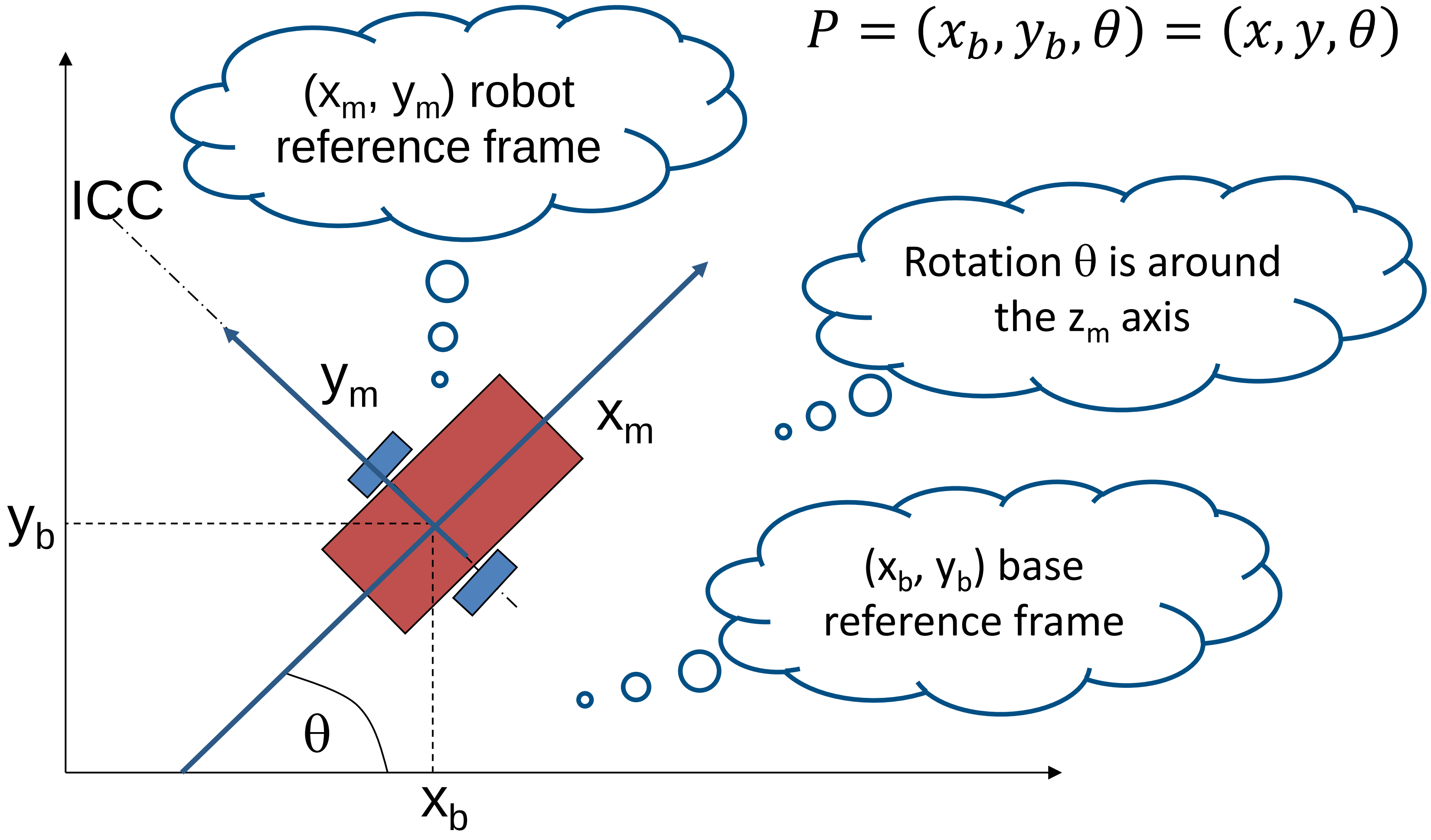
Instantaneous Center of Curvature (or Rotation)

For a robot to move on the plane (3DoF), without slipage, wheels axis have to intersect in a single point named Instantaneous Center of Curvature (ICC) or Instantaneous Center of Rotation (ICR)



Representing a Pose

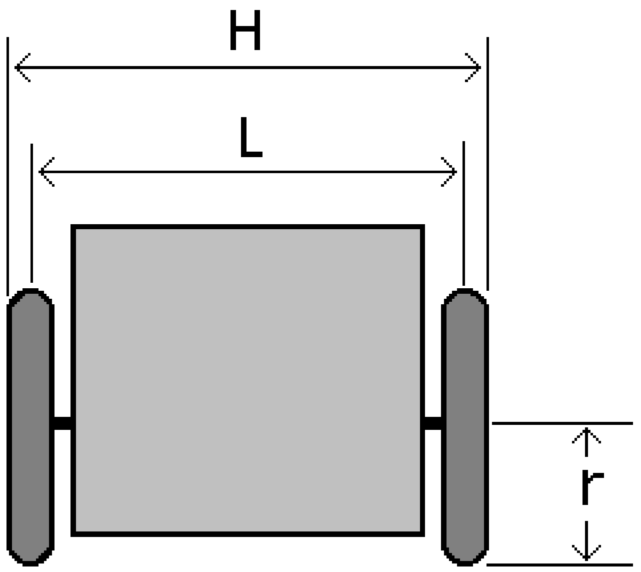
$$P = (x_b, y_b, \theta) = (x, y, \theta)$$



Differential Drive Kinematics (1)

Construction

- 2 wheels on the same axis
- 2 independent motors (one for wheel)
- 3rd passive supporting wheel



Variables independently controlled

- V_R: velocity of the right wheel
- V_L: velocity of the left wheel

Pose representation in base reference:

$$P = (x_b, y_b, \theta) = (x, y, \theta)$$

Control input are:

- v: linear velocity of the robot
- ω: angular velocity of the robot

Linearly related to V_R and V_L ...

Differential Drive Kinematics (2)

Right and left wheels follow a circular path with
ω angular velocity and different curvature radius

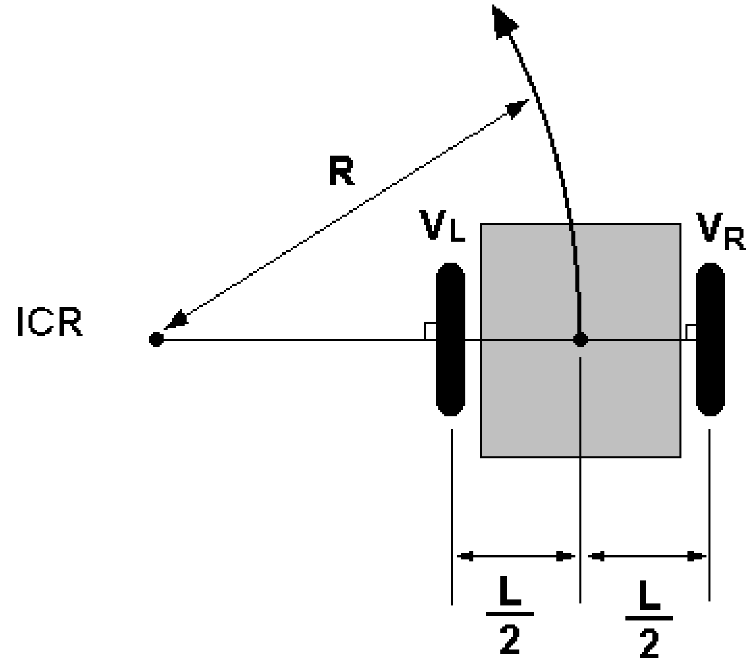
$$\omega (R + L/2) = V_R$$
$$\omega (R - L/2) = V_L$$

Given V_R and V_L you can find ω solving for R and equating

$$\omega = (V_R - V_L)/L$$

Similarly you can find R solving for ω and equating

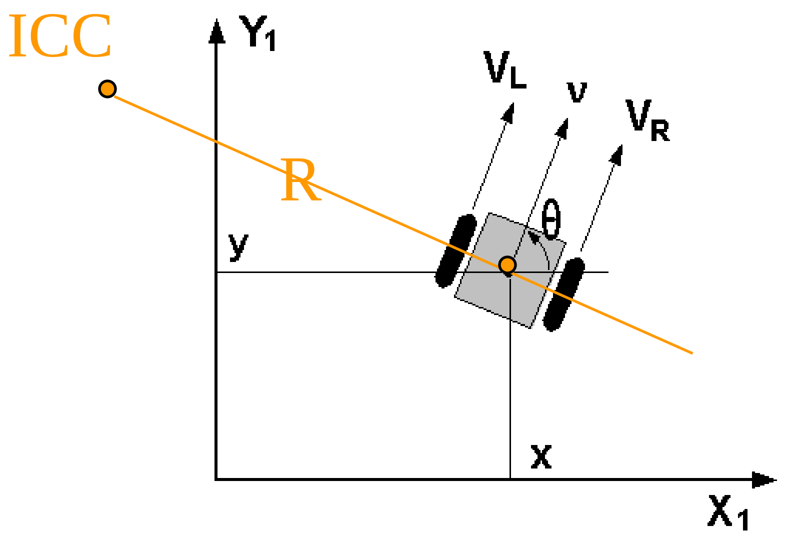
$$R = L/2 \; (V_R + V_L) / (V_R - V_L)$$



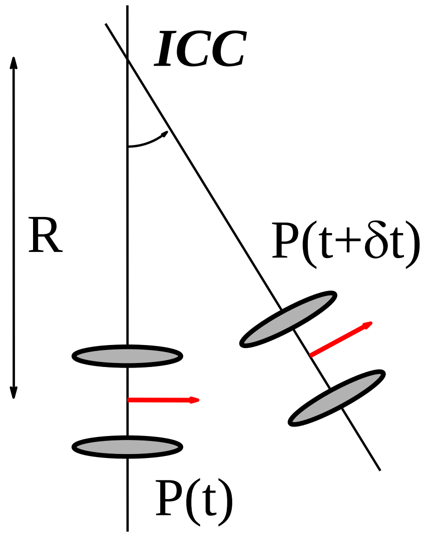
Rotation in place
$R = 0, V_R = -V_L$
Linear movement
$R = \text{infinite}, \; V_R = V_L$

Differential Drive ICC

Wheels move around ICC on a circumference
with istantaneous radius R and angular velocity ω



$$ICC = \left(x + R\cos(\theta + \pi/2),\ y + R\sin(\theta + \pi/2)\right) = \left(x - R\sin(\theta),\ y + R\cos(\theta)\right)$$



