Aalto University
School of Electrical Engineering

Locomotion, Kinematics, and Low level Motion Control

ELEC-E8111 Autonomous Mobile Robots

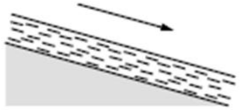
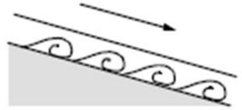
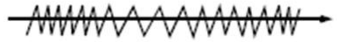
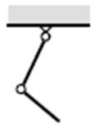
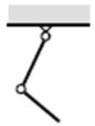
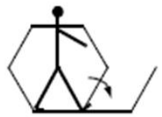
Arto Visala

5.3.2019

Overview

- Locomotion – Principles and mechanisms to make the robot move.
- Kinematics – How to model motion of (rigid) bodies?
- Motion control – How to make a robot attain a goal or follow a trajectory?

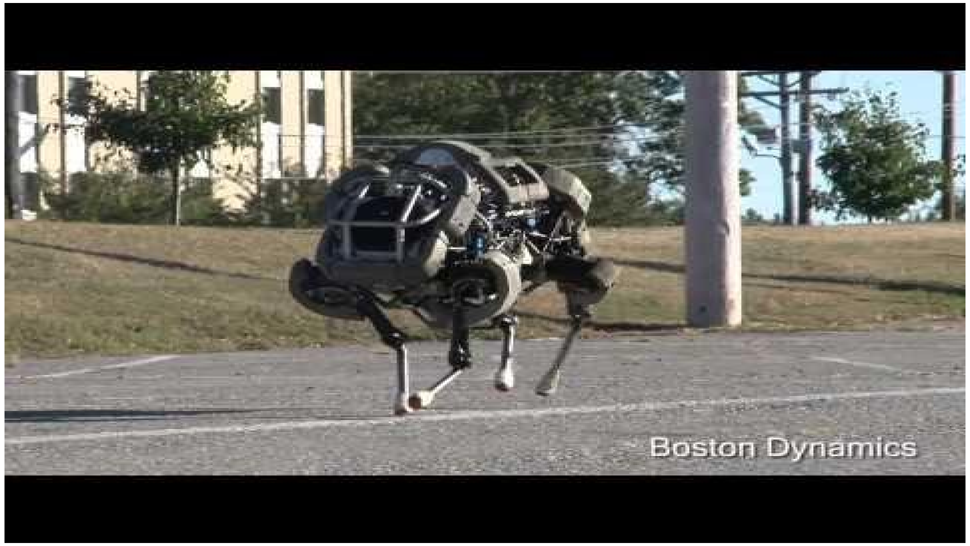
Locomotion – Principles

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) 

Siegwart & Nourbaksh

Locomotion – Principles

- Technical systems
  - Mostly wheels or tracks
- Research active in legged but also other types of locomotion
  - Aerial
  - In water
  - Sliding/Snake robots



Locomotion – Problems to solve

- Locomotion – physical interaction between a vehicle and its environment
- Concerned with **interaction forces,** and generating **mechanisms and actuators**
- Issues
  - **Stability/balancing** – number of contacts, center of gravity, static/dynamic stabilization, terrain inclination
  - **Contact characteristics** – contact type (point, area), friction
  - **Environment** – medium (soft or hard ground, water, air)

Locomotion – On this course

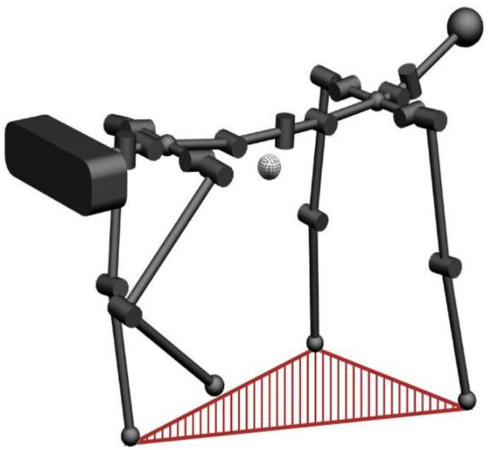
- Very Basics of legged locomotion
  - Why legged locomotion?
  - Number of legs
  - Stability
  - Static vs dynamic walking
- Mainly wheeled locomotion
  - Why wheeled locomotion?
  - Stability
  - Wheel types
  - Number of wheels and wheel arrangements

Legged locomotion

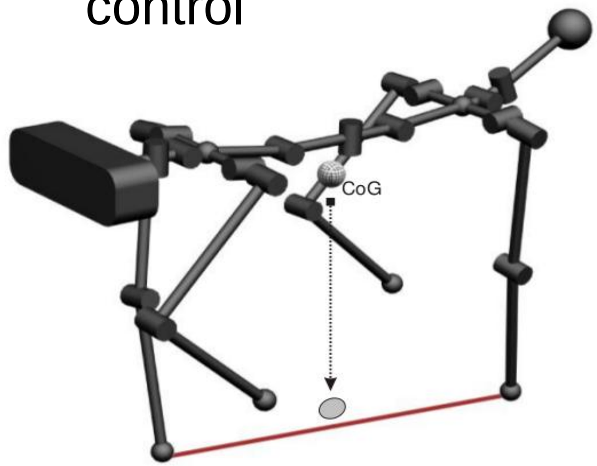
- Why legged locomotion?
  - Wheeled locomotion complicated on difficult terrain.
    - Climbing possible with legs.
- Number of legs
  - At least three point contacts needed for static stability.
    - One surface contact sufficient for static stability.
  - During walking some legs are lifted.
    - For static walking, at least four legs are needed.
  - More legs, the easier it is (to balance).

Static vs dynamic walking

- Static walking
  - COG always in support area.
  - Safe but slow.



- Dynamic walking
  - Fall if not moving.
  - Less than three legs with contact.
  - Fast but demanding to control



Walking forestry harvester

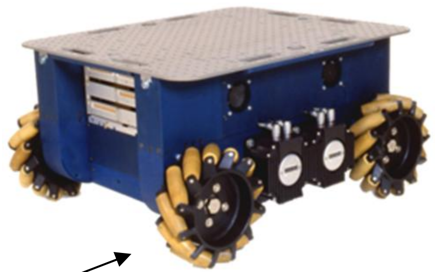
At TKK, Mecant research project lead by prof Aarne Halme, in which Plustech Oy was participant
http://autsys.aalto.fi/en/MECANT

Own prototype of Plustech Oy/ Timberjack (later John Deere Forest Oy)
http://www.youtube.com/watch?v=CD2V8GFqk_Y&feature=related
http://www.youtube.com/watch?v=CgBNjdwYdvE&feature=related

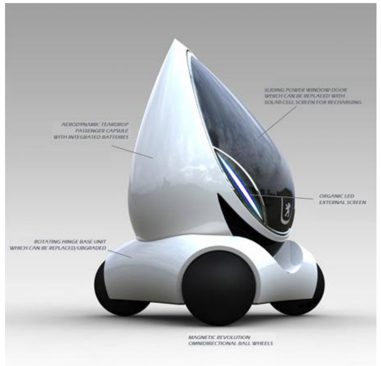
Wheeled locomotion

- Energy efficient.
- Appropriate for most practical applications.
- Three wheels sufficient for stability.
  - More wheels increase stability but require suspension.
- Bigger wheels allow to overcome higher obstacles.
- Selection of wheels depends on application.
  - What kind of wheels there are?

Basic wheel types, indoors



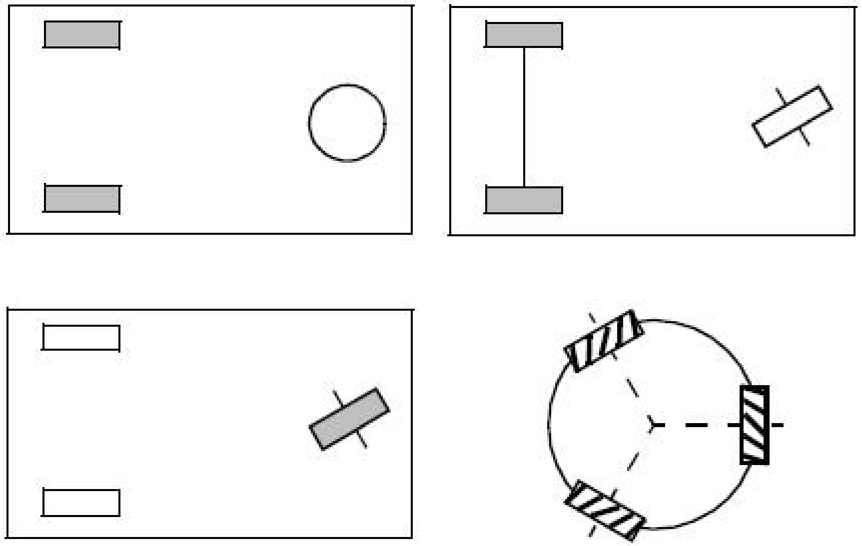
- Fixed standard wheel.
  - Two degrees of freedom: rotation around wheel axle (often motorized) and contact point.
- Steerable standard wheel.
  - Standard wheel + steering.
- Castor/caster wheel.
  - Three degrees of freedom.
- Swedish wheel.
  - Omnidirectional, controllable.
- Ball wheel.
  - Suspension difficult.



