Locomotion

Locomotion

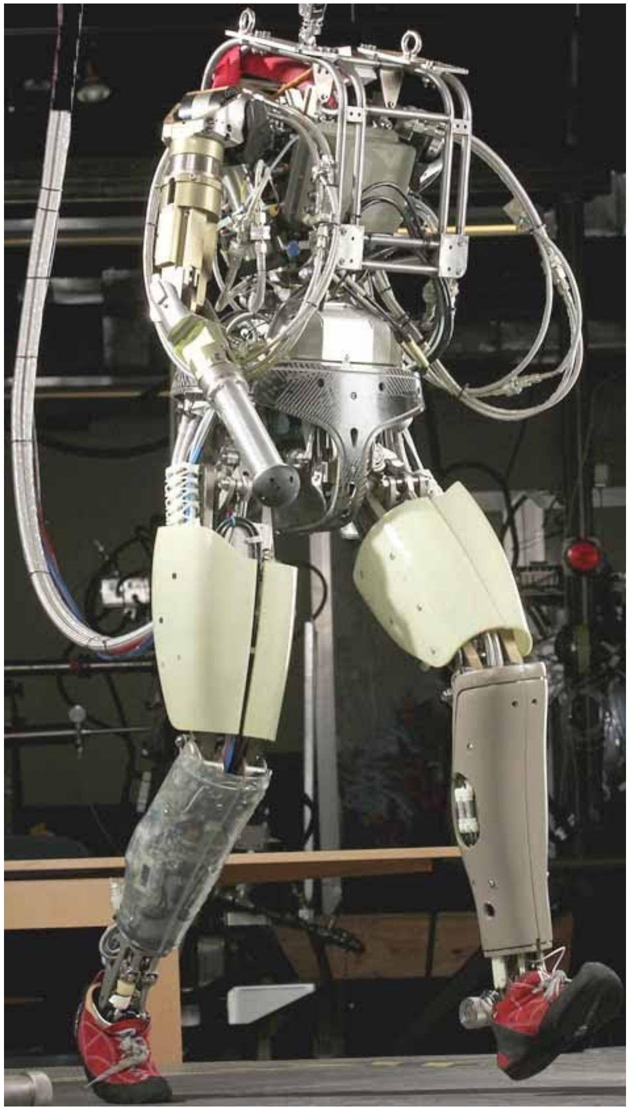
Common types (on land):

- Wheeled
  - differential drive
  - Ackermann
  - synchro
- Tracked
- Walking
- Jumping

Locomotion

general issues

- energy consumption
- control complexity
- mechanical complexity
  - degrees of freedom (DOF)
  - precision
  - stability

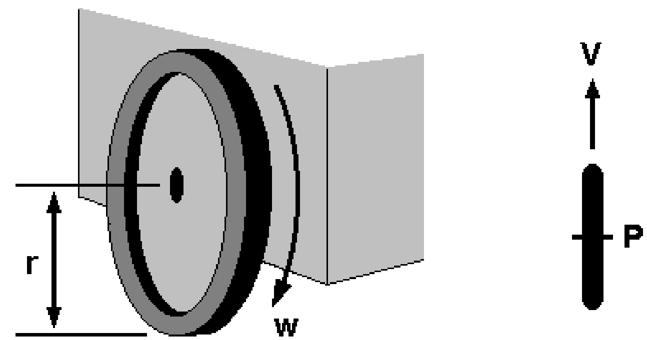


Wheeled Locomotion

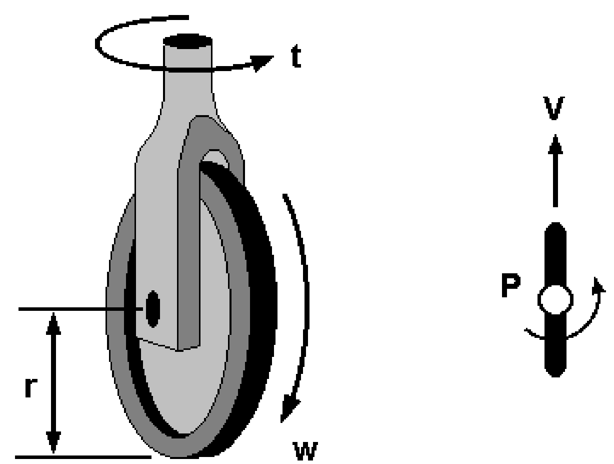
Wheel

- defined by
  - location of its axis
  - and its radius
- motion
  - perpendicular to axis = roll
  - all other = slip
- typically assumed
  - single point of contact (Zero width of the wheel)
  - no slip condition

fixed wheel

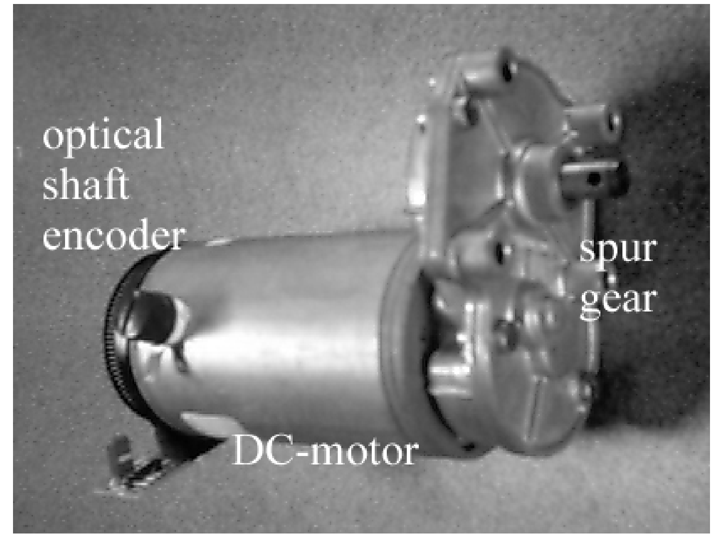


orientable wheel



Wheel Speed

- motor speed N_m
  - typ. measured in RPM (revolutions per minute)
  - by shaft encoder (Qdec, hence signed)
- gear box
  - with gear ratio G
  - typically as fraction 1:X
- wheel axis speed N_a
  - typ. also measured in RPM
  - $N_a = G \cdot N_m$
- wheel speed v
  - typically in m/sec
  - via circumference, i.e., radius r (in m)
  - $v = (N_a / 60\text{sec/min}) \cdot 2\pi \cdot r$

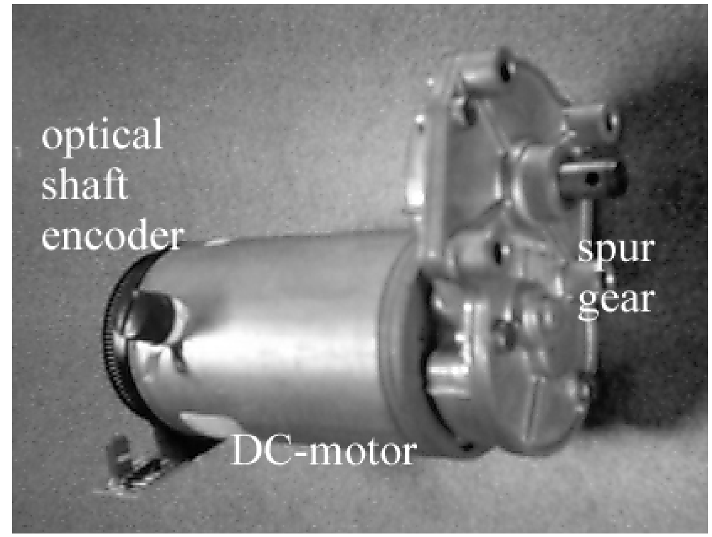


Wheel Speed

note:

- in physics and "abstract" kinematics
- rotation: angular velocity
  - denoted with ω
  - measured in proper SI units
  - i.e., Radians per Second (rad/sec)