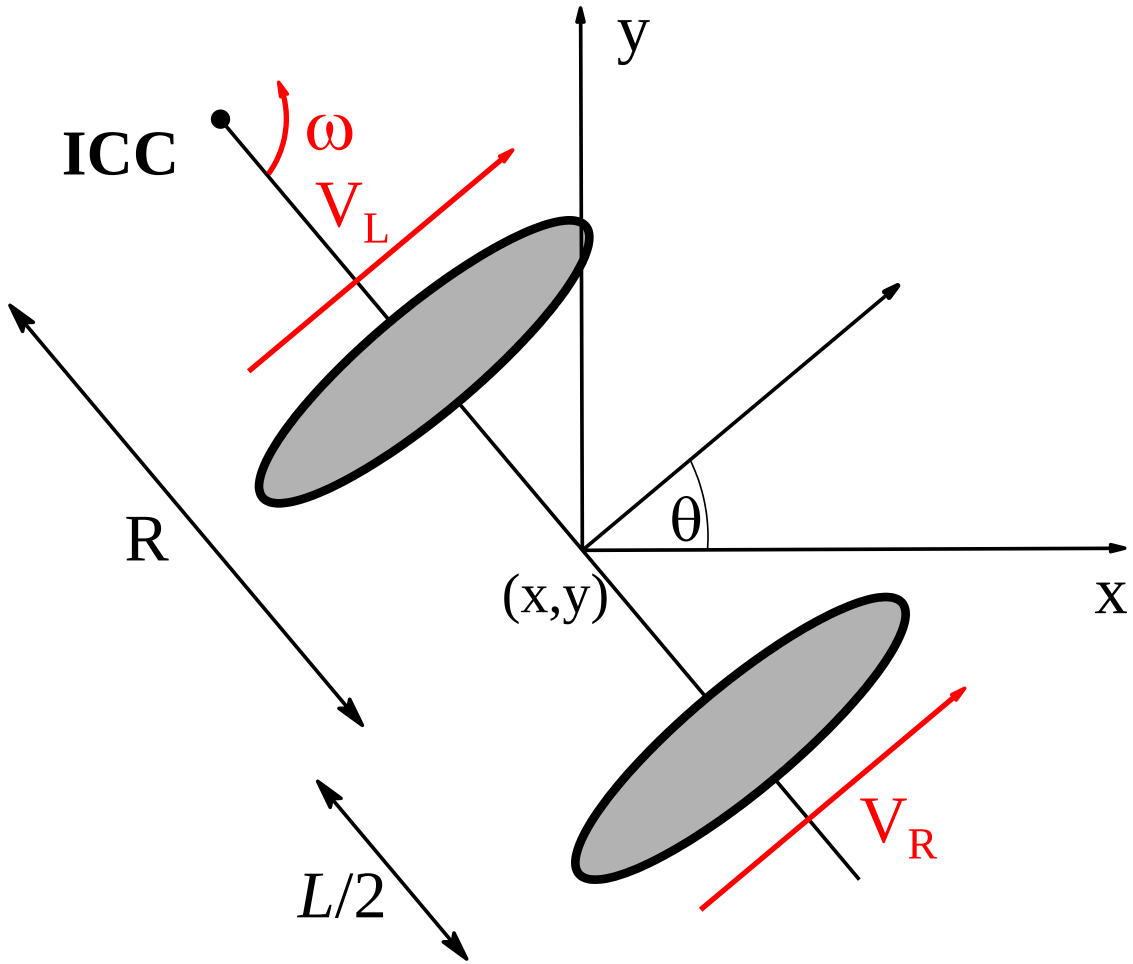
Rotate around ICC

Translate robot in ICC

Translate robot back

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \cdot \delta t) & -\sin(\omega \cdot \delta t) & 0 \\ \sin(\omega \cdot \delta t) & \cos(\omega \cdot \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \cdot \delta t \end{bmatrix}$$

Differential Drive Equations (Remember!)



$$ICC = (x - R\cdot\sin(\theta), y + R\cdot\cos(\theta))$$

$$V_R = \omega\cdot(R + L/2)$$

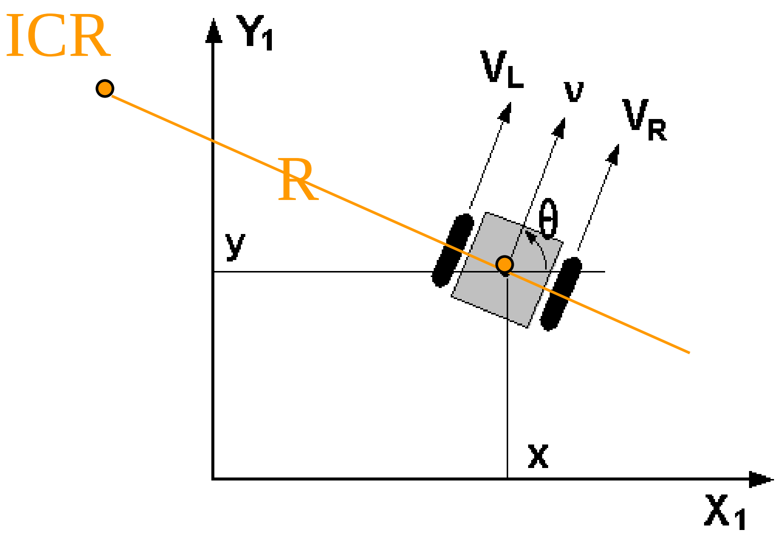
$$V_L = \omega\cdot(R - L/2)$$

$$R = \frac{L}{2}\frac{(V_R + V_L)}{(V_R - V_L)}$$

$$V = \frac{V_R + V_L}{2}$$

$$\omega = \frac{V_R - V_L}{L}$$

Differential drive odometry



Being known

$$\omega = (V_R - V_L)/L$$

$$R = L/2\,(V_R + V_L)/(V_R - V_L)$$

$$V = \omega R = (V_R + V_L)/2$$

Compute the velocity in the base frame

$$V_x = V(t)\cos(\theta(t))$$

$$V_y = V(t)\sin(\theta(t))$$

Integrate position in base frame

$$x(t) = \int V(t)\cos(\theta(t))\,dt$$

$$y(t) = \int V(t)\sin(\theta(t))\,dt$$

$$\theta(t) = \int \omega(t)\,dt$$

$$x(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t'))\cdot\cos(\theta(t'))\cdot dt'$$

$$y(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t'))\cdot\sin(\theta(t'))\cdot dt'$$

$$\theta(t) = \frac{1}{L}\int_0^t (V_R(t') - V_L(t'))\cdot dt'$$

Can integrate
at discrete time

Odometry Integration (1)

Assume constant linear velocity v_k and angular velocity ω_k in $[t_k, t_{k+1}]$ can use Euler integration to compute robot odometry

Approximated

$$x(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t')) \cdot \cos(\theta(t')) \cdot dt'$$

$$y(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t')) \cdot \sin(\theta(t')) \cdot dt'$$

$$\theta(t) = \frac{1}{L}\int_0^t (V_R(t') - V_L(t')) \cdot dt'$$

$$x_{k+1} = x_k + v_k T_S \cos\theta_k$$

$$y_{k+1} = y_k + v_k T_S \sin\theta_k$$

$$\theta_{k+1} = \theta_k + \omega_k T_S$$

$$T_S = t_{k+1} - t_k$$

Exact

matteo.matteucci@polimi.it

Odometry Integration (2)

Assume constant linear velocity v_k and angular velocity ω_k in $[t_k, t_{k+1}]$ can use 2nd order Runge-Kutta integration to compute robot odometry

Better approximation

$$x(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t')) \cdot \cos(\theta(t')) \cdot dt'$$

$$y(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t')) \cdot \sin(\theta(t')) \cdot dt'$$

$$\theta(t) = \frac{1}{L}\int_0^t (V_R(t') - V_L(t')) \cdot dt'$$

$$x_{k+1} = x_k + v_k T_S \cos\left(\theta_k + \frac{\omega_k T_S}{2}\right)$$

$$y_{k+1} = y_k + v_k T_S \sin\left(\theta_k + \frac{\omega_k T_S}{2}\right)$$

$$\theta_{k+1} = \theta_k + \omega_k T_S$$

$$T_S = t_{k+1} - t_k$$

Average orientation

Odometry Integration (3)

Assume constant linear velocity v_k and angular velocity ω_k in $[t_k, t_{k+1}]$ can use exact integration to compute the robot odometry

Exact

$$x(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t')) \cdot \cos(\theta(t')) \cdot dt'$$

$$y(t) = \frac{1}{2}\int_0^t (V_R(t') + V_L(t')) \cdot \sin(\theta(t')) \cdot dt'$$

$$\theta(t) = \frac{1}{L}\int_0^t (V_R(t') - V_L(t')) \cdot dt'$$

$$x_{k+1} = x_k + \frac{v_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k)$$

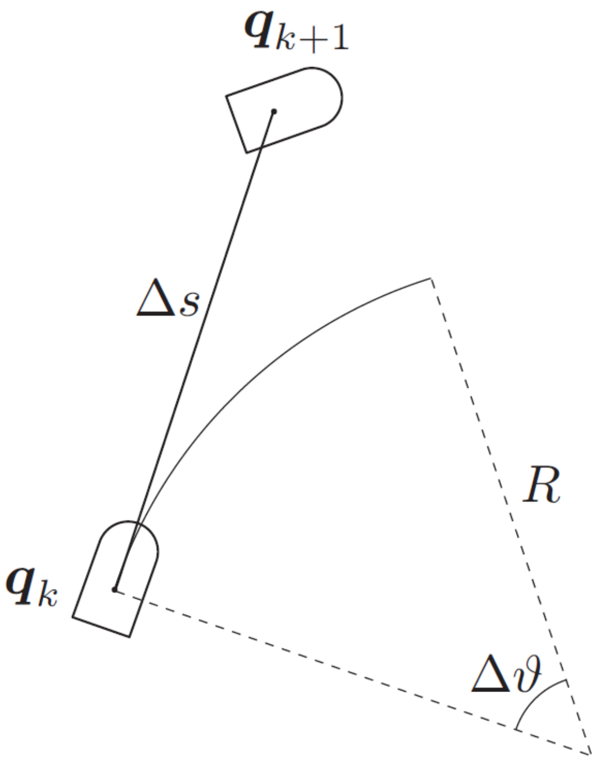
$$y_{k+1} = y_k - \frac{v_k}{\omega_k}(\cos\theta_{k+1} - \cos\theta_k)$$