Typical wheel arrangements with two to three wheels

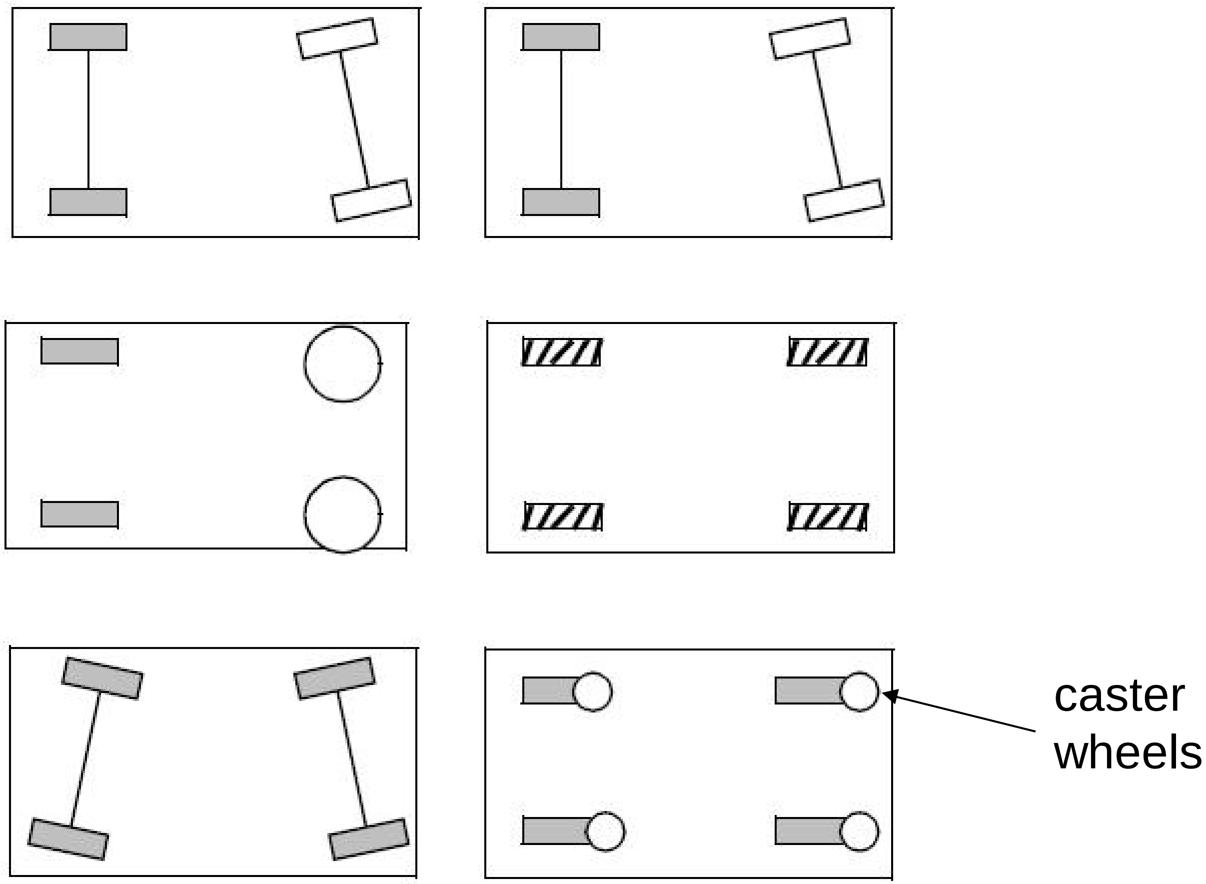
- Two wheels

- Three wheels



Typical wheel arrangements with four wheels

- Four wheels



Kinematics – Recap from basic robotics course

- Kinematics – modeling motion of bodies
  - *kinein* (Gr.) to move
- Forward kinematics $X = X(\theta)$
- Inverse kinematics $\theta = X^{-1}(X)$
- Joint space vs Cartesian space

- What's different in manipulator (arm) vs mobile robot kinematics?
  - What are joint values?
  - How to calculate *pose* from joint values?

Manipulator vs mobile robot kinematics

- Arm is a serial chain of links fixed to ground.
  - Mapping between joint angles and Cartesian pose exists.

$X = X(\theta)$ Exists!

- Wheeled robot motion is caused by rolling and sliding at wheel-ground contacts.
  - No direct mapping from joint angles (encoder values) to pose.
    - Called **non-holonomic** (not integrable) **system**
    - Path affects the state.
  - Encoder velocity maps to Cartesian velocity (*differential kinematics*).
  - Position must be integrated over time.

$X = X(\theta)$ Does not exist!

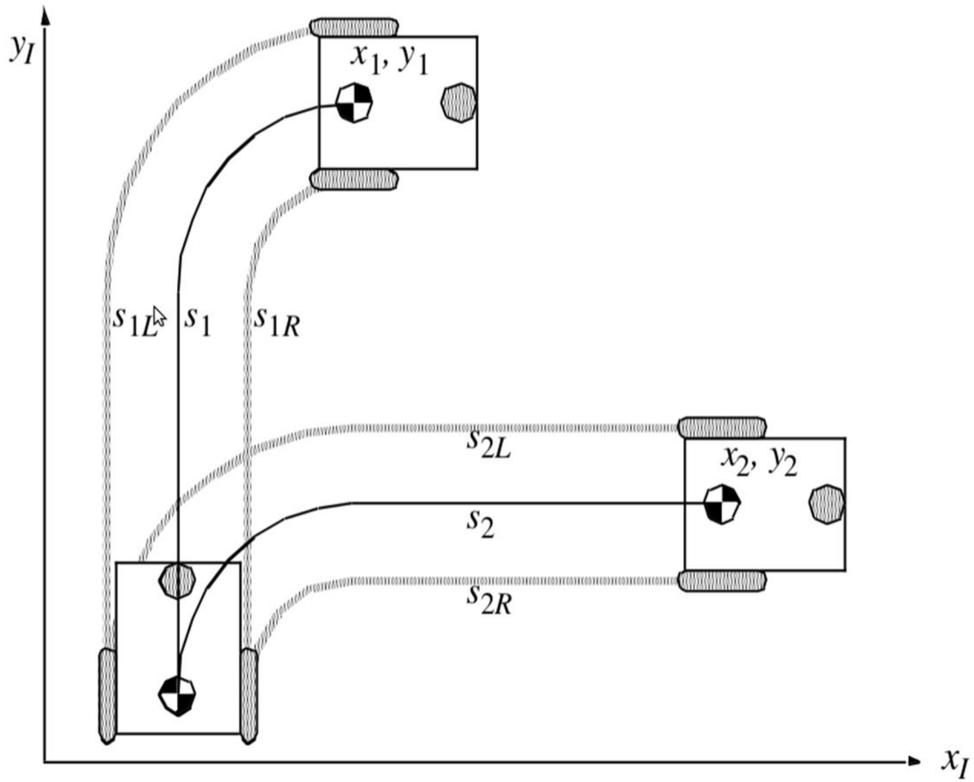
$\dot{X} = f(\dot{\theta})$ Exists!

Non-holonomic system illustrated

- Position must be integrated over time, depends on path taken
- Understanding mobile robot motion starts with understanding wheel constraints placed on the robot's mobility
- The measure of the traveled distance of each wheel is not sufficient to calculate the final position of the robot. One has also to know how this movement was executed as a function of time

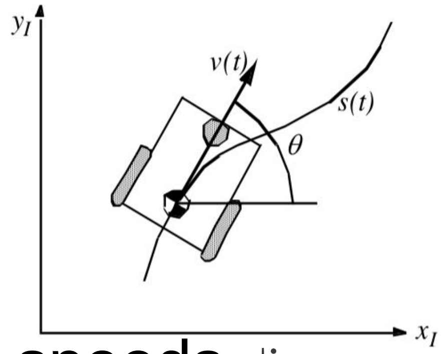
$$s_1 = s_2, s_{1R} = s_{2R}, s_{1L} = s_{2L}$$

$$x_1 \neq x_2, y_1 \neq y_2$$



Siegwart & Nourbaksh

Differential kinematics



- Robot speed $\dot{\xi} = (\dot{x}, \dot{y}, \dot{\theta})^T$ as function of wheel speeds $\dot{\phi}_i$, steering angles β_i and steering speeds $\dot{\beta}_i$
- Forward kinematics

These we can measure

$$(\dot{x}, \dot{y}, \dot{\theta})^T = f(\dot{\phi}_1, \ldots, \dot{\phi}_N, \beta_1, \ldots, \beta_M, \dot{\beta}_1, \ldots \dot{\beta}_M)$$

- Inverse kinematics

$$(\dot{\phi}_1, \ldots, \dot{\phi}_N, \beta_1, \ldots, \beta_M, \dot{\beta}_1, \ldots \dot{\beta}_M)^T = f^{-1}(\dot{x}, \dot{y}, \dot{\theta})$$

- Not integrable (in general) into

$$(x, y, \theta)^T = f(\phi_1, \ldots, \phi_N, \beta_1, \ldots, \beta_M)$$

Robot pose in world (inertial) vs local frame

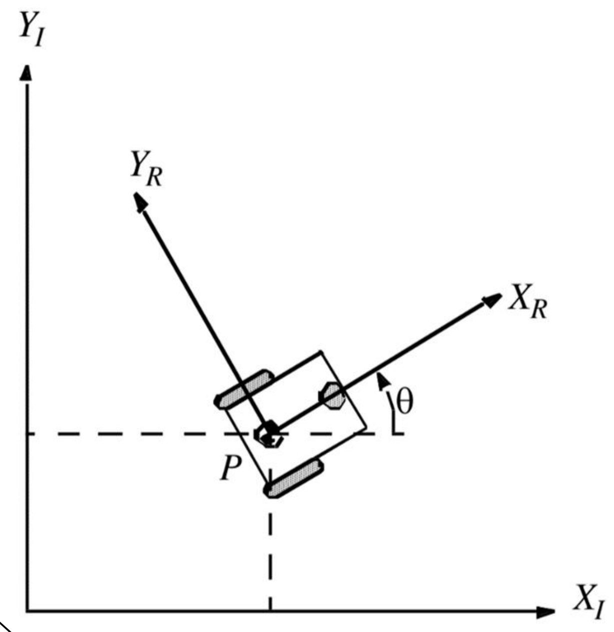
- Robot pose:

$$\xi_I = (x_I, y_I, \theta_I)^T$$

- Mapping velocities between frames:

$$\dot{\xi}_R = R(\theta_I)\dot{\xi}_I = R(\theta_I)(\dot{x}_I, \dot{y}_I, \dot{\theta}_I)^T$$

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

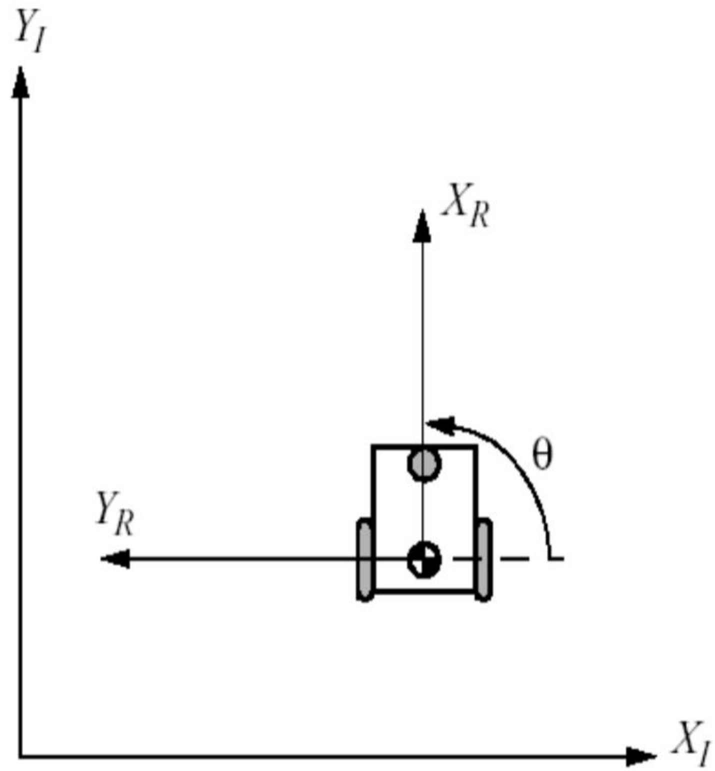


Remember this from basic robotics course?