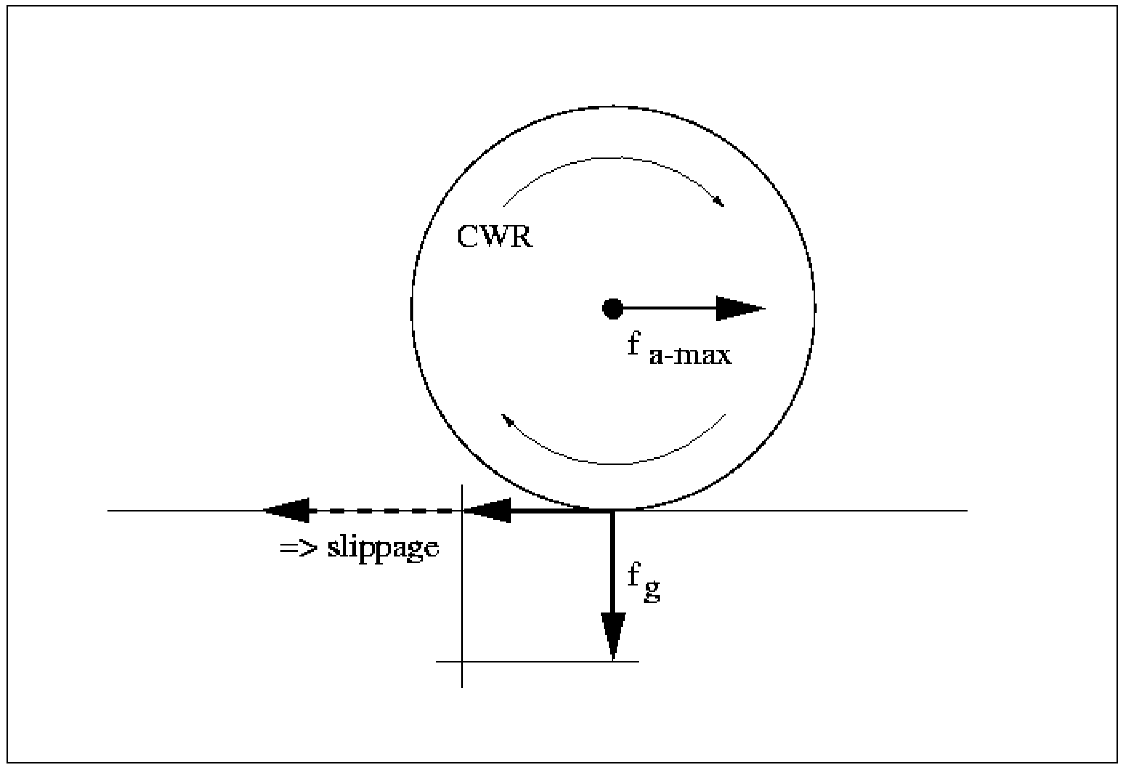
i.e., $\omega = 2\pi \cdot (N_a \,/\, 60\text{sec/min})$



Maximum Acceleration

note:

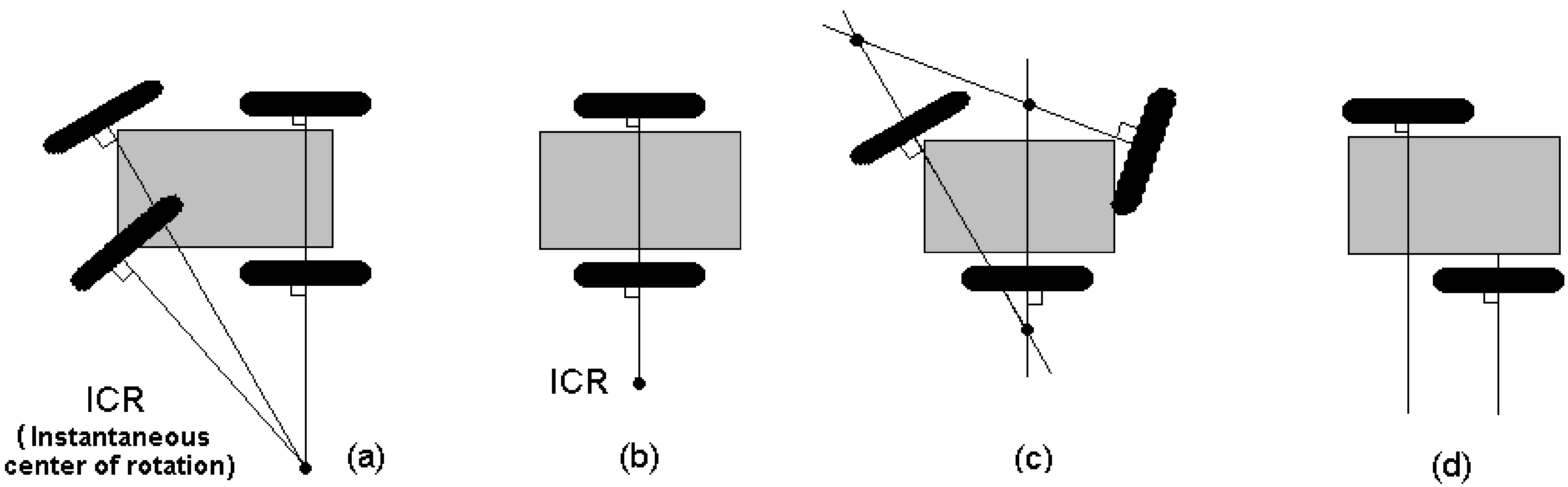
- propulsion requires traction
- even with optimal friction
- acceleration limited by gravitation
- $g = 9.8\ m/s^2$



Instantaneous Center of Curvature (ICC)

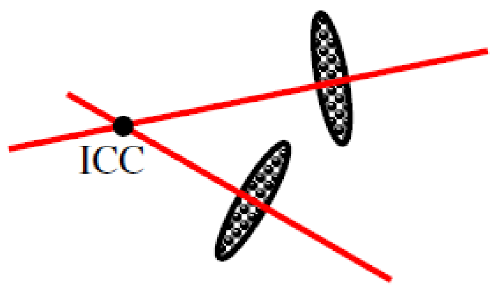
aka Instantaneous Center of Rotation (ICR)

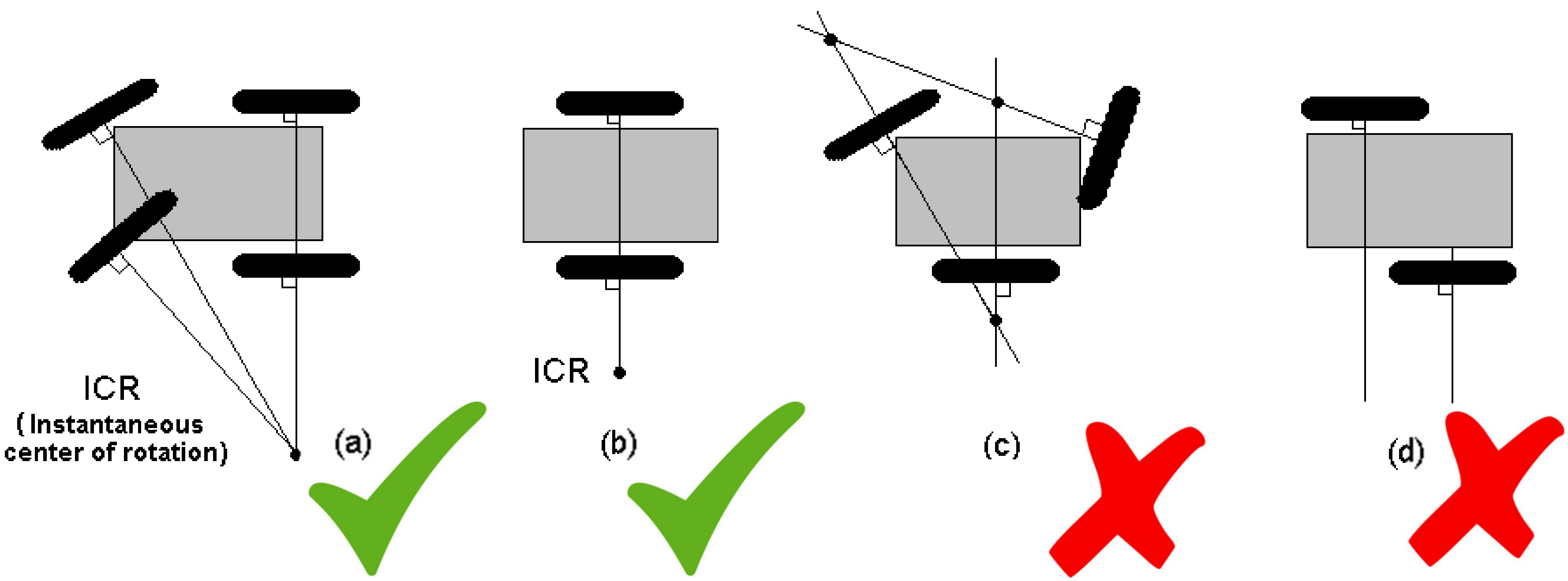
- point(s)
  - where (infinite) axes of the wheels of a system
  - cross each other