$$\theta_{k+1} = \theta_k + \omega_k T_S$$

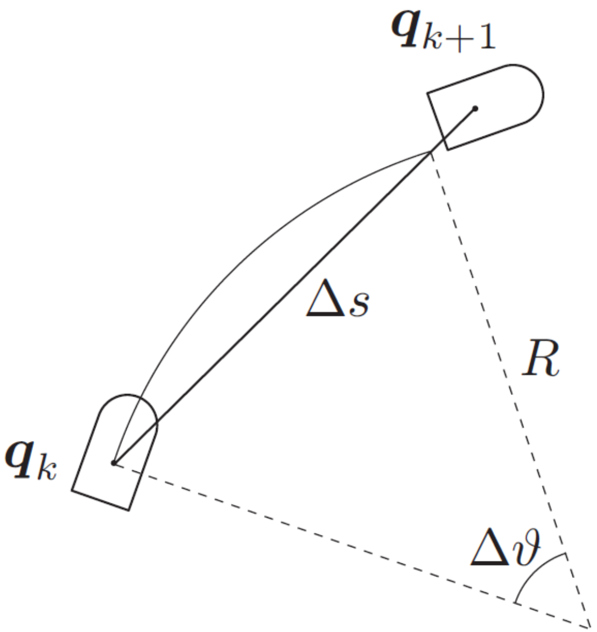
$$T_S = t_{k+1} - t_k$$

Need to use Rugge Kutta for $\omega \sim 0$

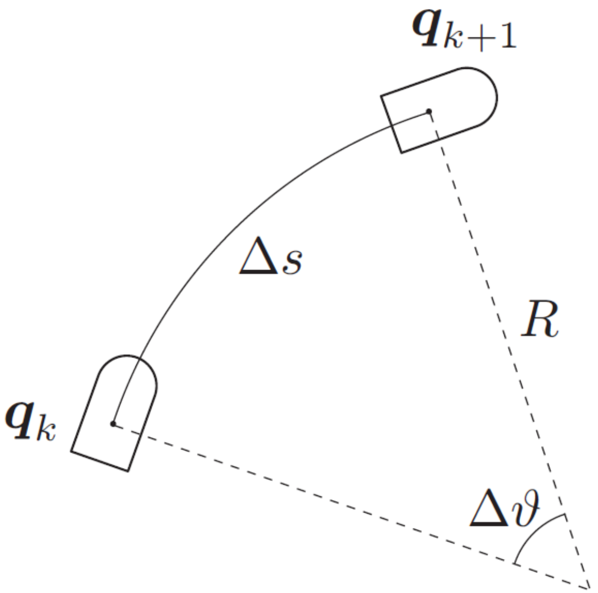
Odometry Comparison



Euler



Runge-Kutta



exact

Tips and Tricks

Proprioceptive measurements are used to compute linear v_k and angular velocity ω_k

$$v_k T_S = \Delta s, \qquad \omega_k T_S = \Delta\theta, \qquad \frac{v_k}{\omega_k} = \frac{\Delta s}{\Delta\theta}$$

with Δs is the traveled distance and $\Delta\theta$ is the orientation change

Nice but drifts because of ...

In a differential drive these become

$$\Delta s = \frac{r}{2}(\Delta\phi_R + \Delta\phi_L), \qquad \Delta\theta = \frac{r}{L}(\Delta\phi_R - \Delta\phi_L)$$

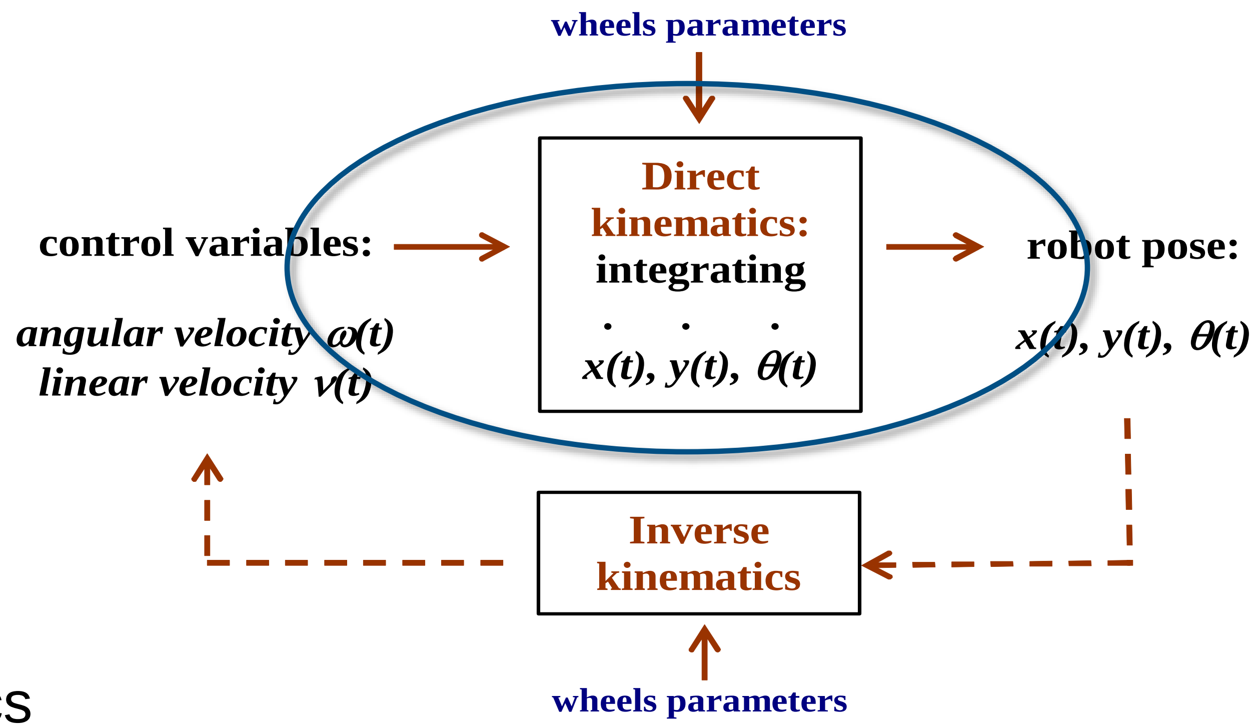
with $\Delta\phi_R$ and $\Delta\phi_L$ the total rotations measured by wheel encoders

... slipage ...

... integration...

... calibration ...

Kinematics



Direct kinematics

- Given control parameters, e.g., wheels and velocities, and a time of movement *t*, find the pose (x, y, θ) reached by the robot

Inverse kinematics

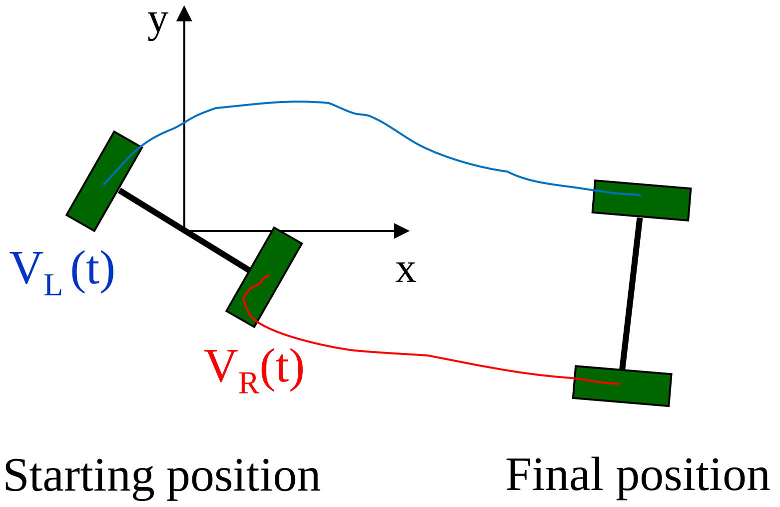
- Given the final pose *(x, y, θ)* find control parameters to move there in a given time *t*

Inverse kinematics

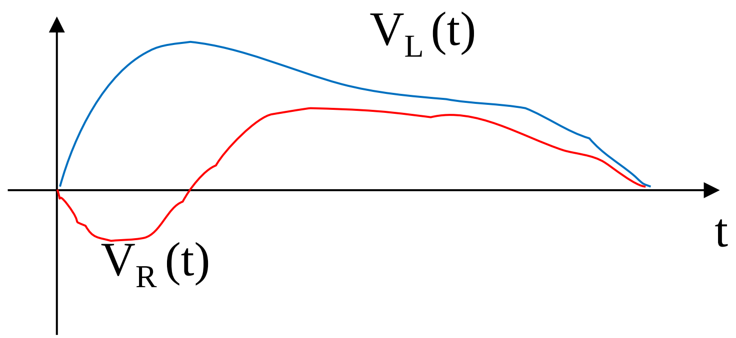
Given a desired position or velocity, what can we do to achieve it?

Finding "some" solution is not hard, but finding the "best" solution can be very difficult:

- Shortest time
- Most energy efficient
- Smoothest velocity profiles



Moreover if we have non holonomic constrains and only two control variables; we cannot directly reach any of the 3DoF final positions ...



Differential drive inverse kinematics

Decompose the problem and control only few DoF at the time

1. Turn so that the wheels are parallel to the line between the original and final position of robot origin

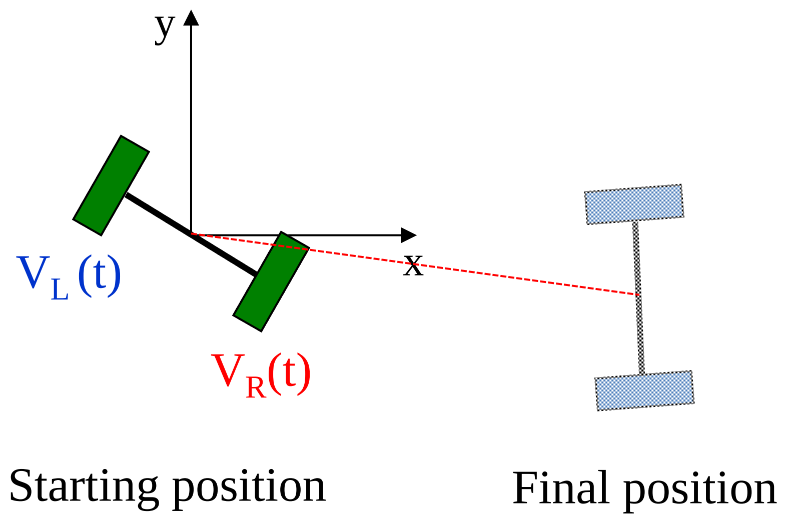
$$-V_L(t) = V_R(t) = V_{max}$$

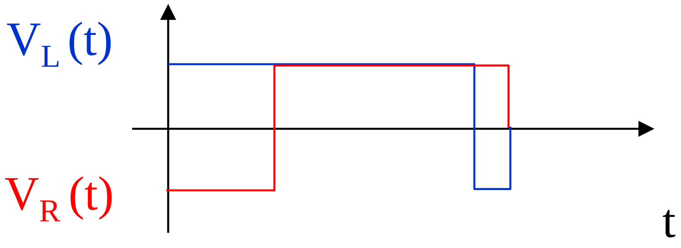
2. Drive straight until the robot's origin coincides with destination