Example: Mapping velocities between frames – Robot aligned with Y-axis

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

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



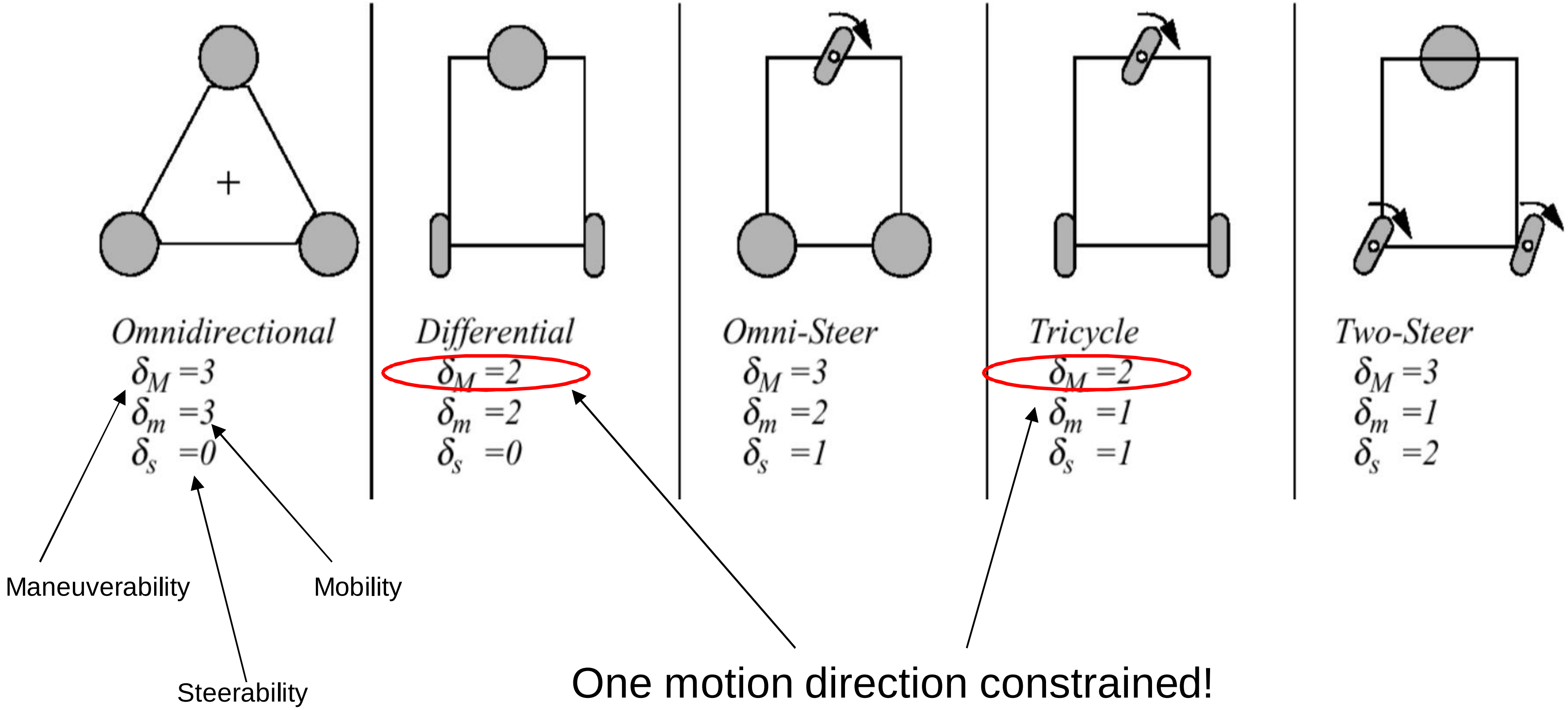
Wheel constraints

- Assumptions:
  - Pure rolling motion, no slipping, skidding or sliding
  - Point contact, no friction for rotation around contact point
  - Movement on horizontal plane, steering axis orthogonal to plane
  - No deformation of wheels, wheels connected by rigid structure
- Wheel constraints:
  - Sliding constraint: No motion along axes where wheel does not permit motion
  - Rolling constraint: Robot motion must be compatible to wheel speeds

Multiple wheels = complete robot

- Only fixed and steerable standard wheels impose constraints
- Three wheels sufficient for static stability
  - Additional wheels need to be synchronized (needed also for some 3-wheel settings)
- Maneuverability = mobility ("in how many directions can move") + steerability ("in how many directions can be steered")

Maneuverability for basic 3-wheel robots



Mobile robot workspace: DOF and DDOF

- Degrees of freedom (DOF) – ability to reach poses.
- Not all poses can be achieved “directly”
  - Cf. parallel parking.
- Differentiable degrees of freedom (DDOF) – ability to reach trajectories.

Holonomy of a robot

turning a crank

- Holonomic vs non-holonomic constraints:
  - Holonomic kinematic constraints can be written in terms of position variables.
  - Non-holonomic kinematic constraints need e.g. derivatives.
  - Fixed and steered standard wheels impose non-holonomic constraints.
- Non-holonomic robots:
  - For example, cars or tractor-trailers.
  - Can still be controllable to any pose.
  - Path planning for non-holonomic robots more difficult.

Purpose of differential equations, kinematic equations or real dynamic models

There are two purposes:

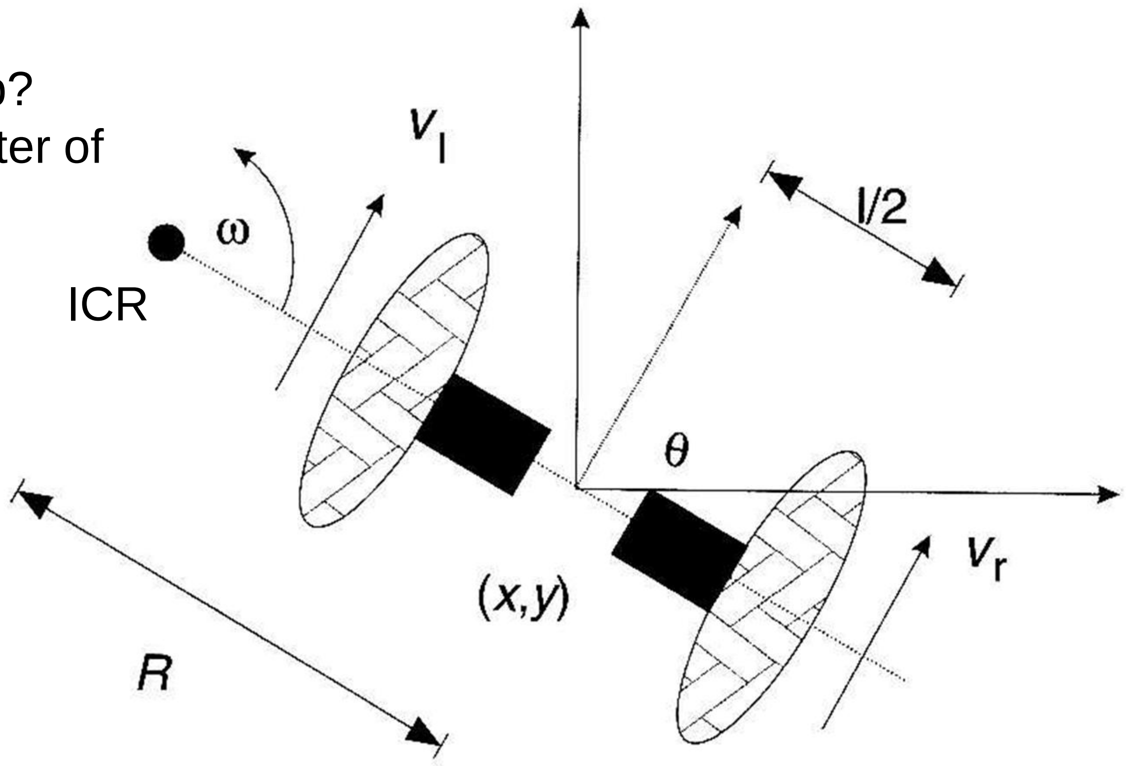
– For state estimation, especially odometry (inputs are measurements)

– For state prediction in predictive control (inputs are controls)

For Indoor, Kinematics for a differential drive robot

v, ω, R, l ?