Instantaneous Center of Curvature (ICC)

there must be exactly one ICC
to allow rotation of the system



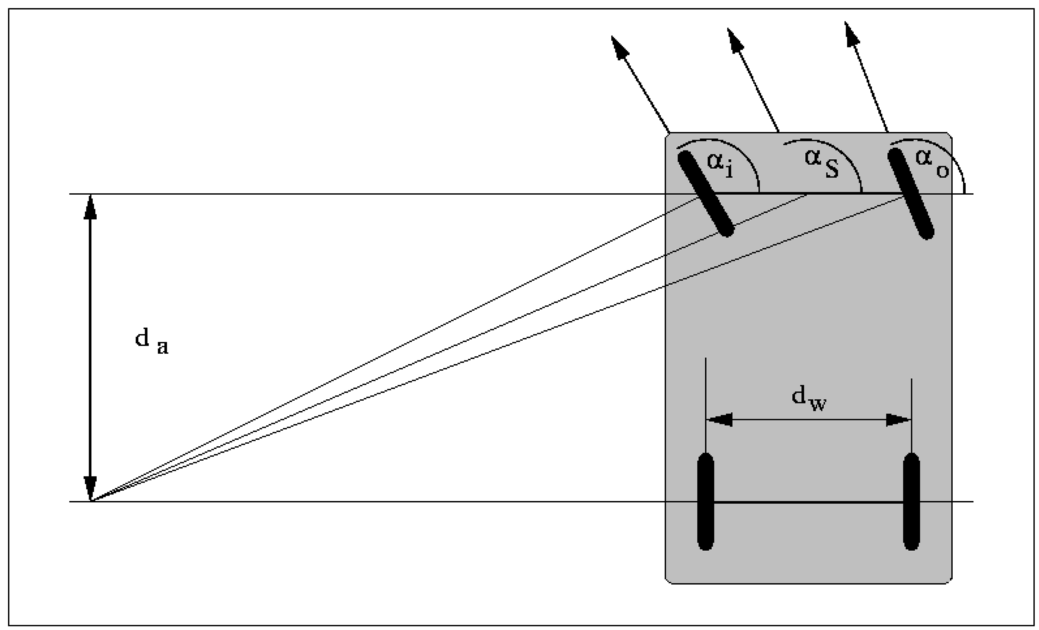


Example: Ackermann Drive

- known from automobiles
- front wheels
  - different steering angles
  - defined by Ackermann equation
  - to minimize slip in curves
- badly suited for robots
  - no turning on the spot
  - complex IK / path planning

$$cot(\alpha_i) - cot(\alpha_o) = d_w / d_a$$

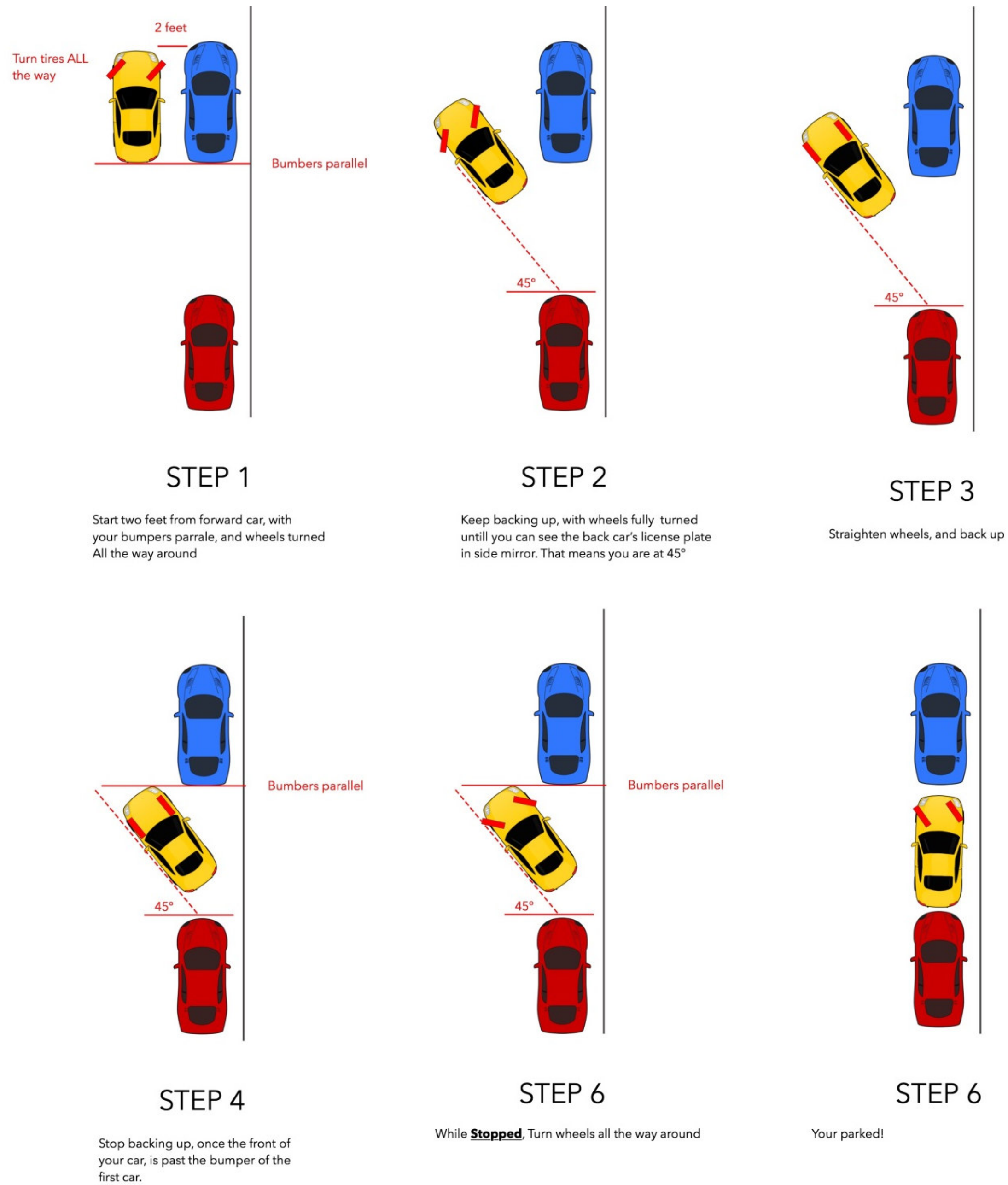
$$cot(\alpha_S) = \frac{cot(\alpha_i) + cot(\alpha_o)}{2}$$



Ackermann Drive

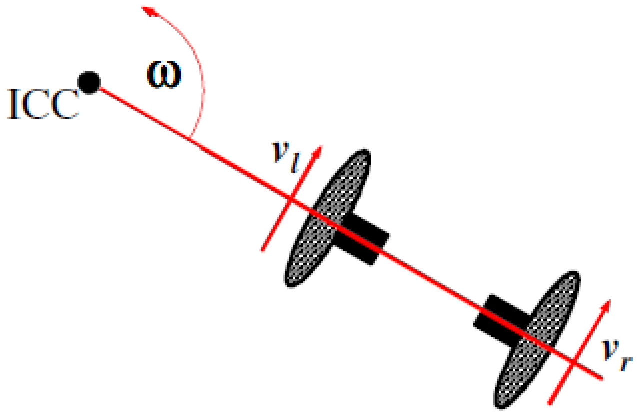
- badly suited for robots
  - no turning on the spot
  - complex IK / path planning
- see e.g. complications with parallel parking

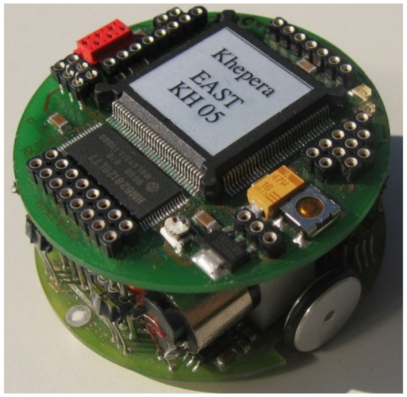
but nice example of single ICC (front wheels are not parallel!!!)

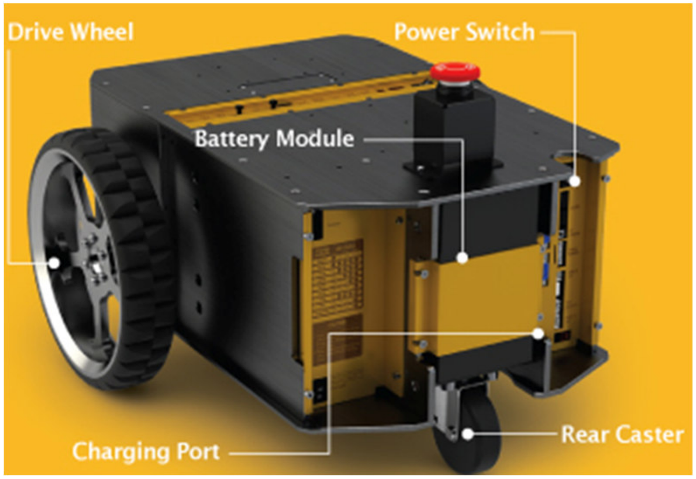


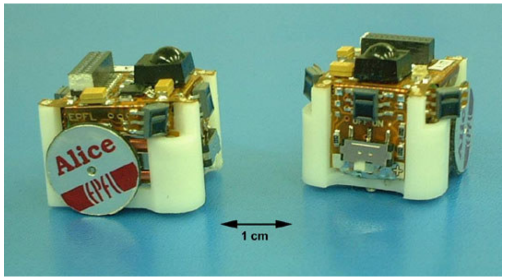
Differential Drive

- very popular robot drive
- two active wheels on one virtual axis
- plus (at least) one caster (passive wheel) for support