$$V_L(t) = V_R(t) = V_{max}$$

3. Rotate again in to achieve the desired final orientation

$$-V_L(t) = V_R(t) = V_{max}$$

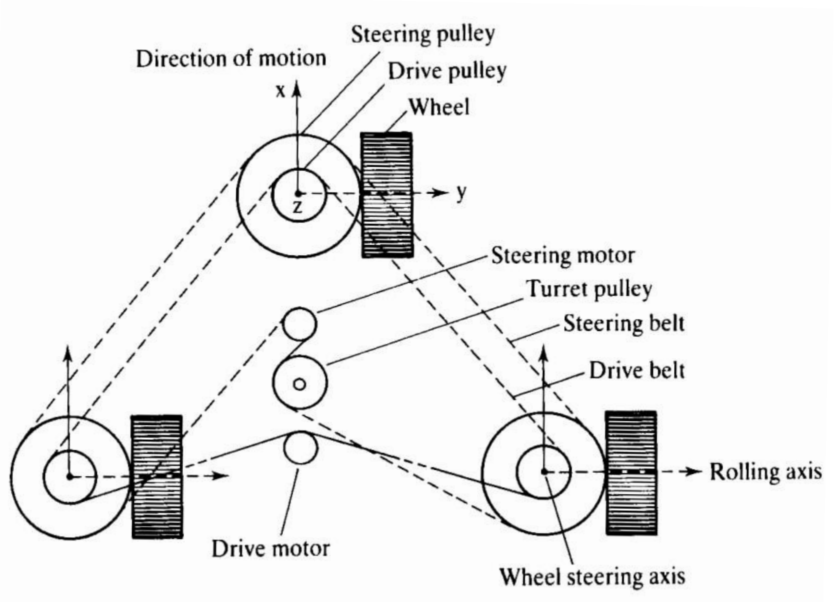


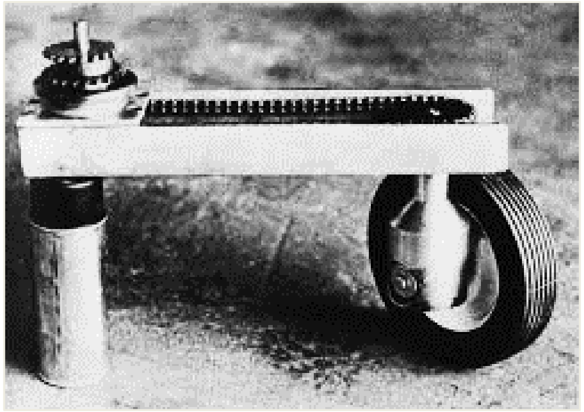


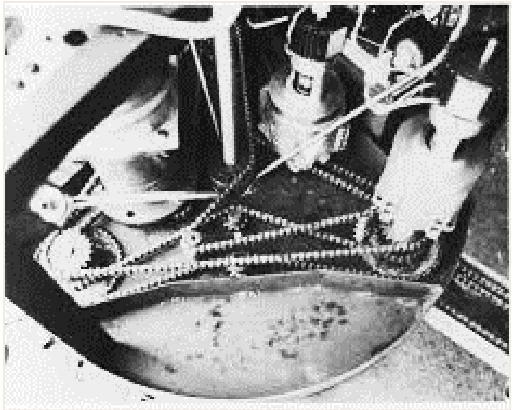
Synchronous drive

Complex mechanical robot design

- (At least) 3 wheels actuated and steered
- A motor to roll all the wheels, a second motor to rotate them
- Wheels point in the same direction
- It is possible to control directly θ with an additional actuator







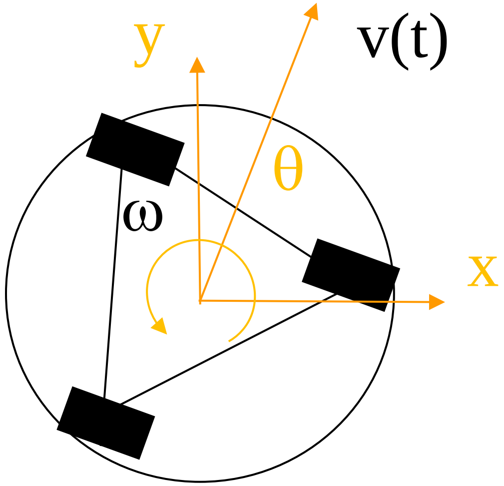
Robot control variables

- Linear velocity v(t)
- Angular velocity ω(t)

Its ICC is always at the infinite and the robot is **<u>non holonomic</u>** (it can only translate freely)

Synchronous drive kinematics

Robot control for the synchronous drive
- Direct control of v(t) and ω(t)
- Steering changes the direction of ICC



Particular cases:
- v(t)=0, ω(t) = ω for dt → robot rotates in place
- v(t)=v, ω(t) = 0 for dt → robot moves linearly

Compute the velocity in the base frame

$$V_x = V(t) \cos(\theta(t))$$

$$V_y = V(t) \sin(\theta(t))$$

Calls odometry also for diff drive!

Integrate position in base frame to get the robot odometry (traversed path) ...

$$x(t) = \int_0^t v(t') \cos[\theta(t')] dt'$$

$$y(t) = \int_0^t v(t') \sin[\theta(t')] dt'$$

$$\theta(t) = \int_0^t \omega(t')\, dt'$$

Synchro drive inverse kinematics

Decompose the problem and control only a few degrees of freedom at a time

1. Turn so that the wheels are parallel to the line between the original and final position of robot origin

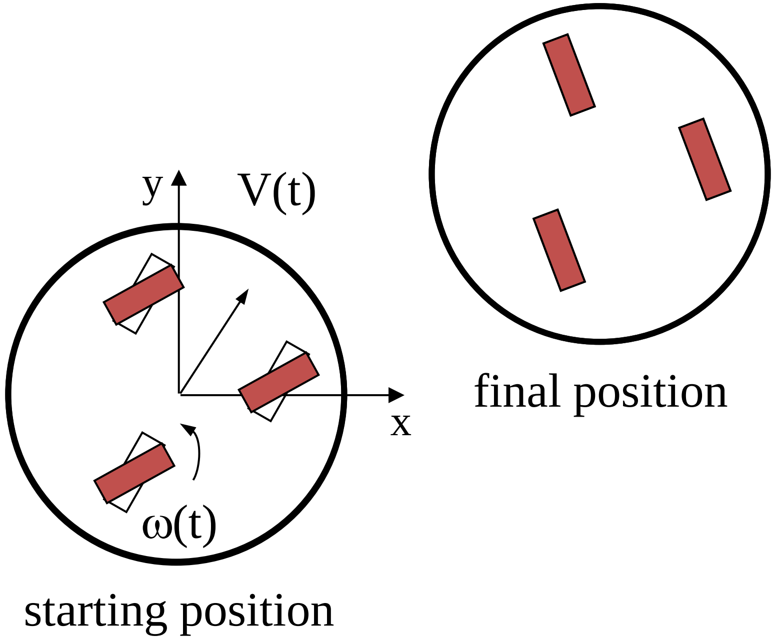
$$\omega(t) = \omega_{max}$$

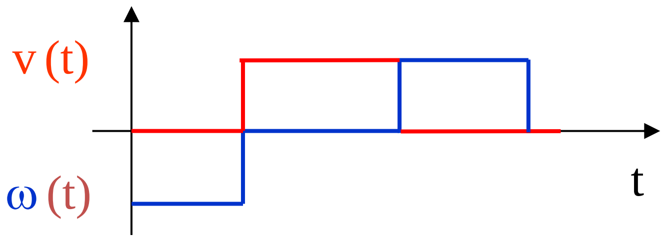
2. Drive straight until the robot's origin coincides with destination

$$v(t) = v_{max}$$

3. Rotate again in to achieve the desired final orientation

$$\omega(t) = \omega_{max}$$





Omnidirectional (Syncro drive)

Omnidirectional Robot

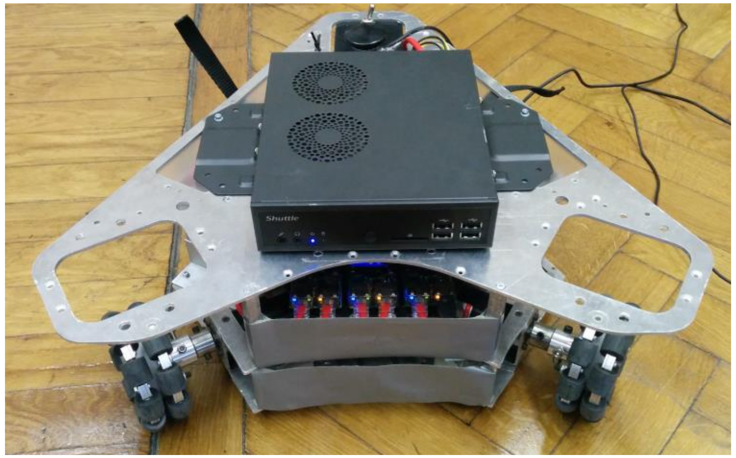
Simple mechanical robot design

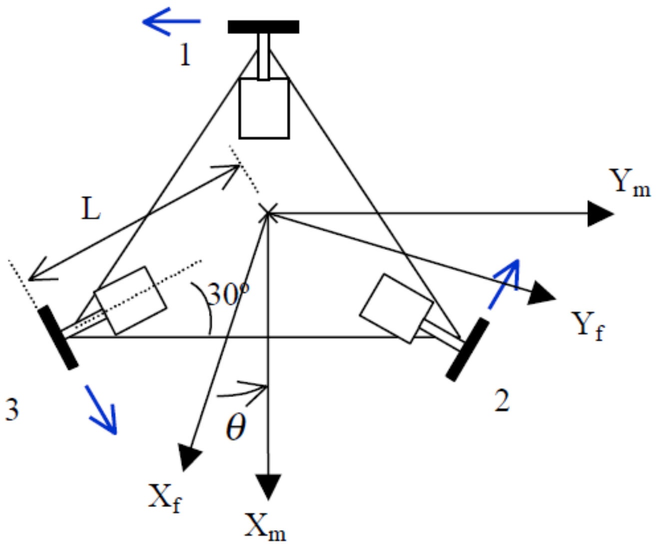
- (At least) 3 Swedish wheels actuated
- One independent motor per wheel
- Wheels point in different direction
- It is possible to control directly x, y, θ

Robot control variables

- Linear velocity v(t) (each component)
- Angular velocity ω(t)