What's their relationship?
ICR Instantaneous Center of Rotation



Dudek & Jenkin

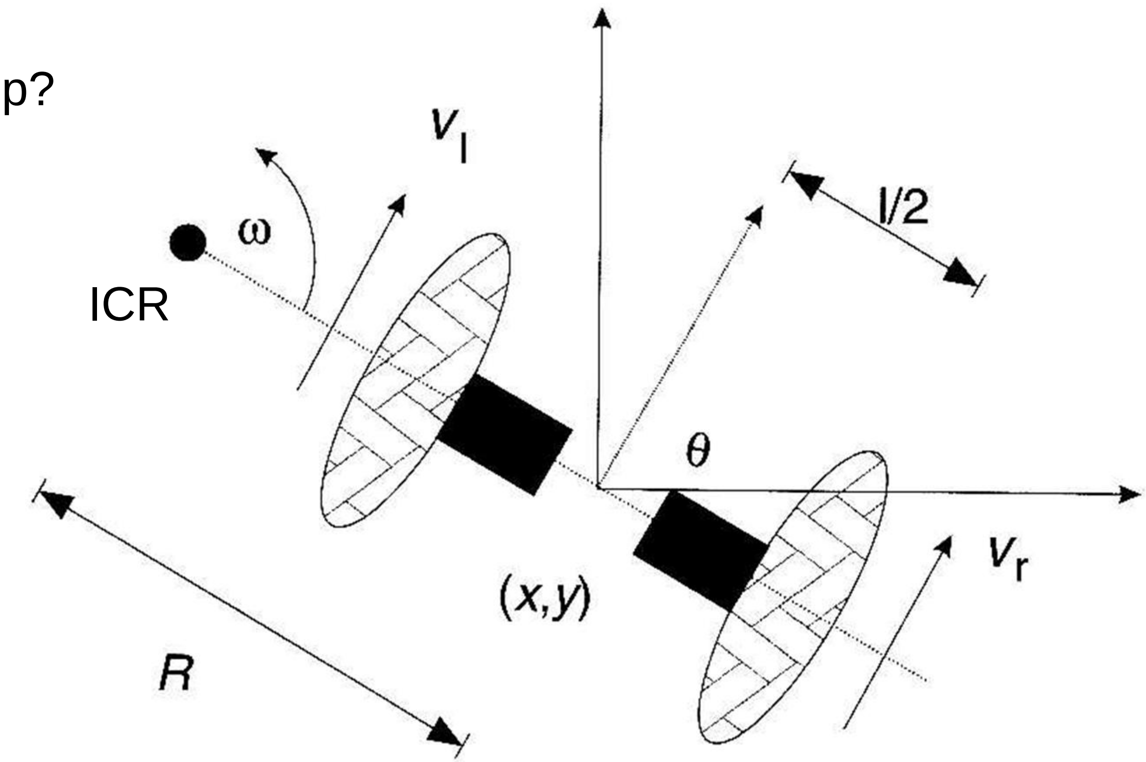
Kinematics for a differential drive robot

$v, \omega, R, l\ ?$

What's their relationship?

$$v_l = \omega(R - l/2)$$

$$v_r = \omega(R + l/2)$$



Dudek & Jenkin

Kinematics for a differential drive robot

$v, \omega, R, l\ ?$

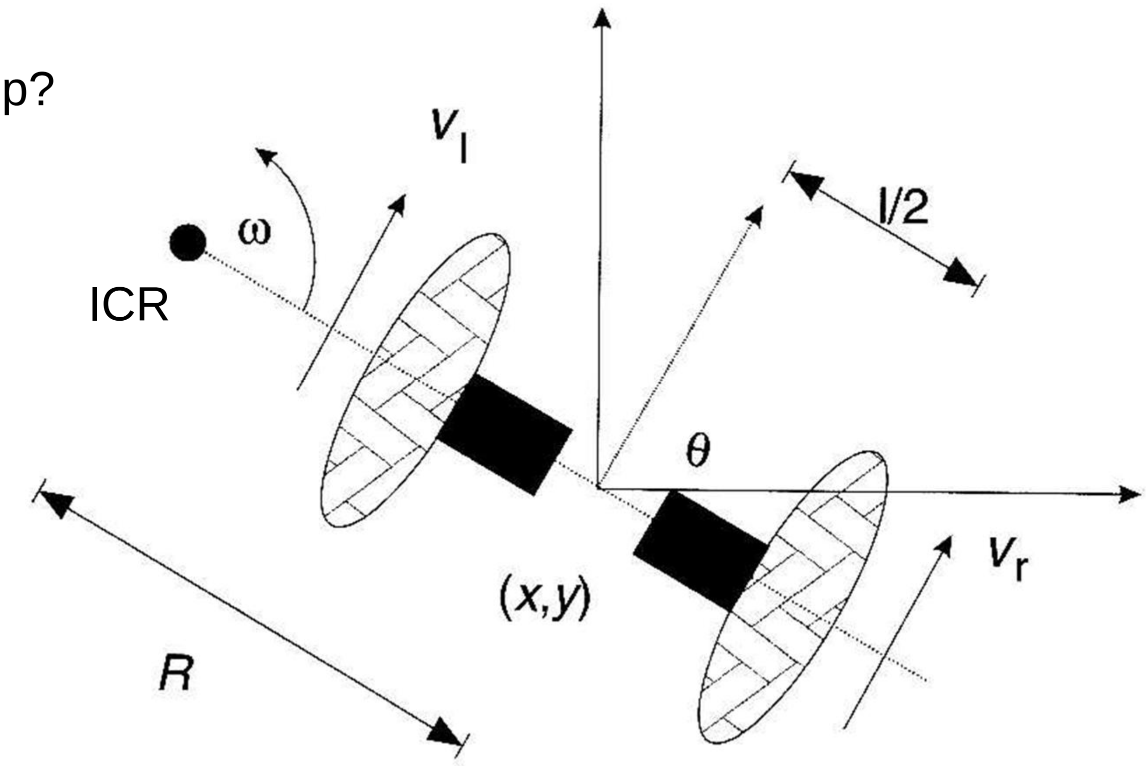
What's their relationship?

$$v_l = \omega(R - l/2)$$

$$v_r = \omega(R + l/2)$$

What if $v_l = v_r$?

What if $v_l = -v_r$?



Dudek & Jenkin

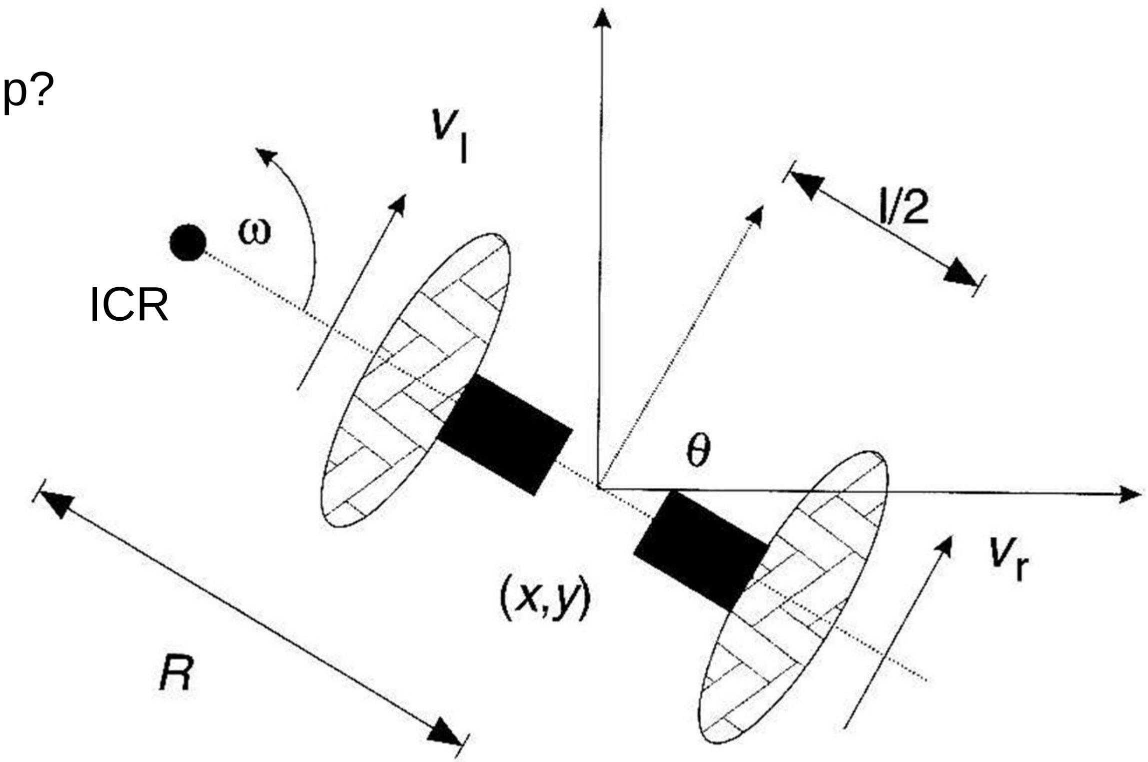
Kinematics for a differential drive robot

v, ω, R, l ?

What's their relationship?

$$v_l = \omega(R - l/2)$$

$$v_r = \omega(R + l/2)$$



What if $v_l = v_r$?

$$\omega = 0$$

What if $v_l = -v_r$?

$$R = 0$$

In general, $R = \frac{l}{2}\frac{v_r + v_l}{v_r - v_l}, \quad \omega = \frac{v_r - v_l}{l}$

Dudek & Jenkin

Kinematics for a differential drive robot

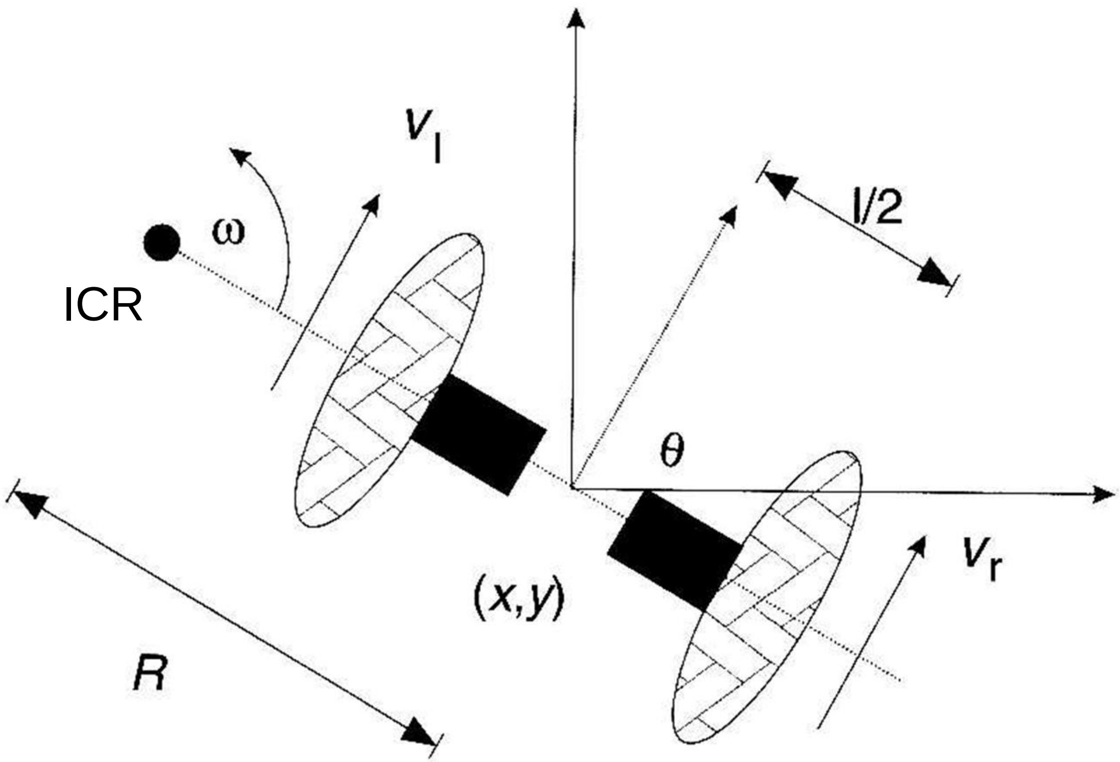
$$v_l = \omega(R - l/2)$$

$$v_r = \omega(R + l/2)$$

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

$$v = \frac{v_l + v_r}{2}$$

What are then the differential kinematics?