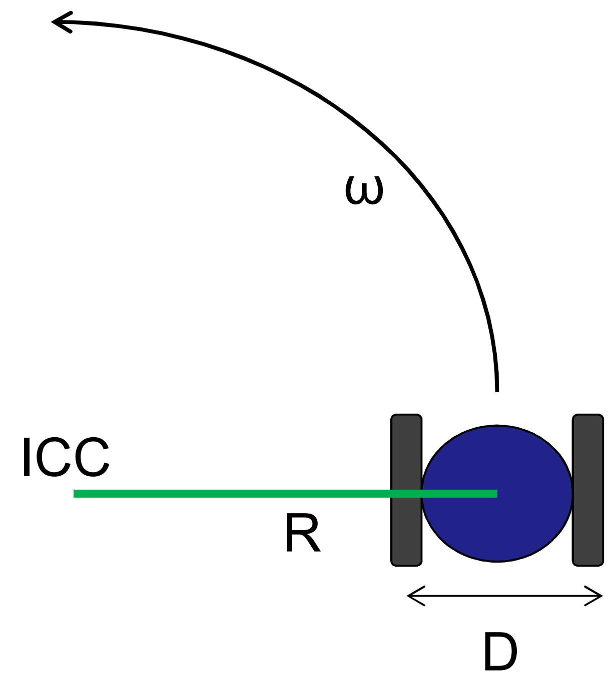
Differential Drive

motion = rotation ω around ICC with radius R with

- $v_r = \omega\,(R + D/2)$
- $v_l = \omega\,(R - D/2)$

$$R = \frac{D}{2}\,\frac{v_r + v_l}{v_r - v_l}$$

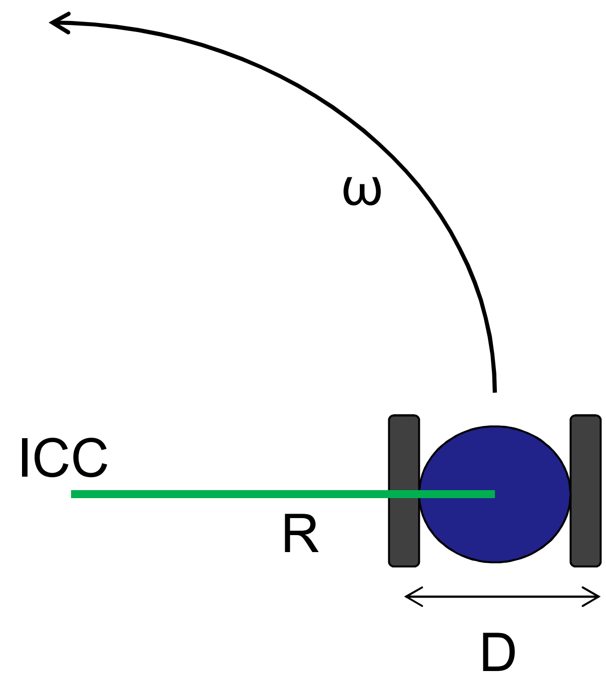
$$\omega = \frac{v_r - v_l}{D}$$



Differential Drive

$$R = \frac{D}{2}\frac{v_r + v_l}{v_r - v_l}$$

$$\omega = \frac{v_r - v_l}{D}$$



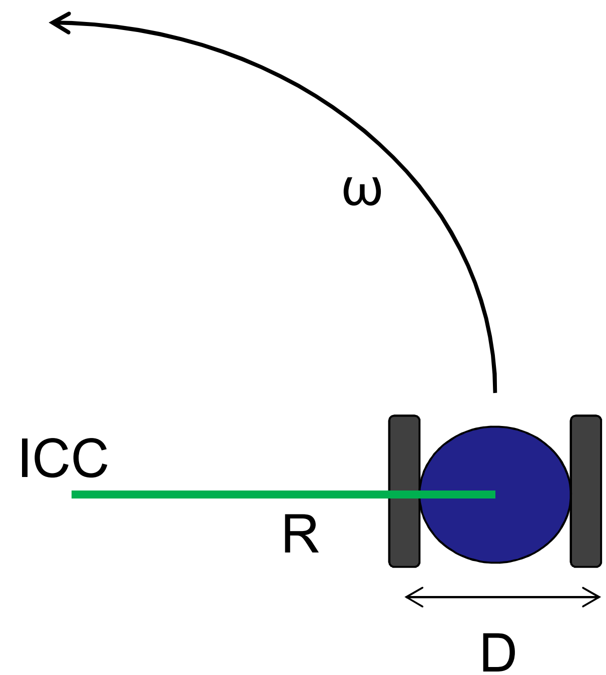
motion in an **arc** (around ICC), esp.

- $v_r = v_l$: R infinite, ω Zero, straight motion
- $v_r = -v_l$: R Zero, rotate in place
- $v_{l/r} = 0$: ICC = left/right wheel, i.e., rotate around l./r. wheel

Differential Drive

robot pose $(x,y,\theta)^T$

motion in time Δt with fixed velocities v_r and v_l

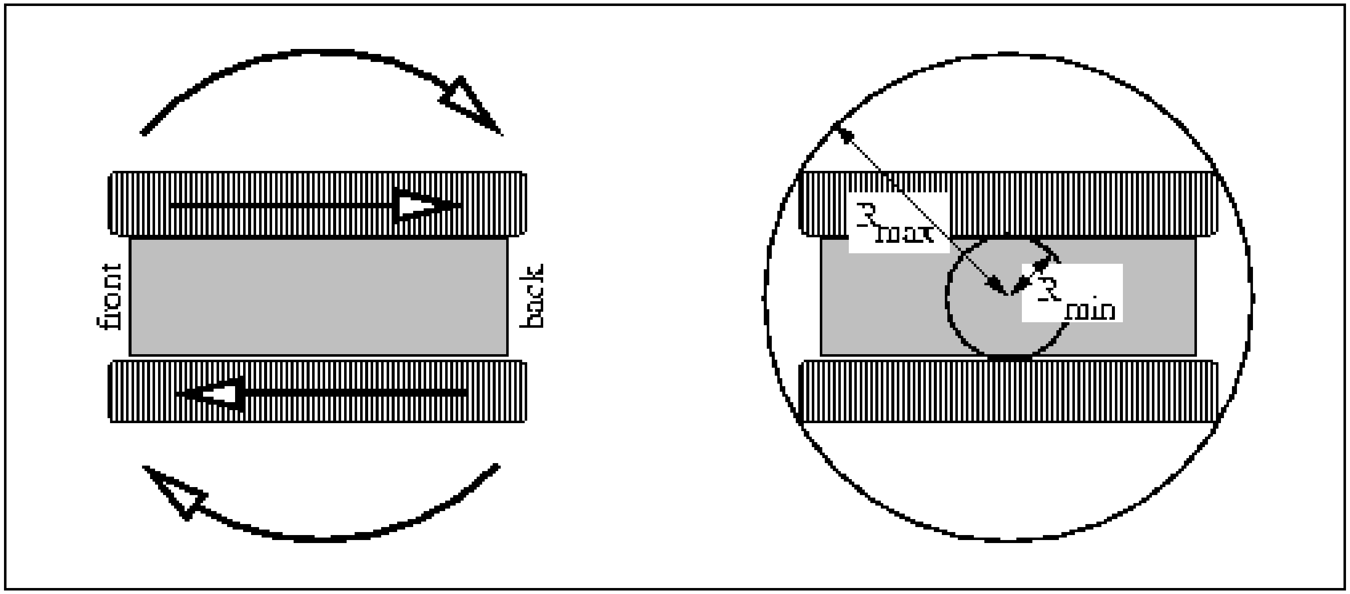


$$p_{ICC} = \left(x_{ICC}, y_{ICC}\right)^T$$

$$= \left(x - R\sin(\theta), y + R\cos(\theta)\right)^T$$

$$\begin{pmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t - x_{ICC} \\ y_t - y_{ICC} \\ \theta_t \end{pmatrix} + \begin{pmatrix} x_{ICC} \\ y_{ICC} \\ \omega\Delta t \end{pmatrix}$$

Tracked Drive



- model: differential drive
- but
  - relies on slip
  - "wheel distance" difficult to determine

Differential Drive Variants

four (or more) active wheels

- driven in parallel on each side (no caster(s) needed)
- similar to tracked drive
- i.e., differential drive (with imprecise “wheel distance”)

Mobile Robot: Inverse Kinematics

how to get from start pose $(x_s, y_s, \theta_s)^T$ to goal pose $(x_g, y_g, \theta_g)^T$?