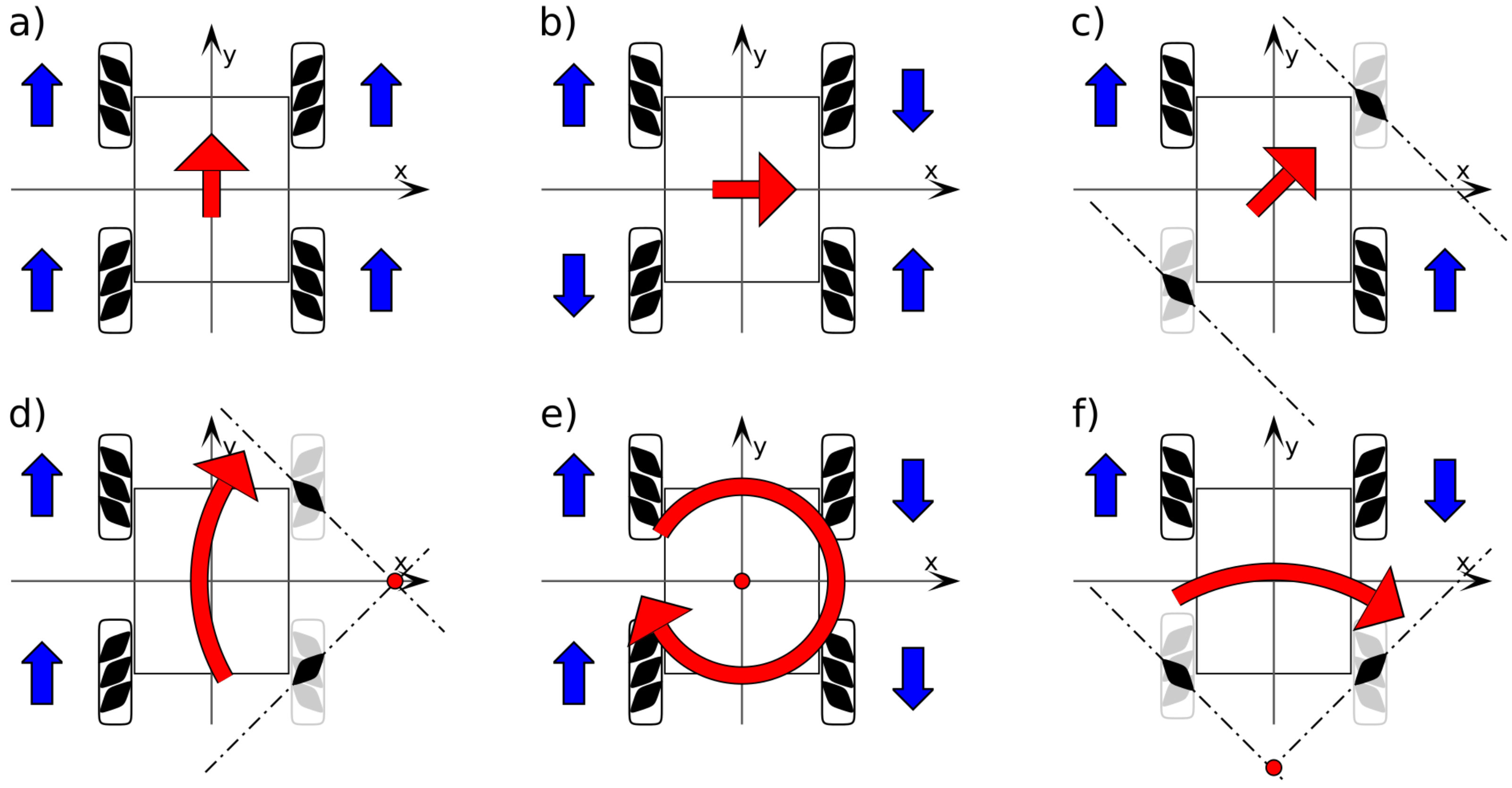
$$\begin{bmatrix} V_x \\ V_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{\sqrt{3}}r & \frac{1}{\sqrt{3}}r \\ -\frac{2}{3}r & \frac{1}{3}r & \frac{1}{3}r \\ \frac{r}{3L} & \frac{r}{3L} & \frac{r}{3L} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$





Omni Wheels (poly wheels)

Mecanum Wheels



Omnidirectional (Swedish wheels)

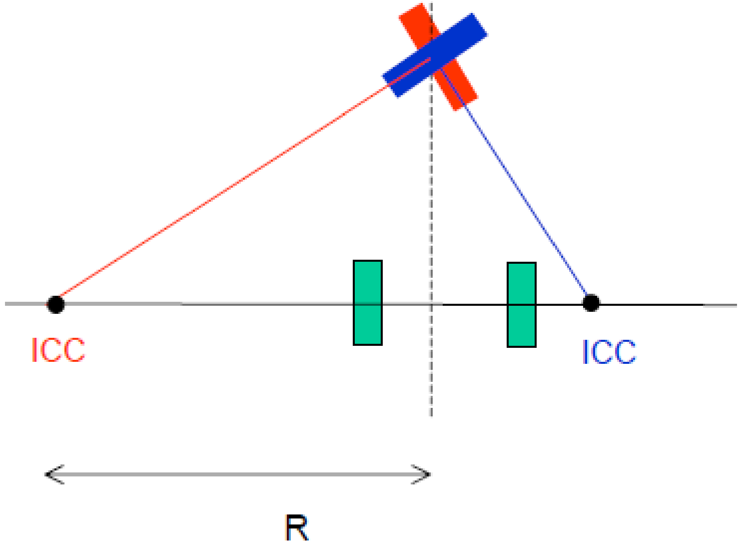
Tricycle kinematics

The Tricycle is the typical kinematics of AGV

- One actuated and steerable wheel
- 2 additional passive wheels
- Cannot control θ independently unless $\alpha(t)$ can be up to 90 degrees
- ICC must lie on the line that passes through the fixed wheels

Robot control variables

- Steering direction $\alpha(t)$
- Angular velocity of steering wheel $\omega_s(t)$



Particular cases:

- $\alpha(t)=0$, $\omega_s(t) = \omega \rightarrow$ moves straight
- $\alpha(t)=90$, $\omega_s(t) = \omega \rightarrow$ rotates in place

Tricycle kinematics

Direct kinematics can be derived as:

$$r = steering\ wheel\ radius$$

$$V_s(t) = \omega_s(t) \cdot r$$

Angular velocity of the moving frame

$$R(t) = d \cdot \tan(\frac{\pi}{2} - \alpha(t))$$

$$\omega(t) = \frac{\omega_s(t) \cdot r}{\sqrt{d^2 + R(t)^2}} = \frac{V_s(t)}{d} \sin \alpha(t)$$

Linear velocity v(t)

In the robot frame

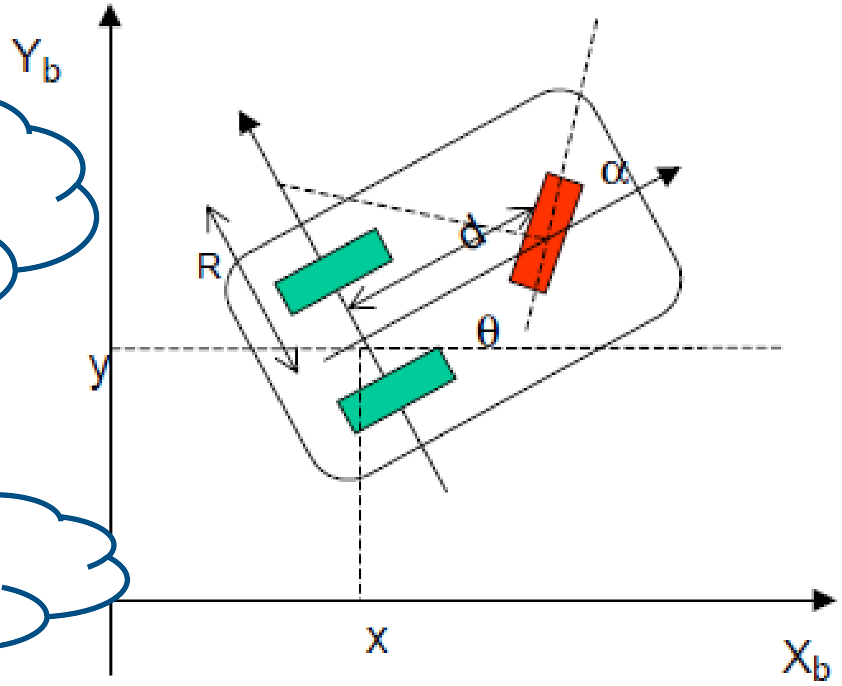
$$V_x(t) = V_s(t) \cdot \cos \alpha(t)$$

$$V_y\,(t) = 0$$

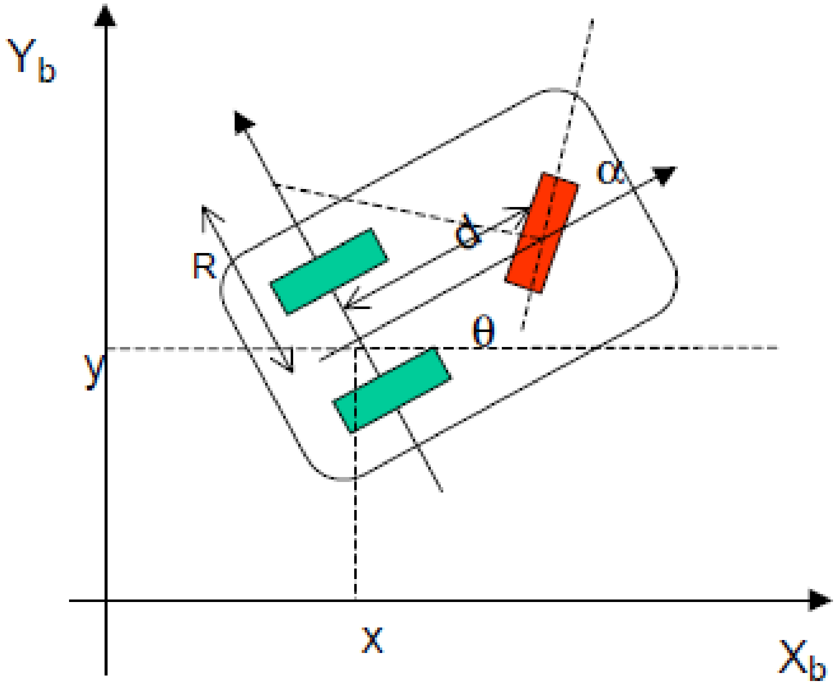
We assume no slipage

Angular velocity ω(t)

$$\dot{\theta} = \frac{V_s(t)}{d} \cdot \sin\ \alpha(t)$$



Tricycle kinematics



Direct kinematics can be derived as:

$$r = steering\ wheel\ radius$$

$$V_s(t) = \omega_s(t) \cdot r$$

$$R(t) = d \cdot \tan(\frac{\pi}{2} - \alpha(t))$$

$$\omega(t) = \frac{\omega_s(t) \cdot r}{\sqrt{d^2 + R(t)^2}} = \frac{V_s(t)}{d} \sin \alpha(t)$$