Dudek & Jenkin

Kinematics for a differential drive robot

$$v_l = \omega(R - l/2)$$

$$v_r = \omega(R + l/2)$$

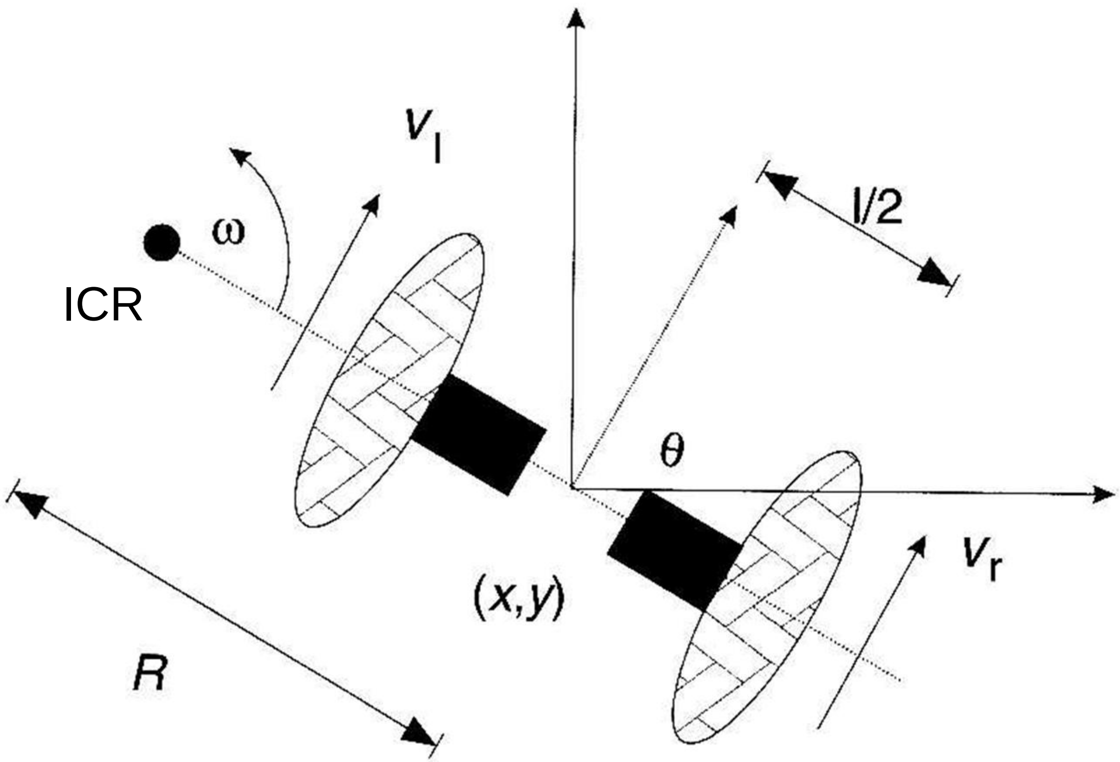
$$\omega = \frac{v_r - v_l}{l}$$

$$v = \frac{v_l + v_r}{2}$$

What are then the differential kinematics?

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$

$$\dot{\theta} = \omega = \frac{v_r - v_l}{l}$$



Dudek & Jenkin

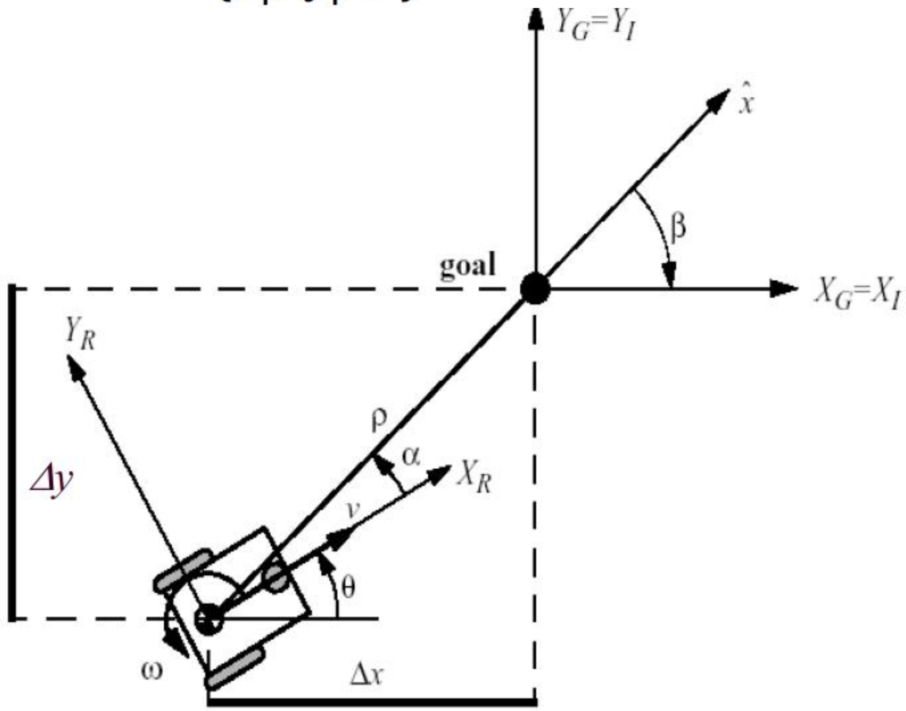
Kinematics for a differential drive robot (DD, like Pioneer)

The kinematics of a differential drive mobile robot described in the inertial frame $\{x_I, y_I, \theta\}$

is given by

$$ {}^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} $$

$$ v = \frac{v_l + v_r}{2} \qquad \omega = \frac{v_r - v_l}{l} $$



Motion control of wheeled DD-robots

- How to make a robot follow a trajectory?
- Trajectory (path) divided in motion segments of clearly defined shape:
  - Straight lines and segments of a circle
  - Dubins car, and Reeds-Shepp car
- Open-loop control
  - Divide trajectory to straight line, circular, and turn-in-place segments
  - No adaptation possible
  - Often non-smooth trajectories
  - Kinematics from the previous slides can be used to find how to execute the individual segments
- Can we do something else to reach a goal? Closed loop

Motion Control: Feedback Control

Find a control matrix K, if exists

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$ with $k_{ij}=k(t,e)$

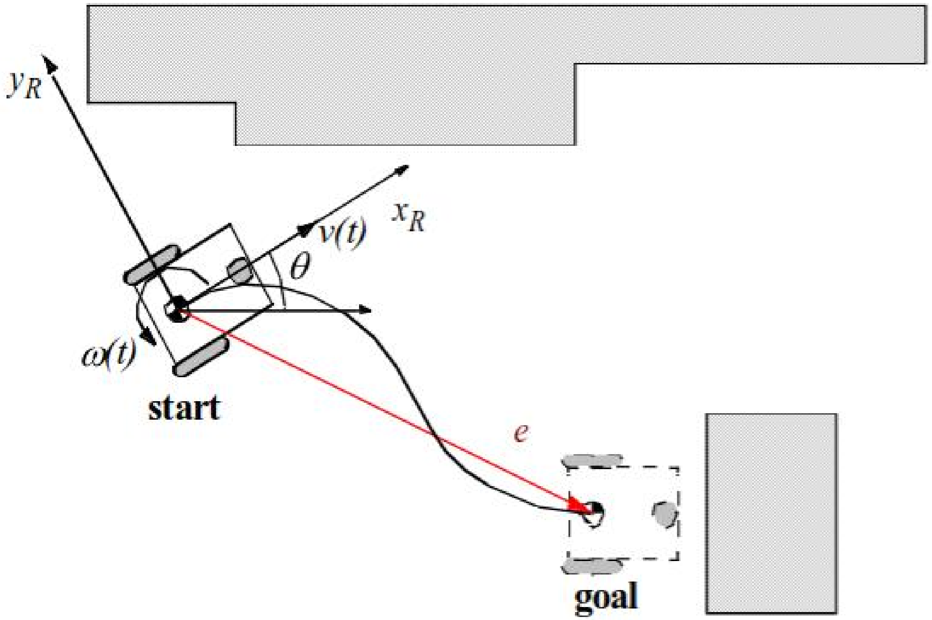
such that the control of

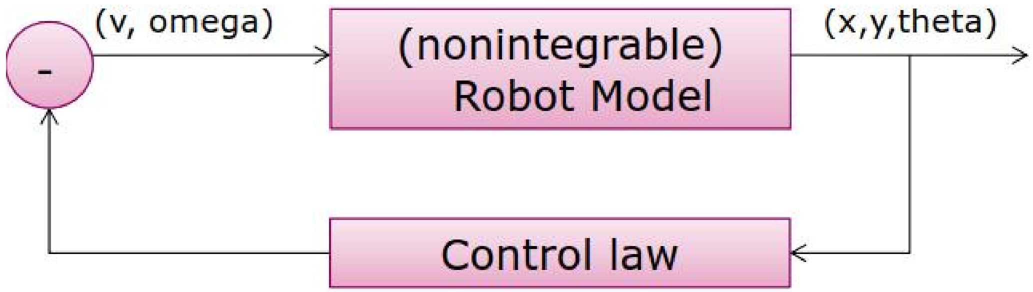
$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \cdot {}^{R}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

drives the error e to zero

$$\lim_{t \to \infty} e(t) = 0$$

MIMO state feedback control





Closed-loop motion control e.g parking for demonstration

- This example for differential drive
- Reparametrization:

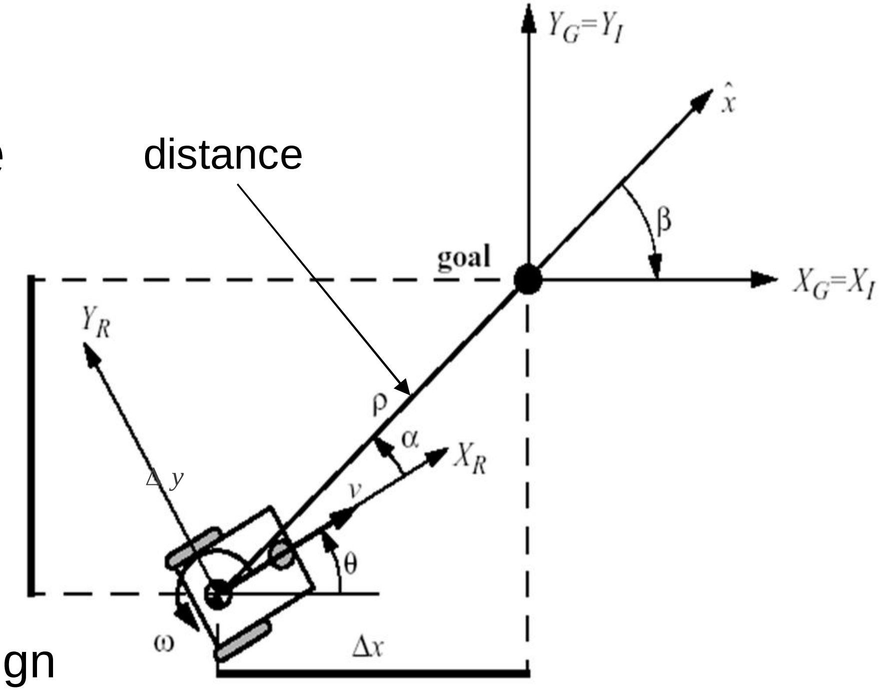
$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = atan2(\Delta y, \Delta x) - \theta$$

$$\beta = -(\theta + \alpha)$$

$$\tan^{-1}\frac{\Delta y}{\Delta x}$$

with correct sign



- We want to make

$$\rho = 0$$
$$\alpha = 0$$
$$\beta = 0$$

Closed-loop control – System

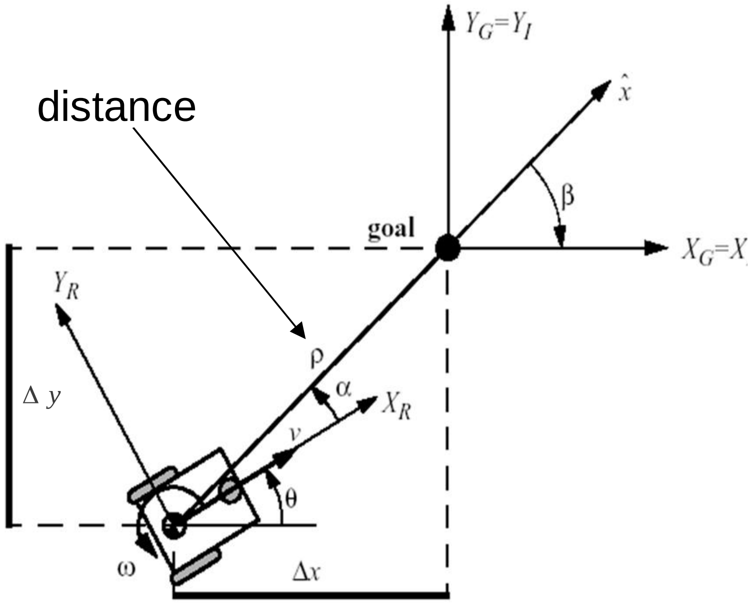
- With reparametrization:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = atan2(\Delta y, \Delta x) - \theta$$

$$\beta = -(\theta + \alpha)$$

$${}^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$



- System in new coordinates

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} \cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ \frac{-\sin\alpha}{\rho} & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \qquad \begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} \cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

$$-\pi/2 < \alpha \le \pi/2 \qquad -\pi < \alpha \le -\pi/2 \lor \pi/2 < \alpha \le \pi$$

Towards goal

Away from goal

Closed-loop control – Control law

- It can be shown that with Control law:

$$v = k_\rho \rho$$

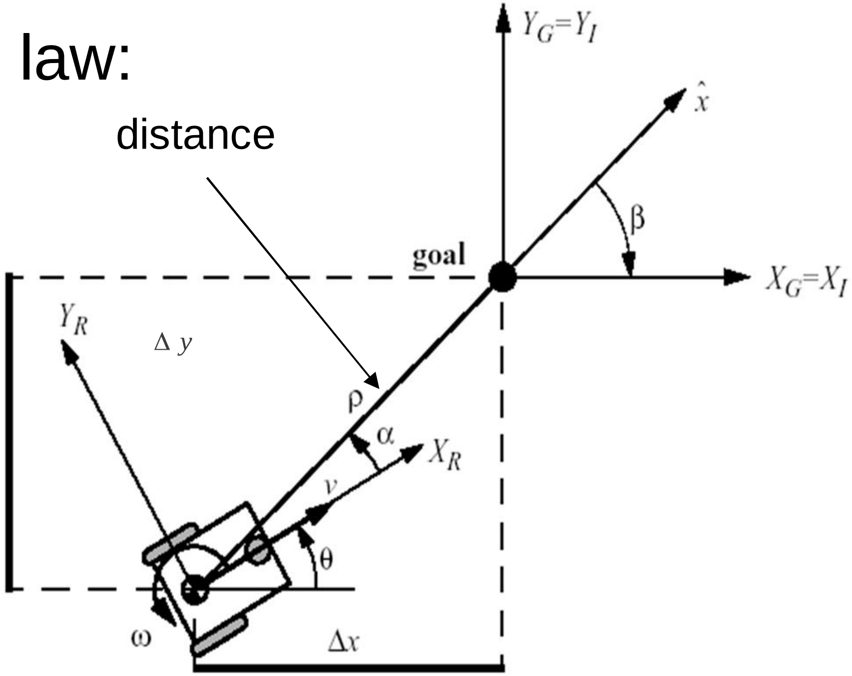
$$\omega = k_\alpha \alpha + k_\beta \beta$$

- We get system with feedback

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin\alpha \end{bmatrix}$$

will be driven to $(\rho, \alpha, \beta) = (0,0,0)$

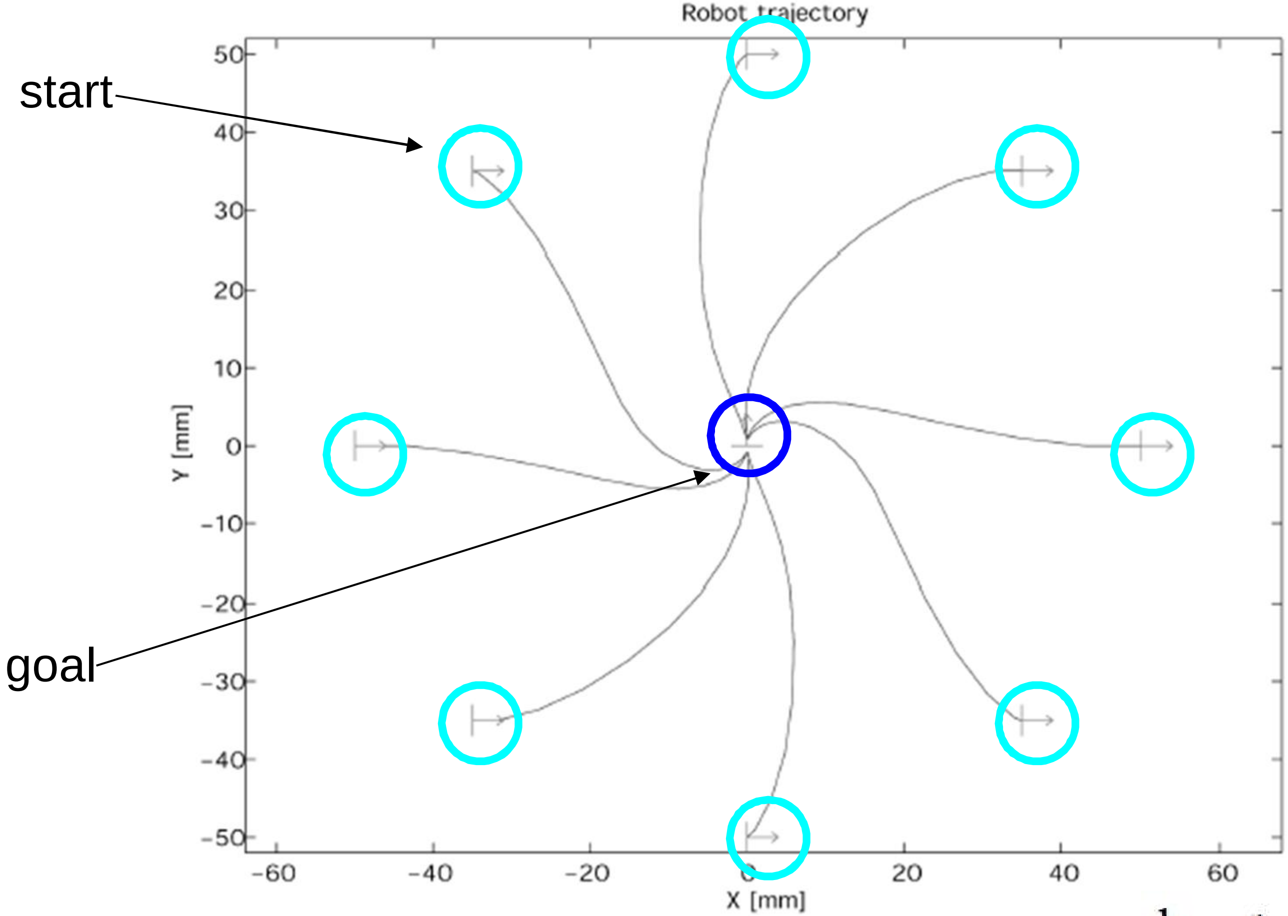
- The control signal v has always constant sign