- consider w.l.o.g. $(x_s, y_s, \theta_s)^T = (0,0,0)^T$

$$\mathrm{x}(\mathrm{t}_g) = \int_0^{\mathrm{t}_g} \mathrm{v}(\mathrm{t}) \cos(\theta(\mathrm{t}))\,\mathrm{dt}$$

$$\mathrm{y}(\mathrm{t}_g) = \int_0^{t} \mathrm{v}(\mathrm{t}) \sin(\theta(\mathrm{t}))\,\mathrm{dt}$$

$$\theta(\mathrm{t}_g) = \int_0^{\mathrm{t}_g} \omega(\mathrm{t})\,\mathrm{dt}$$

- holds for any robot that moves with
  - translational velocity v(t)
  - angular velocity ω(t)

Differential Drive: Inverse Kinematics

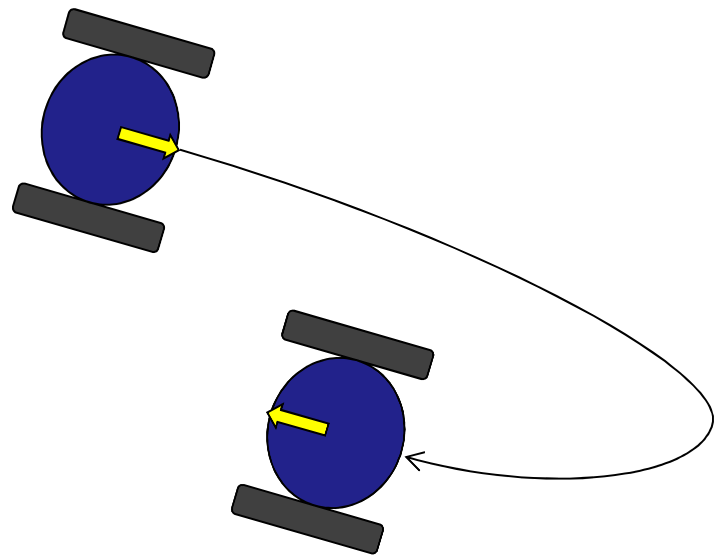
how to get from pose $(x_s,y_s,\theta_s)^T$ to pose $(x_g,y_g,\theta_g)^T$?

need to solve

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

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

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



in case of Differential Drive with

$$\omega = (v_r - v_l)/D$$

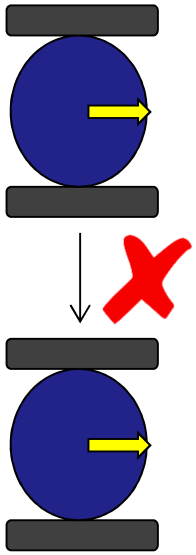
$$v = \omega R = (v_r + v_l)/2$$

for $v_L(t)$ and $v_R(t)$

Differential Drive: Inverse Kinematics

problem

- differential drive can not do arbitrary motions
- especially, no lateral motions (sidewise)
- aka in general as **non-holonomic constraints**



$\Rightarrow$ no straightforward mapping from $(x_g, y_g, \theta_g)^T$ to $v_L(t)$ and $v_R(t)$

solution: piece-wise combination of motion primitives

Differential Drive: Inverse Kinematics

piece-wise combination of motion primitives, e.g.

option 1: vector motions

1. rotate on the spot towards target position
2. move in straight line to target position
3. rotate on the spot to target orientation

hence only

- **rotations on the spot**
  with $-v_L = v_r = v$

$$\begin{pmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t + 2v\Delta t / D \end{pmatrix}$$

plus

- **straight line motion**
  with $v_L = v_r = v$

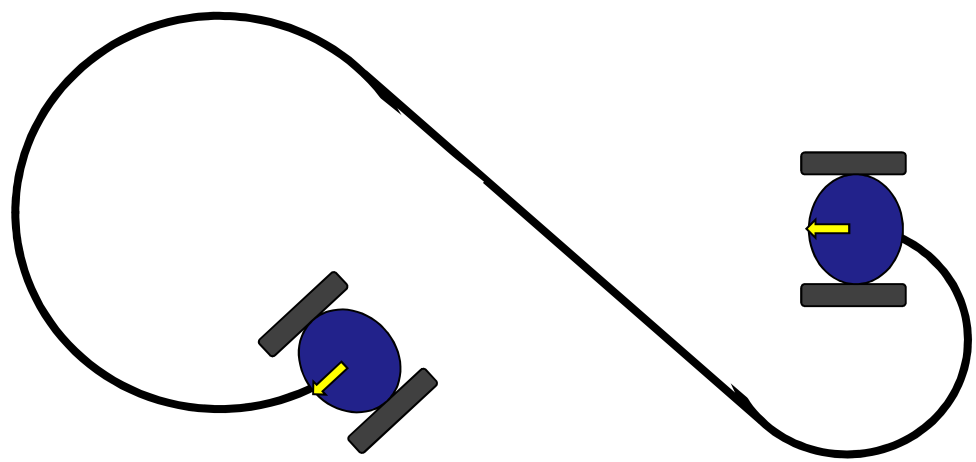
$$\begin{pmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{pmatrix} = \begin{pmatrix} x_t + v\cos(\theta_t)\Delta t \\ y_t + v\sin(\theta_t)\Delta t \\ \theta_t \end{pmatrix}$$

Differential Drive: Inverse Kinematics

piece-wise combination of motion primitives, e.g.

option 2: arcs & tangent lines

tangent line => smooth speed changes of the wheels



- lines can be derived from tangential constraint
- but which radii for the arcs?

Differential Drive: Inverse Kinematics

option 2: arcs & (tangent) lines

which radii for the arcs?

$\Rightarrow$ consider additional constraints

$\Rightarrow$ e.g.: minimize time t or energy E

$\Rightarrow$ need to consider **dynamics**
including forces and related parameters
(drive power, robot weight, etc.)

Differential Drive: Inverse Kinematics

option 2: arcs & (tangent) lines

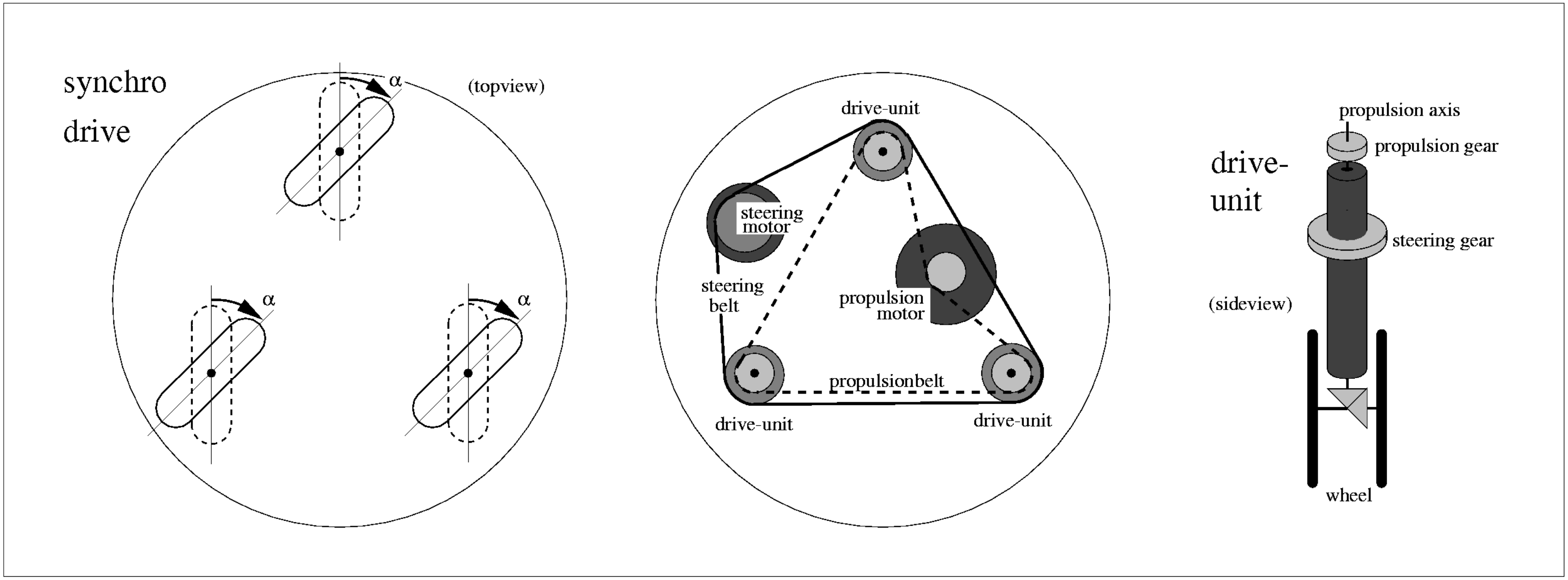
radii for the arcs => consider **dynamics**

in addition: obstacles in practice

$\Rightarrow$ matter of **path-planning** and **control**, i.e.,

- planning methods (see AI lecture) to generate path of way-points
- plus control loops (see Control lecture) to steer robot to next way-point

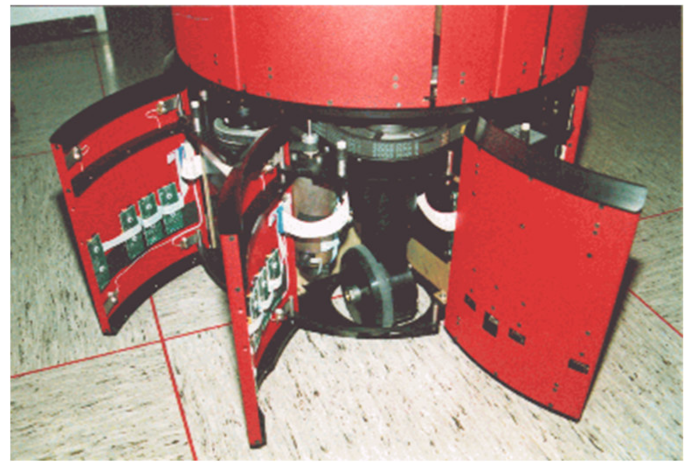
Synchro Drive



- attempt to be „more mobile“
- three active wheels
  - synchronously driven with same fixed velocity
  - all parallel, pointing in one direction
  - wheel orientations can be synchronously changed