In the base frame

$$\dot{x}\,(t) = V_s(t) \cdot \cos \alpha(t) \cdot \cos \theta(t) = V(t) \cdot \cos \theta(t)$$

$$\dot{y}\,(t) = V_s(t) \cdot \cos \alpha(t) \cdot \sin \theta(t) = V(t) \cdot sin\, \theta(t)$$

$$\dot{\theta} = \frac{V_s(t)}{d} \cdot \sin\ \alpha(t) = \omega(t)$$

Ackerman steering

Most diffused kinematics on the planet

- Four wheels turning
- Wheels have limited turning angles
- No in-place rotation

Similar to the Trycicle model

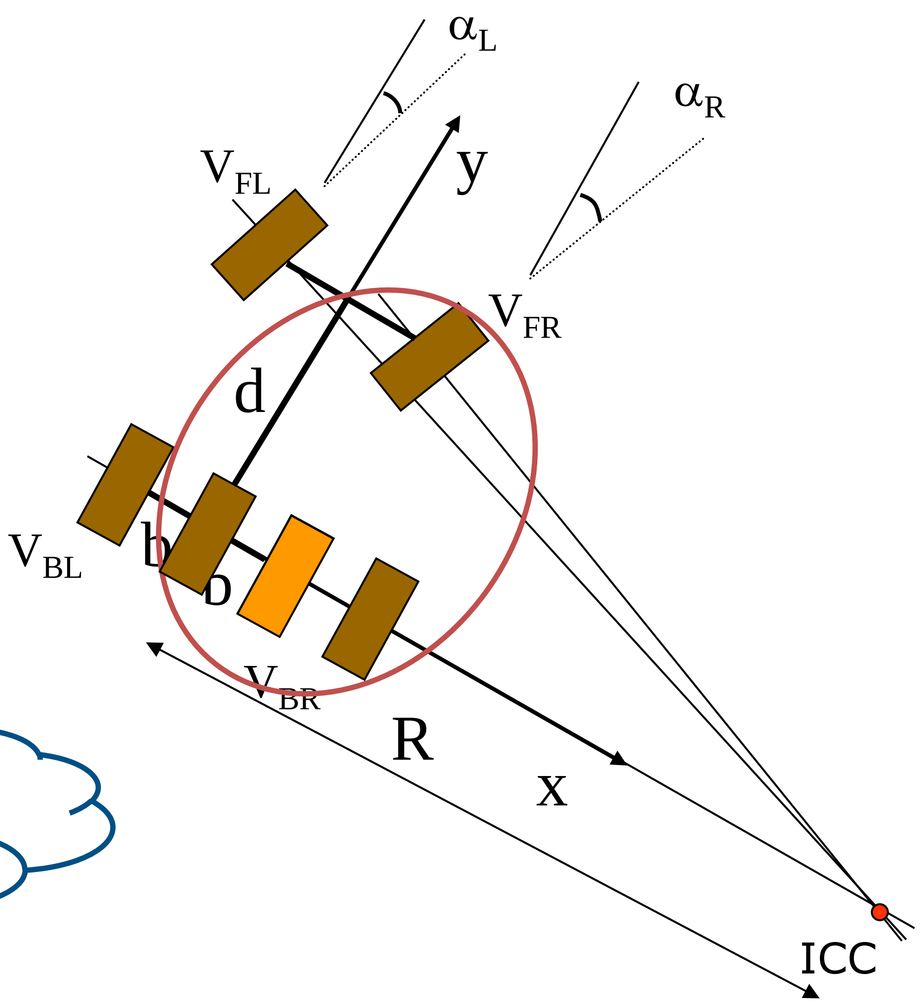
$$R = \frac{d}{\tan \alpha_R} + b$$

$$\frac{\omega d}{\sin \alpha_R} = V_{FR}$$

Determines angular velocity ω(t)

Derive the rest as:

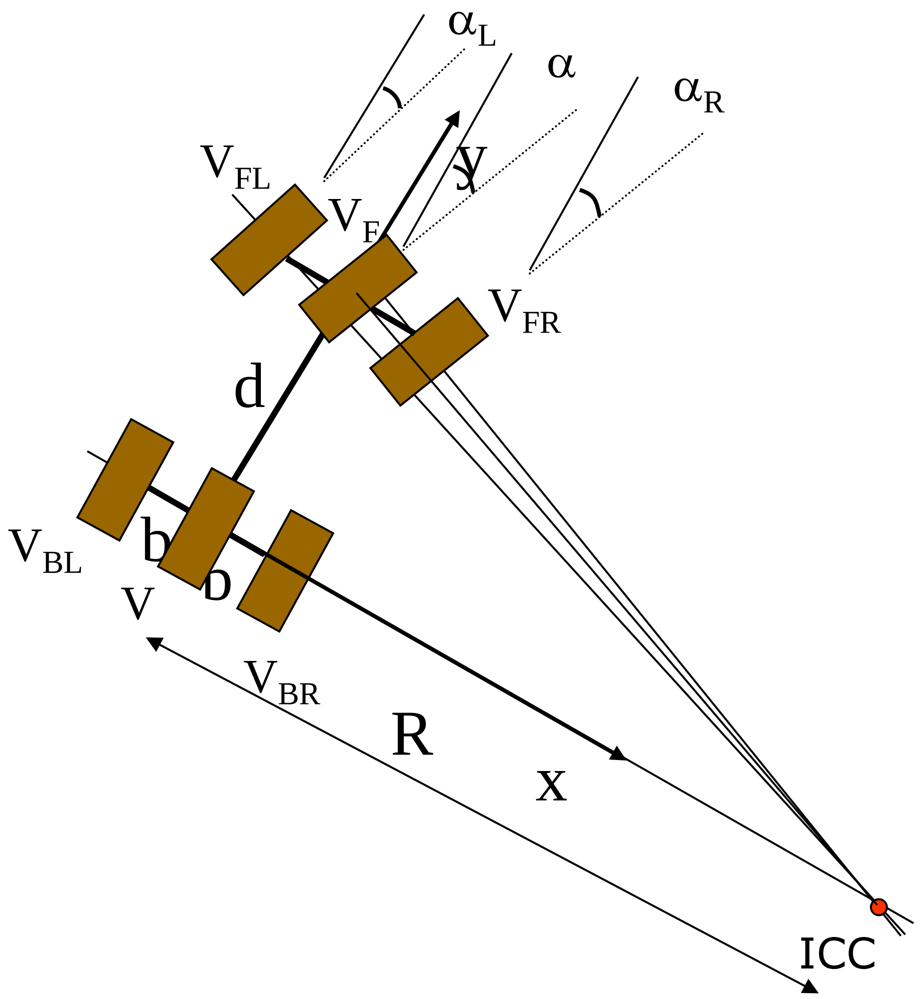
$$\frac{\omega d}{\sin \alpha_L} = V_{FL} \qquad \alpha_L = \tan^{-1}\left(\frac{d}{R+b}\right) \qquad \omega(R+b) = V_{BL} \qquad \omega(R-b) = V_{BR}$$



Ackerman steering (bicycle approximation)

Most diffused kinematics on the planet

- Four wheels turning
- Wheels have limited turning angles
- No in-place rotation



Bicycle approximation

$$R = \frac{d}{\tan\alpha}$$

$$\frac{\omega d}{\sin\alpha} = V_F$$

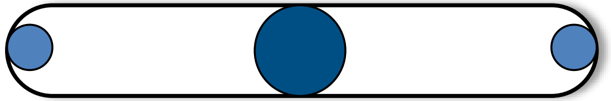
Referred to the center of real wheels

$$\omega R = V \quad \Rightarrow \quad \omega = V \cdot \frac{\tan\alpha}{d}$$

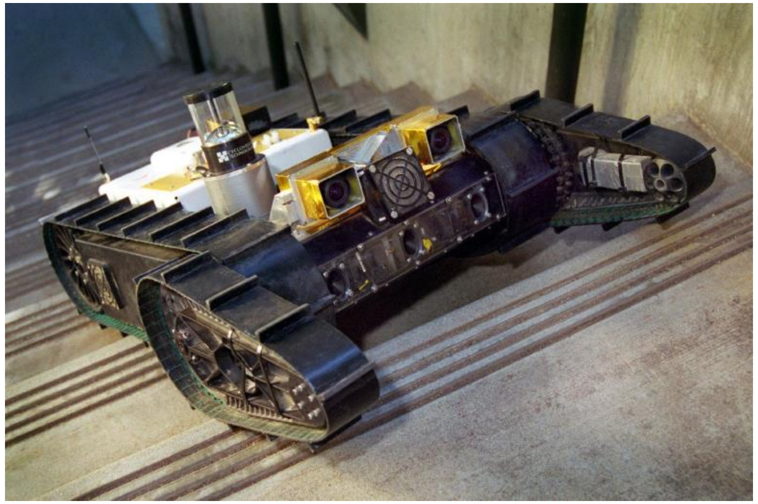
Vehicles with tracks

Vehicles with track have a kinematics similar to the differential drive

- Speed control of each track
- Use the height of the track as wheel diameter



- Often named Skid Steering



Need proper calibration and slippage modeling …

Skid Steering (approximate) Kinematics

Let' assume:

- Mass in the center of the vehicle
- All wheels in contact with the ground
- Wheels on the same side have same speed

$$\omega_l = \omega_1 = \omega_2 \qquad \omega_r = \omega_3 = \omega_4$$

While moving we have multiple ICR and all share ω_z

$$y_G = \frac{v_x}{\omega_z}$$

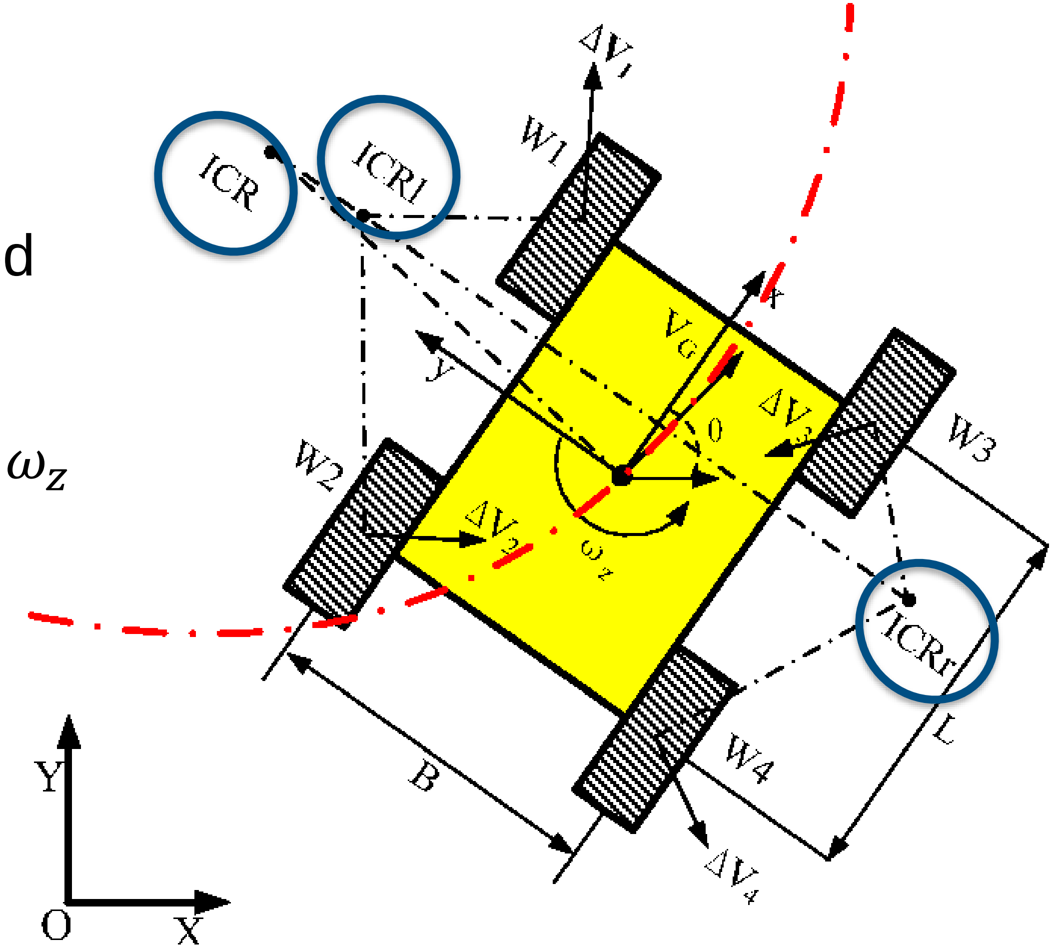
$$y_l = \frac{v_x - \omega_l r}{\omega_z}$$

$$y_r = \frac{v_x - \omega_r r}{\omega_z}$$

$$x_G = x_l = x_r = -\frac{v_y}{\omega_z}$$

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = J_\omega \begin{bmatrix} \omega_l r \\ \omega_r r \end{bmatrix}$$

$$J_\omega = \frac{1}{y_l - y_r} \begin{bmatrix} -y_r & y_l \\ x_G & -x_G \\ -1 & 1 \end{bmatrix}$$



Skid Steering (approximate) Kinematics

Let' assume:

- Mass in the center of the vehicle
- All wheels in contact with the ground
- Wheels on the same side have same speed

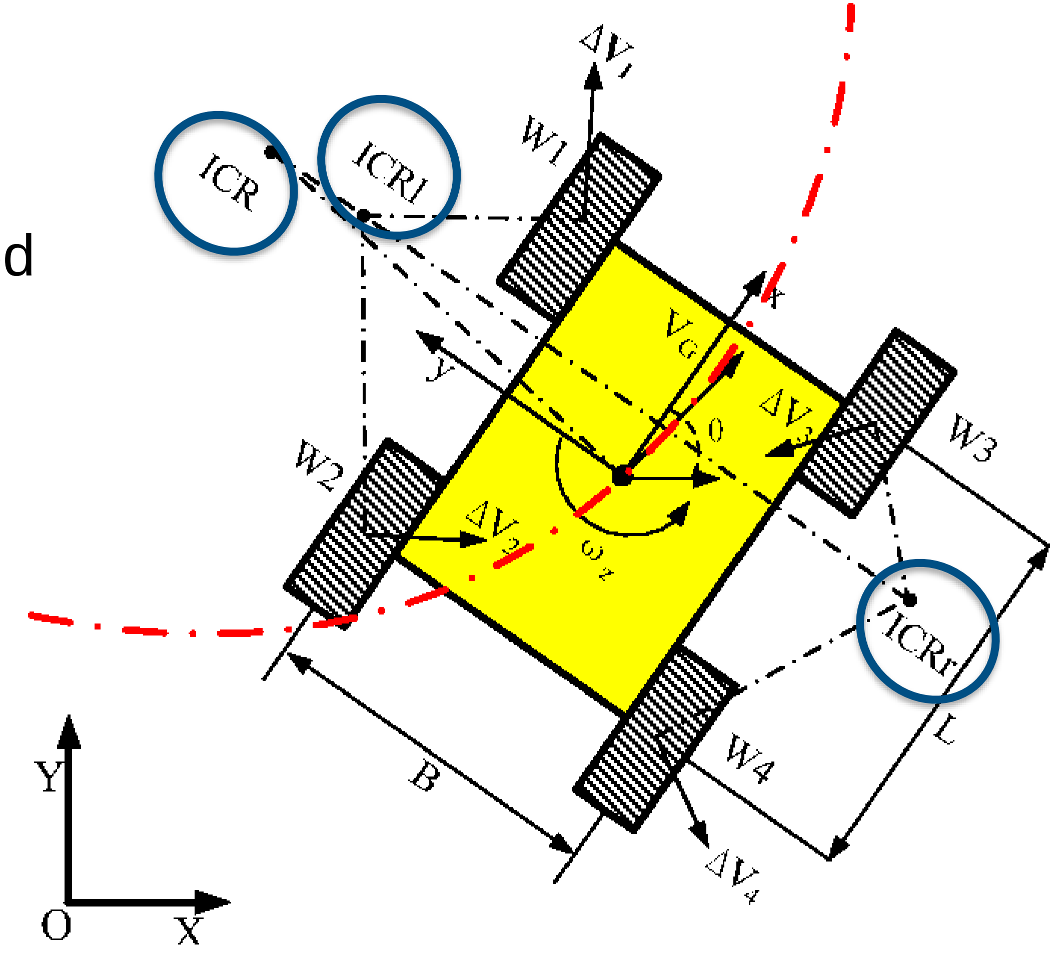
$$\omega_l = \omega_1 = \omega_2 \qquad \omega_r = \omega_3 = \omega_4$$

Assume the robot is symmetric

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = J_\omega \begin{bmatrix} \omega_l r \\ \omega_r r \end{bmatrix}$$

$$J_\omega = \frac{1}{2y_0} \begin{bmatrix} y_0 & y_0 \\ x_G & -x_G \\ -1 & 1 \end{bmatrix} \Rightarrow \begin{cases} v_x = \dfrac{v_l + v_r}{2} \\ v_y = 0 \\ \omega_z = \dfrac{-v_l + v_r}{2y_0} \end{cases}$$

$$y_0 = y_l = -y_r$$



Skid Steering (approximate) Kinematics

Let' assume:

- Mass in the center of the vehicle
- All wheels in contact with the ground
- Wheels on the same side have same speed

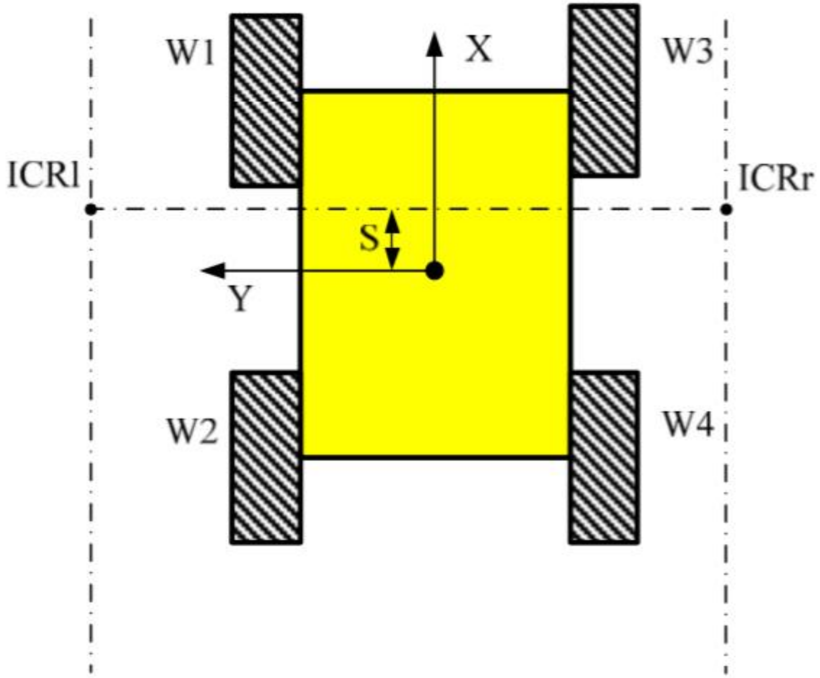
$$\omega_l = \omega_1 = \omega_2 \qquad \omega_r = \omega_3 = \omega_4$$