Locally exponentially stable when

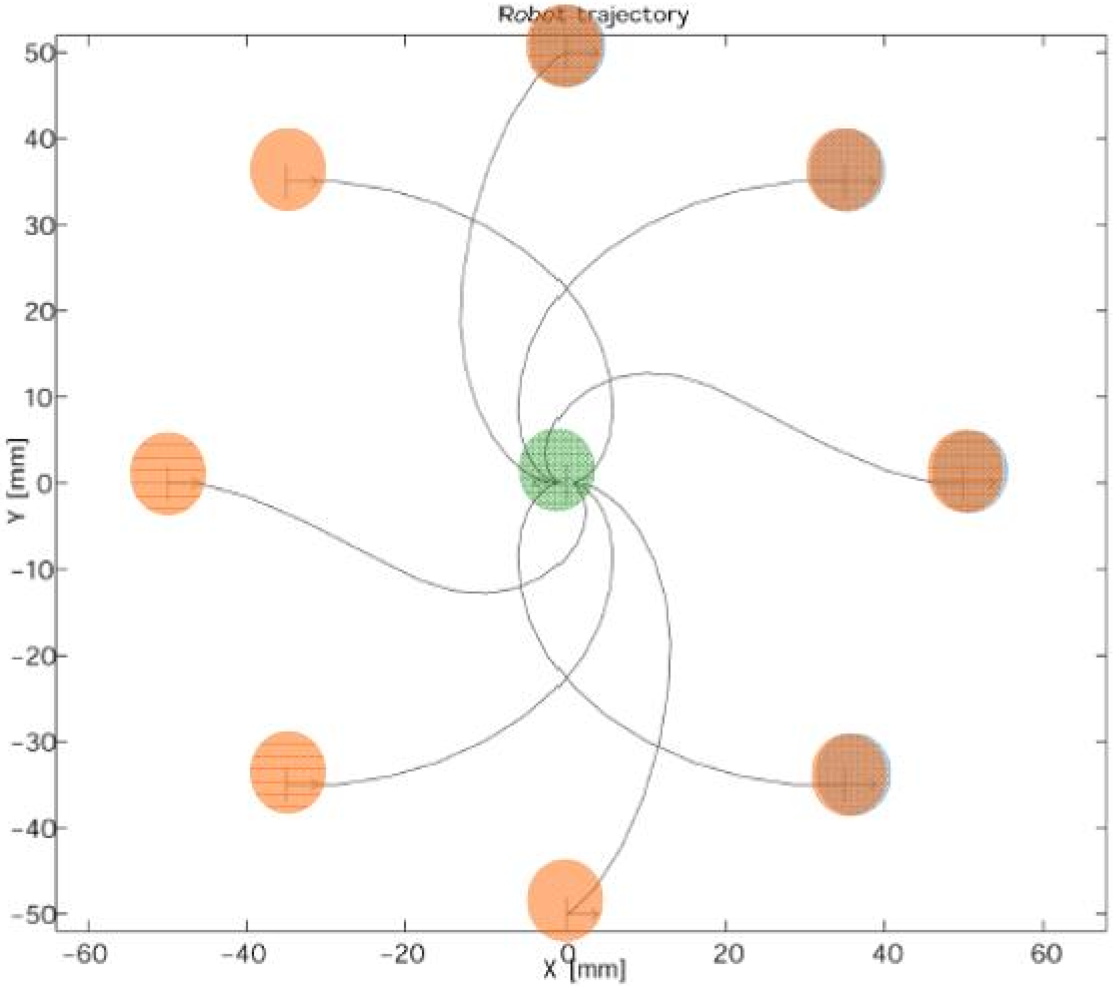
$$k_\rho > 0 \; ; k_\beta < 0 \; ; k_\alpha - k_\beta > 0$$

Closed-loop control – Example β=π/2 !



$$k = (k_\rho, k_\alpha, k_\beta) = (3, 8, -1.5)$$

Closed-loop control – Example β=0



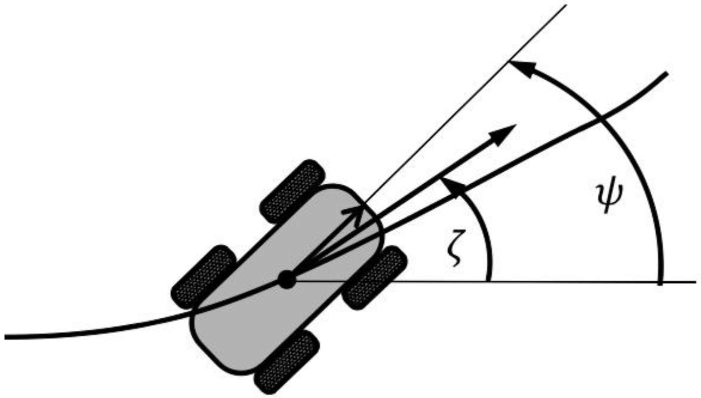
$$k = (k_\rho, k_\alpha, k_\beta) = (3, 8, -1.5)$$

Material

- Siegwart & Nourbash, chapters 2 and 3

For Outdoor, Kinematics for a Wheeled Mobile Roobot (WMR) in 2D plane

- **Heading** ζ is the angle of the path tangent
- **Yaw** ψ is the direction of the forward looking axis of the body frame
- These may be related or unrelated on a given vehicle.
- **Curvature** κ is a property of the path
- **Radius of curvature** R



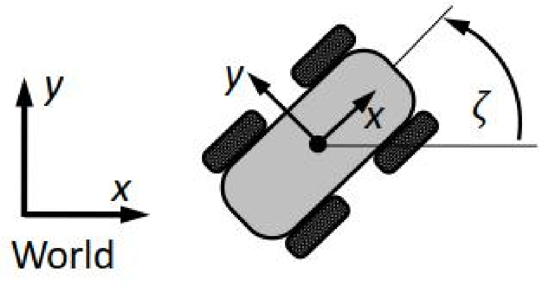
$$\kappa = \frac{d\zeta}{ds}$$

$$R = 1/\kappa$$

Fully Actuated WMR in the Plane

- Velocity is often intrinsically known in the body frame

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} \cos\psi(t) & -\sin\psi(t) & 0 \\ \sin\psi(t) & \cos\psi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(t) \\ v_y(t) \\ \dot{\psi}(t) \end{bmatrix}$$



- The matrix converts coordinates of velocity from body to terrain tangent plane.
- This is the generic 2D velocity kinematics of any vehicle

Fully Actuated WMR in the Plane

- If heading and yaw are the same *(ζ = ψ),* lateral velocity vanishes by definition

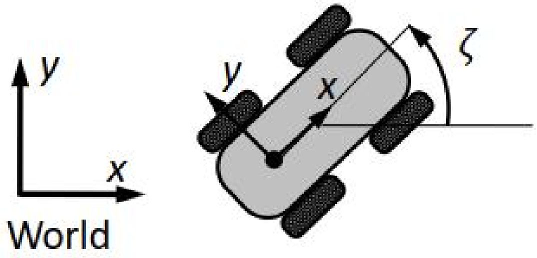
$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \zeta(t) \end{bmatrix} = \begin{bmatrix} \cos\zeta(t) & -\sin\zeta(t) & 0 \\ \sin\zeta(t) & \cos\zeta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(t) \\ 0 \\ \dot{\zeta}(t) \end{bmatrix}$$

- By assumption, the velocity vector is expressed in a frame aligned with the velocity vector
- The lateral velocity is zero by definition of heading

UnderActuated WMR in Plane

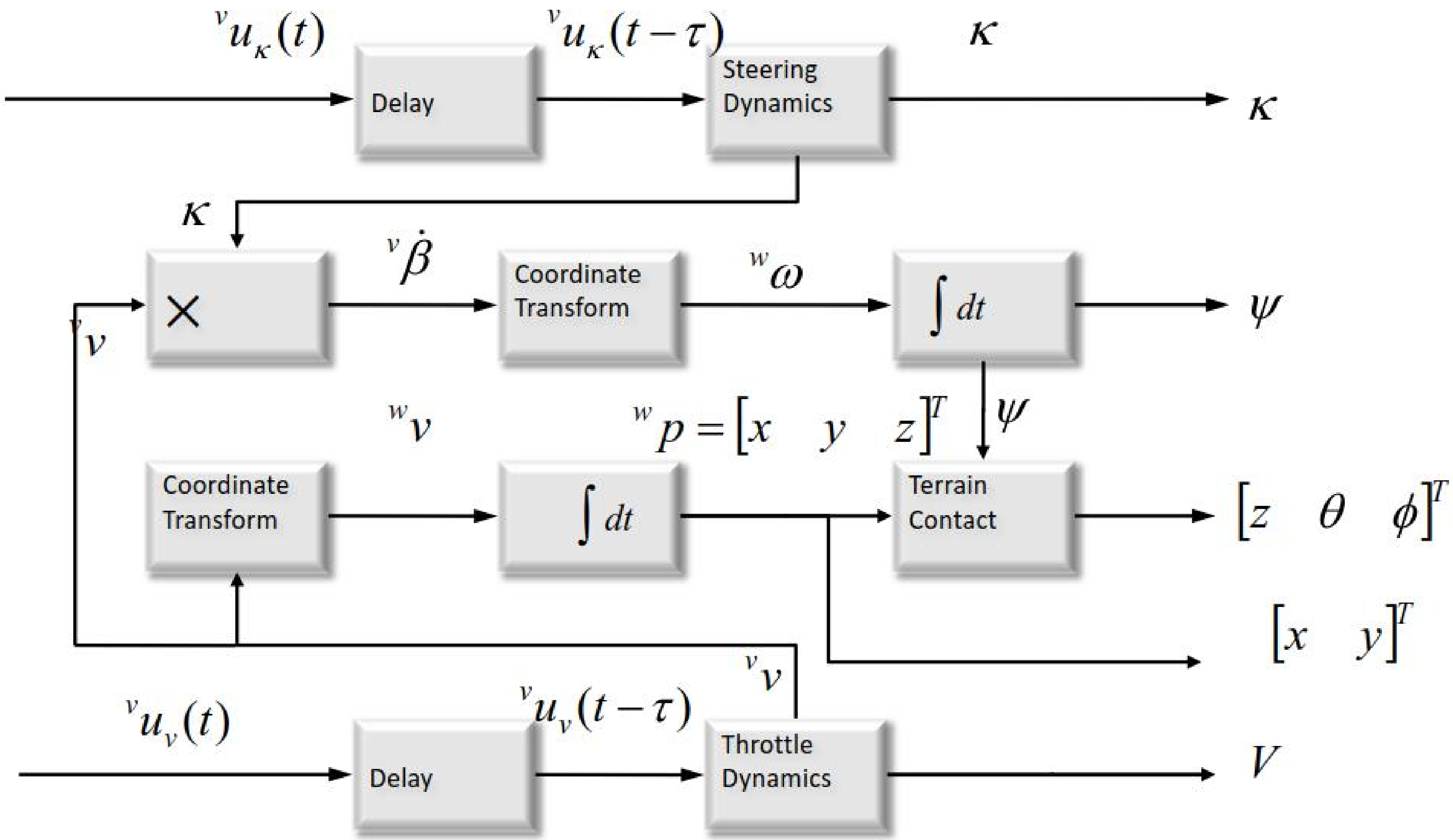
- Many wheeled vehicles are underactuated as a consequence of the fact that lateral motion of wheels is not actuated.
- If the vehicle frame is at center of rear wheels of a car and $\zeta = \psi$. $v_x(t) = v$ $v_y(t) = 0$