Synchro Drive

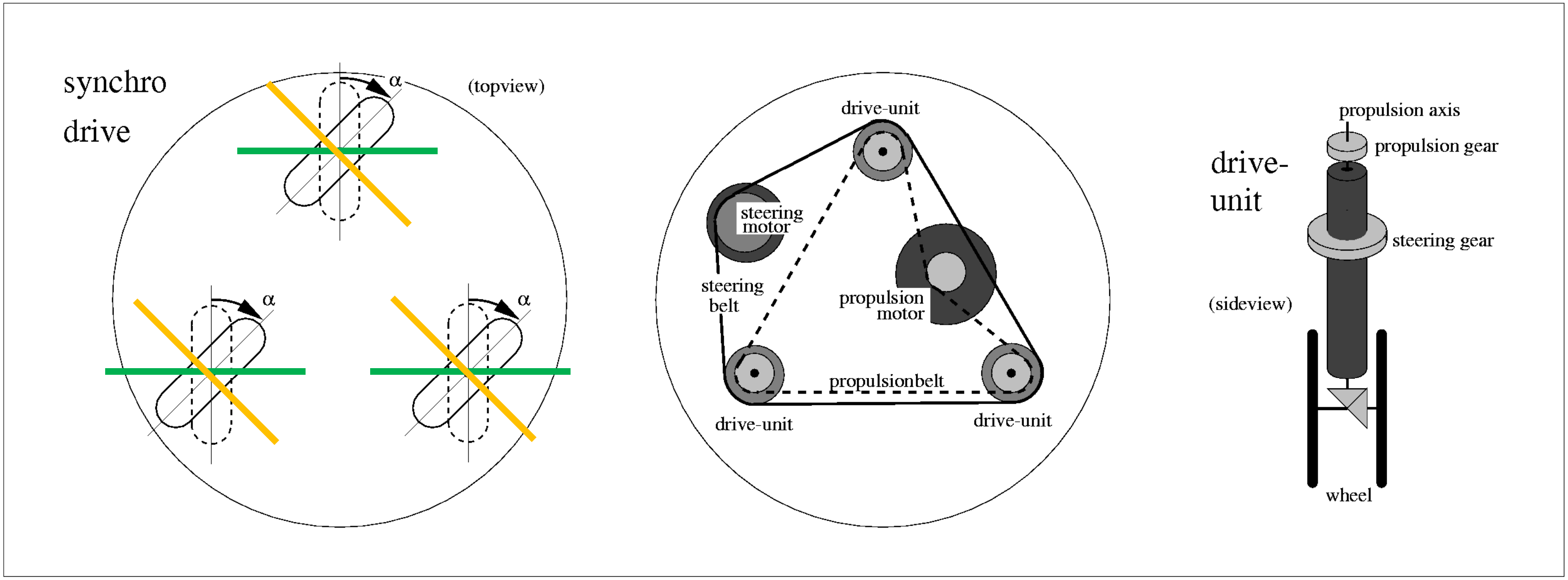
e.g., RWI (later iRobot) B-21 robot



Synchro Drive

- variants & extensions (wheels independently steerable)
- e.g., Neobotix MPO-700

Synchro Drive



- parallel axes => no ICC, respectively ICC at infinity
- but the ICC (at infinity) can be changed
- allows arbitrary rotation

nice(r): but not arbitrary rotation & translation combined

Omni-Drive

- omni-wheel
- aka swedish aka mecanum wheel
- additional passive rollers on the wheel

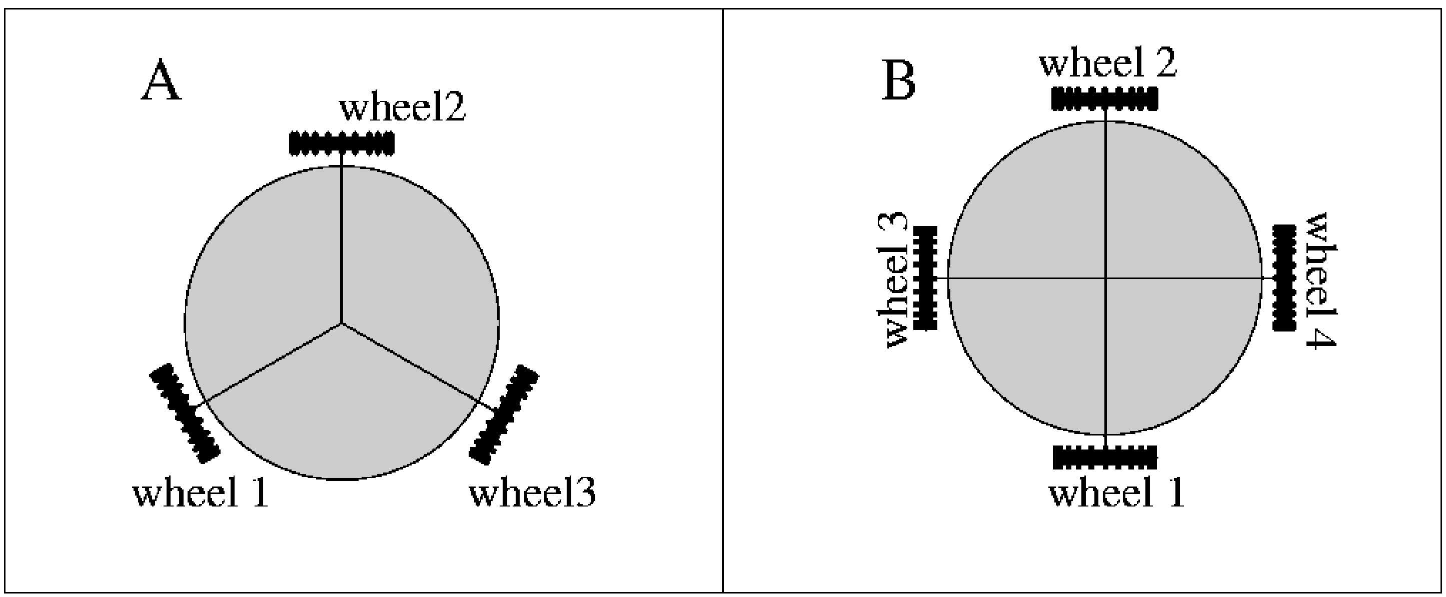
note:
meccanum and omni differ
wrt Kinematics






Omni-Drive

- why omni-wheels / drives?
  - 2D pose: 3 DOF
  - <3 active DOF: constraints on motion (non-holonomic)
- solution
  - ≥3 active DOF with omni-wheels



Omni-Drive

can hence move sideward

Omni-Drive

used from toy to high-end industrial

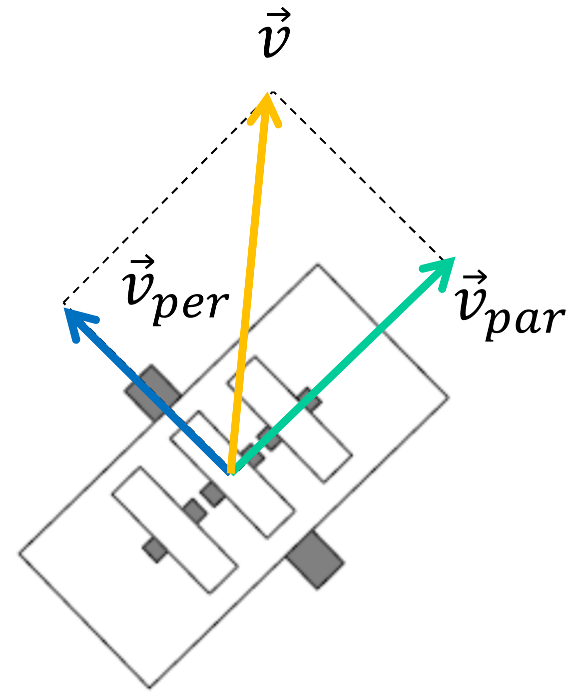
Omni-Drive

and not only robots...