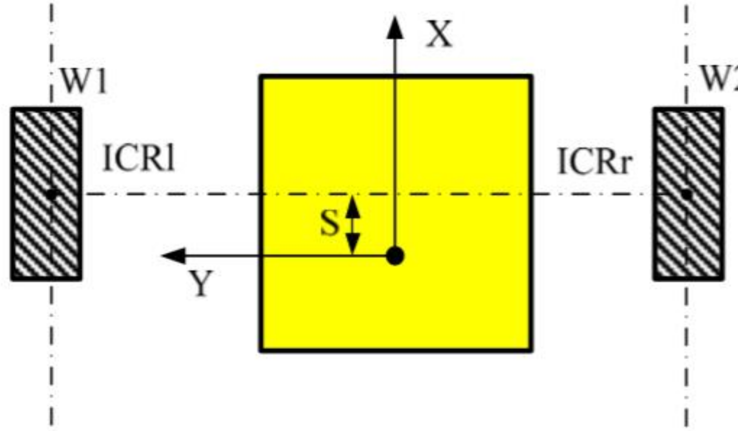
We can get the istantaneous radius of curvature

$$\begin{cases} v_x = \dfrac{v_l + v_r}{2} \\ v_y = 0 \\ \omega_z = \dfrac{-v_l + v_r}{2y_0} \end{cases} \Rightarrow \quad R = \frac{v_G}{\omega_z} = \frac{v_l + v_r}{-v_l + v_r} y_0$$

$$\lambda = \frac{v_l + v_r}{-v_l + v_r}$$

$$\chi = \frac{y_l - y_r}{B} = \frac{2y_0}{B}, \qquad \chi \geq 1$$

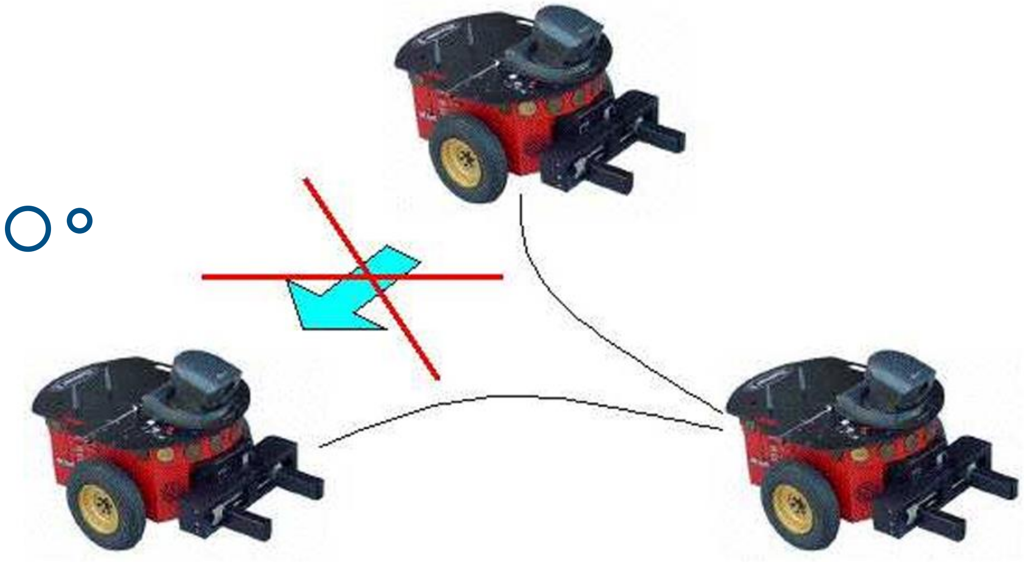




Motion constrains

Due to the presence of wheels (the standard ones) robots cannot move sideways, this is due to the "rolling without slipping" constraint

Pictures and examples from «Control of Nonholonomic Systems» by Giuseppe Oriolo



This is a special case of nonholonomic behavior:

- A nonholonomic mechanical system cannot move in arbitrary directions in its configuration space

Kinematic Constraints (1/2)

The motion of the vehicle is represented by a set of **generalized coordinates**

$$q = (q_1 \; q_2 \; \dots \; q_n)^T$$

Configurations Space Q is an n-dimensional smooth manifold, locally diffeomorfic to $\mathbb{R}^n$

The generalized velocity at a generic point of a trajectory $q(t) \subset Q$ is the tangent vector

$$\dot{q} = (\dot{q}_1 \; \dot{q}_2 \dots \dot{q}_n)^T$$

<u>**Geometric constraints**</u> may exist (or be imposed) on the mechanical system restricting the possible motions to a $(n-k)$ –dimensional submanifold

$$h_i(q) = 0 \qquad i = 1, \dots, k$$

Kinematic Constraints (2/2)

A mechanical system can be subject also to **Kinematic constraints** involving generalized coordinates and their derivatives

$$a_i(q, \dot{q}) = 0 \qquad i = 1, \dots, k$$

These constraints are often **Pfaffian**, i.e., they are linear in the velocities

$$a_i^T(q)\dot{q} = 0 \quad i = 1, \dots, k \qquad \text{or} \qquad A^T(q)\dot{q} = 0$$

Kinematic constraints can be **integrable**, i.e., there might exist k functions h_i such that

$$\frac{\partial h_i(q)}{\partial q} = a_i^T(q) \quad i = 1, \dots, k$$

In this case, the kinematic constraints are geometric constrains.

matteo.matteucci@polimi.it

Holonomic vs Nonoholonomic Constraints

A set of Pfaffian constraints is said holonomic if it is integrable (a geometric limitation); otherwise, it is called nonholonomic (a kinematic limitation).

Consider a single Pfaffian constrain: $a(q, \dot{q}) = 0$

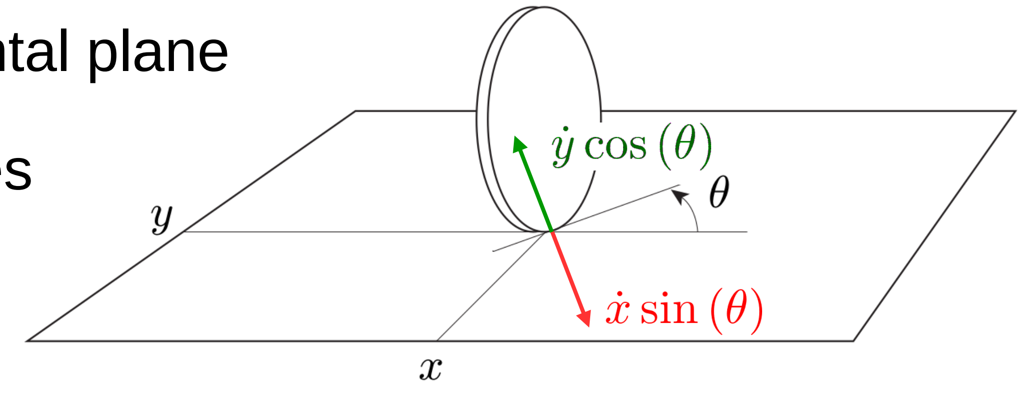
- If the constraint is **holonomic** then it can be integrated as

$$h(q) = c$$

  with $\frac{\partial h_i(q)}{\partial q} = a^T(q)$ and c an integration constraint. The system is confined to lie on a particular level surface (leaf) of h, depending on initial condition $\mathrm{c} = h(q_0)$
- If the constraint is **nonholonomic**, it cannot be integrated.

Although the instantaneous motion (velocity) of the system is restricted to a $n - 1$-dimensional space (the Null Space of the constraint matrix $A^T(q)$), **it is still possible to reach any configuration in $\boldsymbol{Q}$!**

Example: Rolling Disk (1/2)



Consider a disc that rolls without sliding on a horizontal plane

- Disc configuration described three coordinates

$$q = (x\ y\ \theta)^T$$

- Pure rolling constraint can be written as

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$$

- A non holonomic constraint as it does not reduce the generalized coordinates

$$a^T(q) = (\sin\theta \quad -\cos\theta \quad 0) \quad \rightarrow \quad N\big(a^T(q)\big) = span\left\{\begin{pmatrix}\cos\theta\\ \sin\theta\\ 0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}\right\}$$

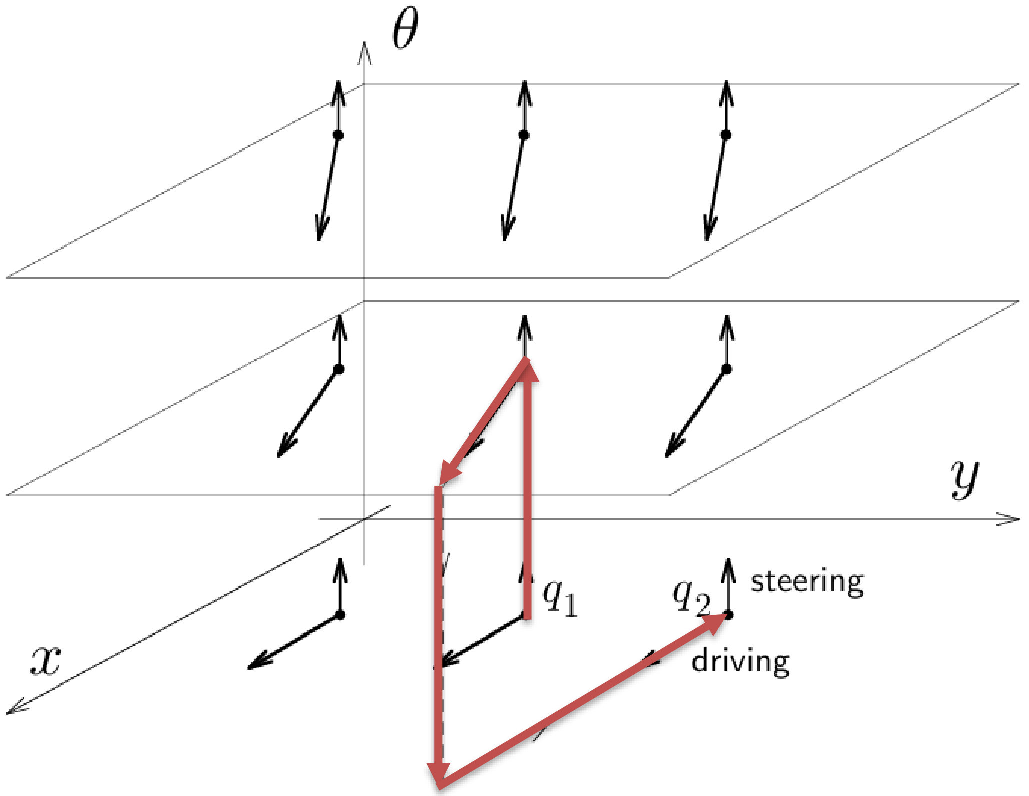
- The line passing through the wheel contact point and having the direction orthogonal to the sagittal axis of the vehicle is called <u>zero motion line</u>

Example: Rolling Disk (1/2)

At each q only two direction of motion are possible according to the null space of the motion constrain matrix

Any configuration $q_f = \left(x_f\ y_f\ \theta_f\right)^T$ can be reached

- Rotate the disk until it aims at $\left(x_f\ y_f\right)$
- Roll the disk until it reaches $\left(x_f\ y_f\right)$
- Rotate the disk until its orientation is θ_f



Holonomicity and Controllability

Holonomicity and non holonomicity are related to controllability of the kinematic model

- A kinematic model is controllable if, given two arbitrarily selected configurations $\boldsymbol{q}_i$ and $\boldsymbol{q}_f$, an input vector $\boldsymbol{u}$ exists generating a trajectory that takes the system from $\boldsymbol{q}_i$ to $\boldsymbol{q}_f$ without violating the kinematic constraints

- If the model is controllable, all the kinematic constraints are nonholonomic, while If the model is not completely controllable, some of the kinematic constraints are holonomic

Mobile robots beyond the wheels

To move on the ground
- Multiple wheels
- Whegs
- Legs

To move in water
- Torpedo-like (single propeller)
- Bodies with thrusters
- Bioinspired

To move in air
- Fixed wings vehicles
- Mobile wings vehicles
- Multi-rotors