Omni-Drive Kinematics

omni-wheel (perpendicular passive rollers)

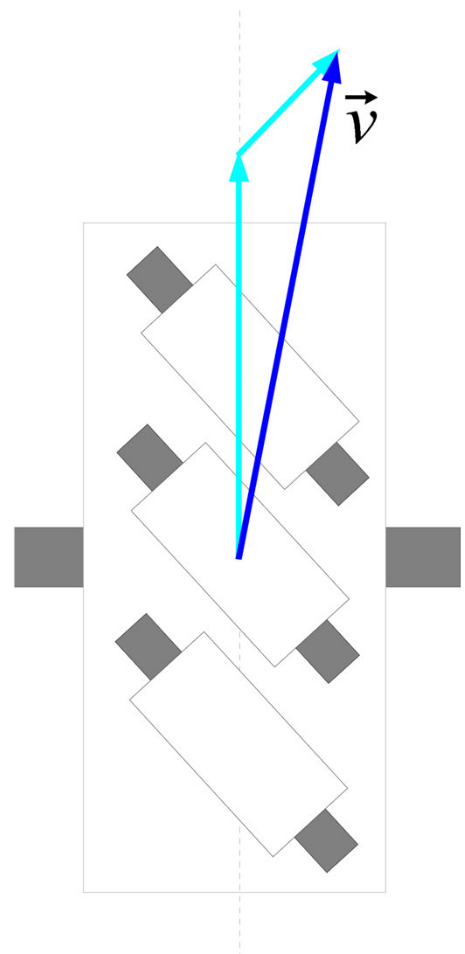
- wheel motion $\vec{v}$
- resolve into
  - parallel $\vec{v}_{par}$ and
  - perpendicular component $\vec{v}_{per}$



Omni-Drive Kinematics

mecanum wheel
(passive rollers at $\alpha \neq 90^o$)

- typically $\alpha = 45^o$
- analysis similar to omni-wheel
- split into parallel to wheel and parallel to roller (i.e., perpendicular to each axis)



Multiple Wheels (in general)

$\vec{v}_t$: translational velocity

$\vec{\omega}$: angular velocity

of the robot (at ICC)

point p = (x_p, y_p) (resp. vector $\vec{p}$)

- could be arbitrary point (relative to ICC)
- here: wheel mounted relative to ICC

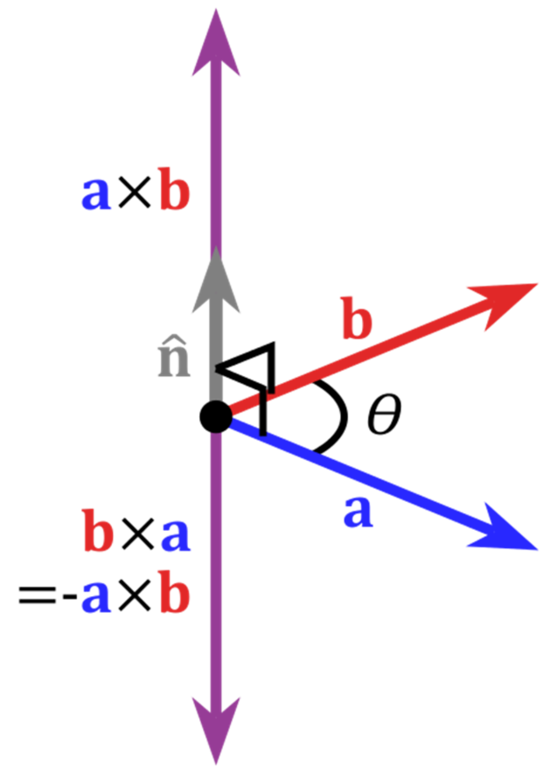
what is the velocity $\vec{v}$ of p?

Angular Velocity from a Vector Viewpoint

recap: a x b

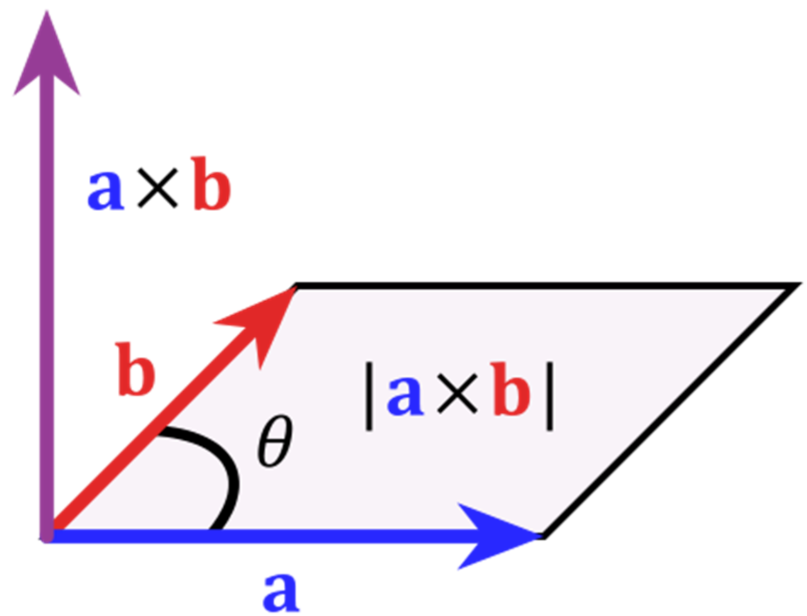
- cross product aka vector product
- of 2 linear independent vectors a,b
- is perpendicular to both a and b
- and hence to the plane of a and b
- following right hand rule

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 \cdot b_3 - a_3 \cdot b_2 \\ a_3 \cdot b_1 - a_1 \cdot b_3 \\ a_1 \cdot b_2 - a_2 \cdot b_1 \end{pmatrix}$$



Angular Velocity from a Vector Viewpoint

- magnitude of the cross product a x b
  - is the positive area of the parallelogram
  - having a and b as sides
- i.e., ||a x b|| = ||a|| ||b|| sin(θ)
  - with ||.|| is the Euclidean norm
  - i.e., $||x|| = \sqrt{{x_1}^2 + \cdots + {x_n}^2}$

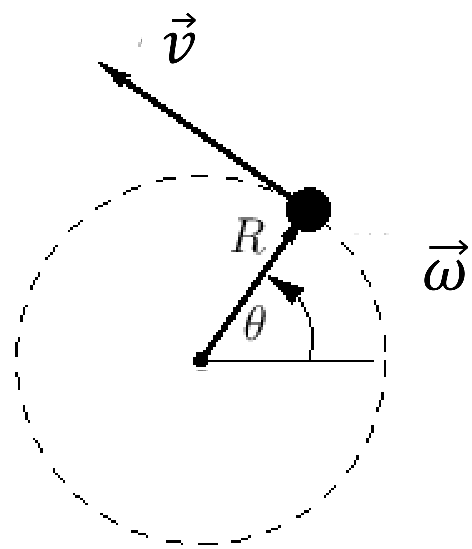


Angular Velocity from a Vector Viewpoint

angular velocity $\vec{\omega}$

- rotation is always around an axis $\vec{a}$
- tangential velocity $\vec{v}$
  - of point p at $\vec{R}$
  - rotating with $\vec{\omega}$

$$\vec{\boldsymbol{v}} = \vec{\boldsymbol{\omega}} \times \vec{\boldsymbol{R}}$$



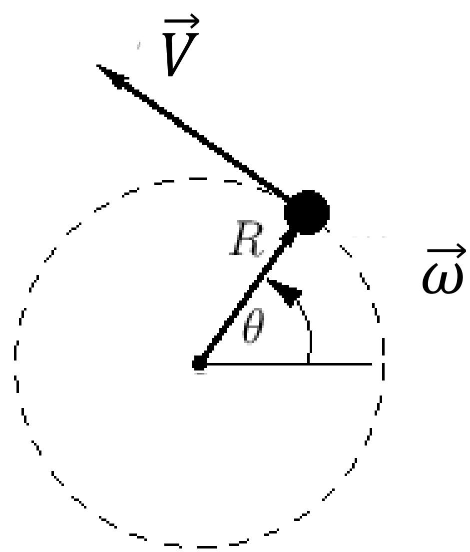
Angular Velocity from a Vector Viewpoint

- rotation around an axis
  - usually **orthonormal axis $\vec{u}$**, i.e., $\theta_{\vec{u}} = \pi/2$ & $||\vec{u}|| = 1$
  - and hence scalar ω sufficient
    (ω change of angle over time, i.e., $\omega = {}^{\partial\theta}/_{\partial t} = \dot{\theta}$)
- $\vec{u}$ orthonormal:

$$\left\|\vec{V}\right\| = \left\|\vec{\omega} \times \vec{R}\right\|$$

$$= \left\|\vec{\omega}\right\| \cdot \left\|\vec{R}\right\| \cdot \underbrace{\sin(\theta_{\vec{u}})}_{=\sin(\pi/2)=1}$$

$$= \omega \cdot \left\|\vec{R}\right\| = \omega \cdot R$$



Multiple Wheels (in general)

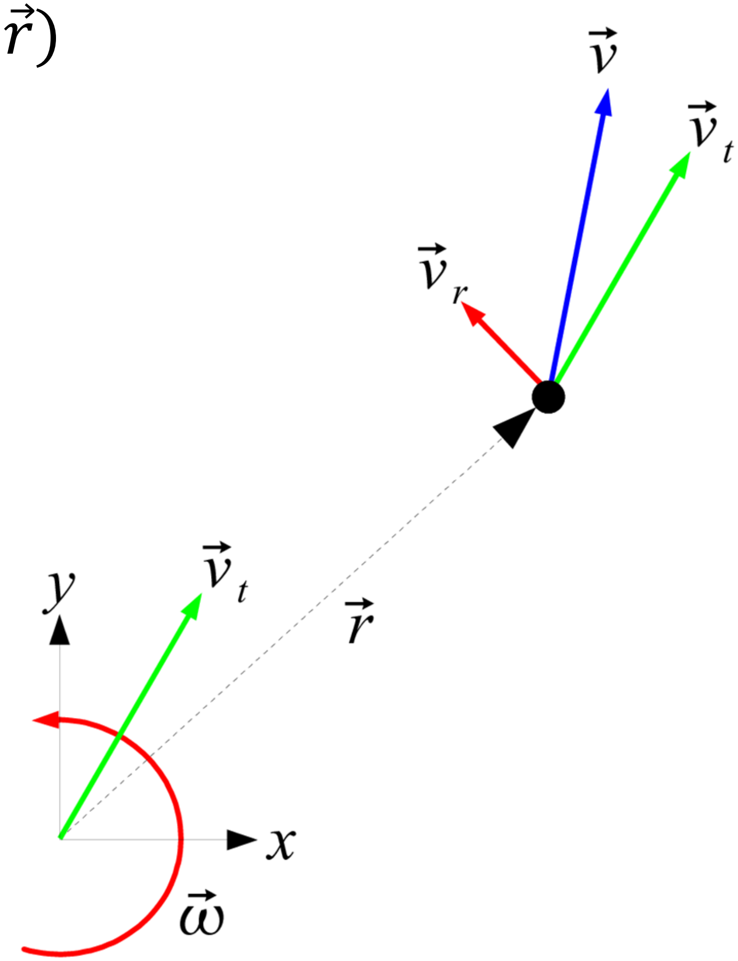
- wheel at r = (x_r, y_r) (resp. vector $\vec{r}$)
- split robot $\vec{v}$ in two components:
  - tangential speed $\vec{v}_r$
  - translation $\vec{v}_t$

in vector notation:

$$\vec{v} = \vec{v}_t + \vec{\omega} \times \vec{r}$$

or in scalars:

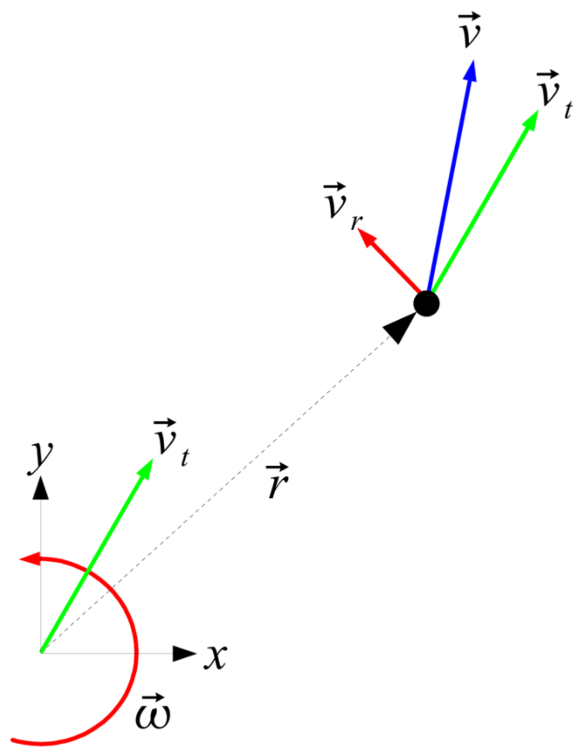
$$v_x = v_{t.x} - \omega \cdot y_r$$
$$v_y = v_{t.y} + \omega \cdot x_r$$



Multiple Wheels (in general)

note!!!

$$v_x = v_{t.x} - \omega \cdot y_r$$
$$v_y = v_{t.y} + \omega \cdot x_r$$



$$v_r = \begin{pmatrix} v_{r.x} \\ v_{r.y} \\ v_{r.z} \end{pmatrix}$$

$$= \left(\omega \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) \times \begin{pmatrix} x_r \\ y_r \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \cdot 0 - \omega \cdot y_r \\ \omega \cdot x_r - 0 \cdot 0 \\ 0 \cdot y_r - 0 \cdot x_r \end{pmatrix}$$

$$= \begin{pmatrix} -\omega \cdot y_r \\ \omega \cdot x_r \\ 0 \end{pmatrix}$$

Omni-Drive Example

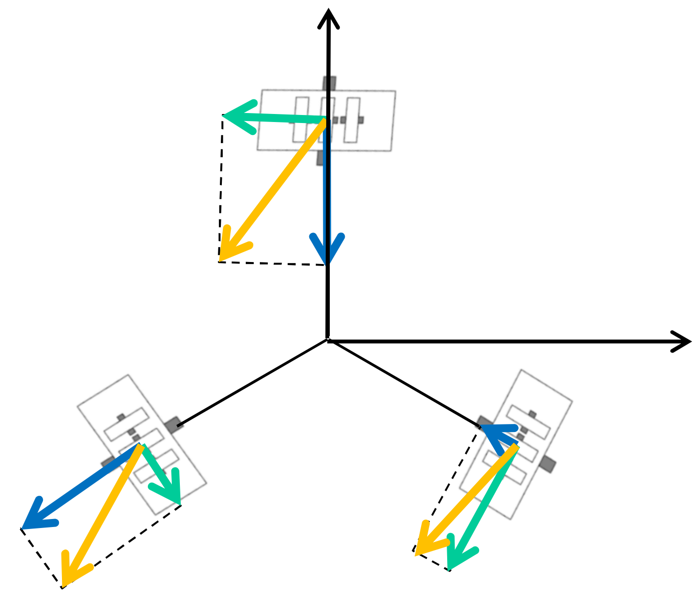
three omni wheels: evenly distributed

- $\alpha_1 = 90^o$
- $\alpha_2 = 210^o$
- $\alpha_3 = 330^o$

with distance *R* to robot center

(global) robot pose: $(x, y, \theta)^T$

(global) robot velocity: $v = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$



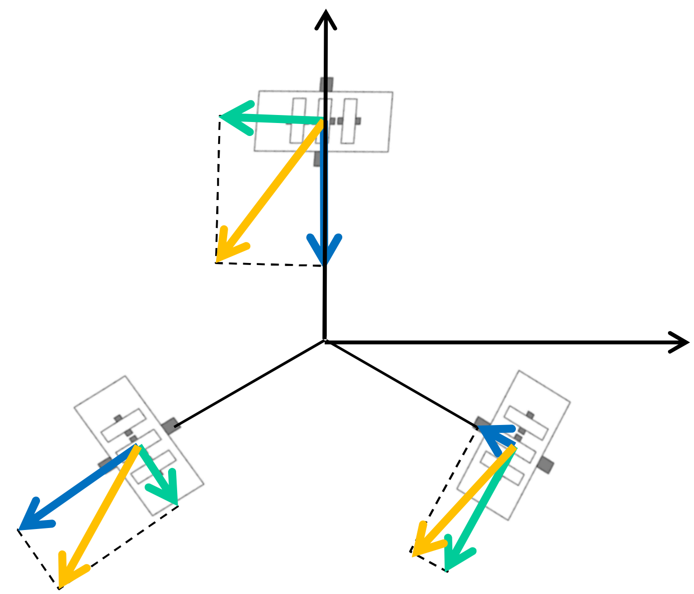
Omni-Drive Example

wheel *i* : velocity $v_i = v_{ti} + v_r$

- (local) translation: $v_{ti} = -\sin(\alpha_i)\dot{x} + \cos(\alpha_i)\dot{y}$
- (local) rotation: $v_r = R\dot{\theta}$

wheel parameters

- speed: $\dot{\varphi}_i$
- (fixed) radius: *r*



$$\dot{\varphi}_i = \frac{1}{r}(-\sin(\alpha_i)\,\dot{x} + \cos(\alpha_i)\,\dot{y} + R\dot{\theta})$$

Omni-Drive Example

Inverse Kinematics

$$\begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{pmatrix} = \frac{1}{r}\begin{pmatrix} -\sin(\alpha_1) & \cos(\alpha_1) & R \\ -\sin(\alpha_2) & \cos(\alpha_2) & R \\ -\sin(\alpha_3) & \cos(\alpha_3) & R \end{pmatrix}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$

$$= \frac{1}{r}\begin{pmatrix} -1 & 0 & R \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & R \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & R \end{pmatrix}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$

	sin	cos
$\boldsymbol{\alpha_1 = 90^o}$	1	0
$\boldsymbol{\alpha_2 = 210^o}$	$-1/2$	$-\sqrt{3}/2$
$\boldsymbol{\alpha_3 = 330^o}$	$-1/2$	$\sqrt{3}/2$

(in local robot frame)

Omni-Drive Example

Forward Kinematics

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\sin(\alpha_1) & \cos(\alpha_1) & R \\ -\sin(\alpha_2) & \cos(\alpha_2) & R \\ -\sin(\alpha_3) & \cos(\alpha_3) & R \end{pmatrix}^{-1} r \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix}$$

$$= \begin{pmatrix} -1 & 0 & R \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & R \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & R \end{pmatrix}^{-1} \begin{pmatrix} r\dot{\phi}_1 \\ r\dot{\phi}_2 \\ r\dot{\phi}_3 \end{pmatrix}$$

(in local robot frame)