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} \cos\psi(t) & 0 \\ \sin\psi(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \kappa(t)v(t) \end{bmatrix}$$



- This is roughly the basic kinematic model used for the tractor in the Agromassi-case.

Fully Actuated WMR in 3D



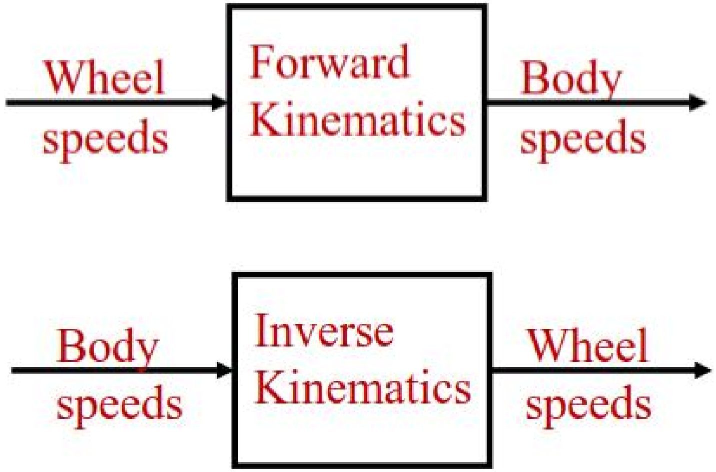
WMR Velocity Kinematics for Fixed Contact Point – General Theory, not required

$$\underline{\dot{x}} = \underline{f}(\underline{x}, \underline{u})$$

$$w(\underline{x})\,\underline{\dot{x}} = 0$$

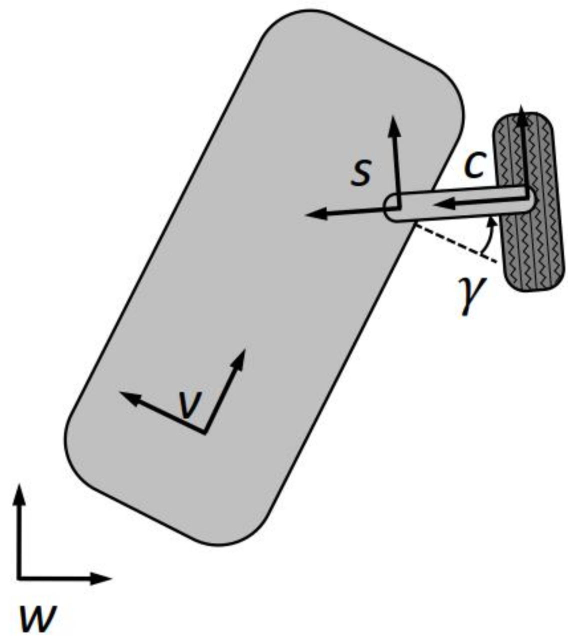
For (WMRs), we care about the underactuated and constrained rate kinematics. It is important for:

- Estimating state in odometry, Kalman filter system models, and more generally in pose estimation of any kind.
- Predicting state in predictive control
- Simulating motion in simulators.



Frame Conventions

- w: world
- v: vehicle
- s: steer
- c: contact point.
- Regard vehicles as rigid bodies (no suspension).
  – Except for steering and wheel rotation.
- Contact point moves on wheel and on floor but it is fixed in wheel frame.



Offset Wheel Equation

- Key assumption: wheel contact point is fixed to wheel. So... $\underline{v}_s^v = \underline{v}_c^s = 0$

$$\underline{v}_c^w = \left[I \mid -[\underline{r}_c^v]^\times \right] \begin{bmatrix} \underline{v}_v^w \\ \underline{\omega}_v^w \end{bmatrix} + \left[I \mid -[\underline{r}_c^s]^\times \right] \begin{bmatrix} \underline{v}_s^v \\ \underline{\omega}_s^v \end{bmatrix} + \underline{v}_c^s \quad (4.30)$$

$$\underline{v}_c^w = \underline{v}_v^w - [\underline{r}_c^v]^\times \underline{\omega}_v^w - [\underline{r}_c^s]^\times \underline{\omega}_s^v \quad (4.39)$$

Offset Wheel Equation

- When s and c frames are coincident

$$\underline{v}_c^w = \underline{v}_v^w - [\underline{r}_c^v]^\times \underline{\omega}_v^w \quad (4.40)$$

Wheel Equation

Multiple Offset Wheels

After long derivation, the result looks like

$$ {}^{v}\underline{v}_c^{w} = H_c^v(\underline{\gamma})\underline{\dot{x}}_v^w + Q_c^s(\underline{\gamma})\underline{\dot{\gamma}} $$

the left pseudoinverse:

$$ \underline{\dot{x}}_v^w = [H_c^v(\underline{\gamma})^T H_c^v(\underline{\gamma})]^{-1} H_c^v(\underline{\gamma})^T [\underline{v}_c^w - Q_c^s(\underline{\gamma})\underline{\dot{\gamma}}] $$

- $\underline{\dot{x}}_v^w$: Robot linear and angular velocity
- $\underline{\gamma}$: Steer Angles
- $\underline{v}_c^w$: Wheel Speeds
- $\underline{\dot{\gamma}}$: Steer Angle Rates

WMR Kinematics

Box 4.2: WMR Forward Kinematics: Offset Wheels

Offset wheel equations for all wheels can be grouped together to produce

$$\underline{v}_c^w = H_c^v(\underline{\gamma})\,\underline{\dot{x}}_v^w + Q_c^s(\underline{\gamma})\,\underline{\dot{\gamma}}$$

where each pair of rows of H_c^v and Q_c^s comes from an offset equation expressed in body coordinates, $\underline{v}_c^w$ is the wheel velocities, $\underline{\dot{x}}_v^w$ is the linear and angular velocity of the vehicle, and $\underline{\gamma}$ is the steer angles.

The inverse mapping (for two or more wheels) can be computed with:

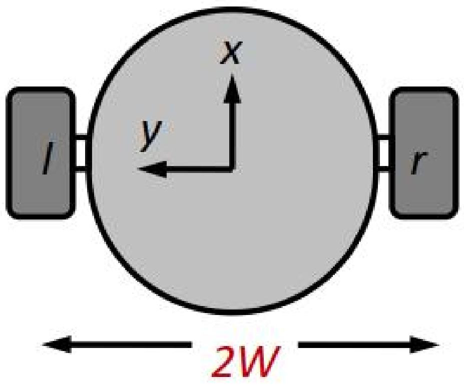
$$\underline{\dot{x}}_v^w = [H_c^v(\underline{\gamma})^T H_c^v(\underline{\gamma})]^{-1} H_c^v(\underline{\gamma})^T [\underline{v}_c^w - Q_c^s(\underline{\gamma})\underline{\dot{\gamma}}]$$

For nonoffset wheels H_c^v simplifies, and Q_c^s disappears.

Example: Differential Steer (Inv)

- Let 'l' and 'r' denote left and right wheel frames.
- The dimensions are:

$$\underline{r}_l^v = \begin{bmatrix} 0 & W \end{bmatrix}^T \qquad \underline{r}_r^v = \begin{bmatrix} 0 & -W \end{bmatrix}^T$$



- In body frame, velocities have only an x component. Equation 4.40 reduces to:

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} v_x + \omega W \\ v_x - \omega W \end{bmatrix} = \begin{bmatrix} 1 & W \\ 1 & -W \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix}$$

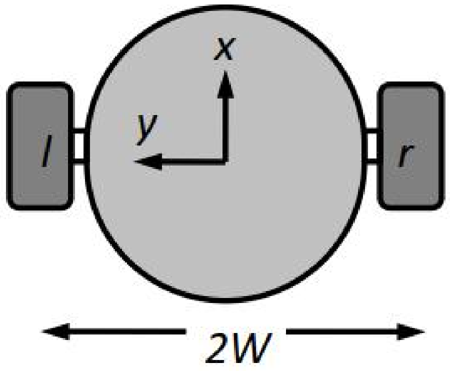
Can solve for 2 dof of 3 dof motion. Other dof is zero in body frame (for this choice of body frame).

- Two equations giving sideways wheel velocities were of the form v_y=0, so these were not written.

Example: Differential Steer (Fwd)

- Inverse kinematics again are:

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} 1 & W \\ 1 & -W \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix}$$



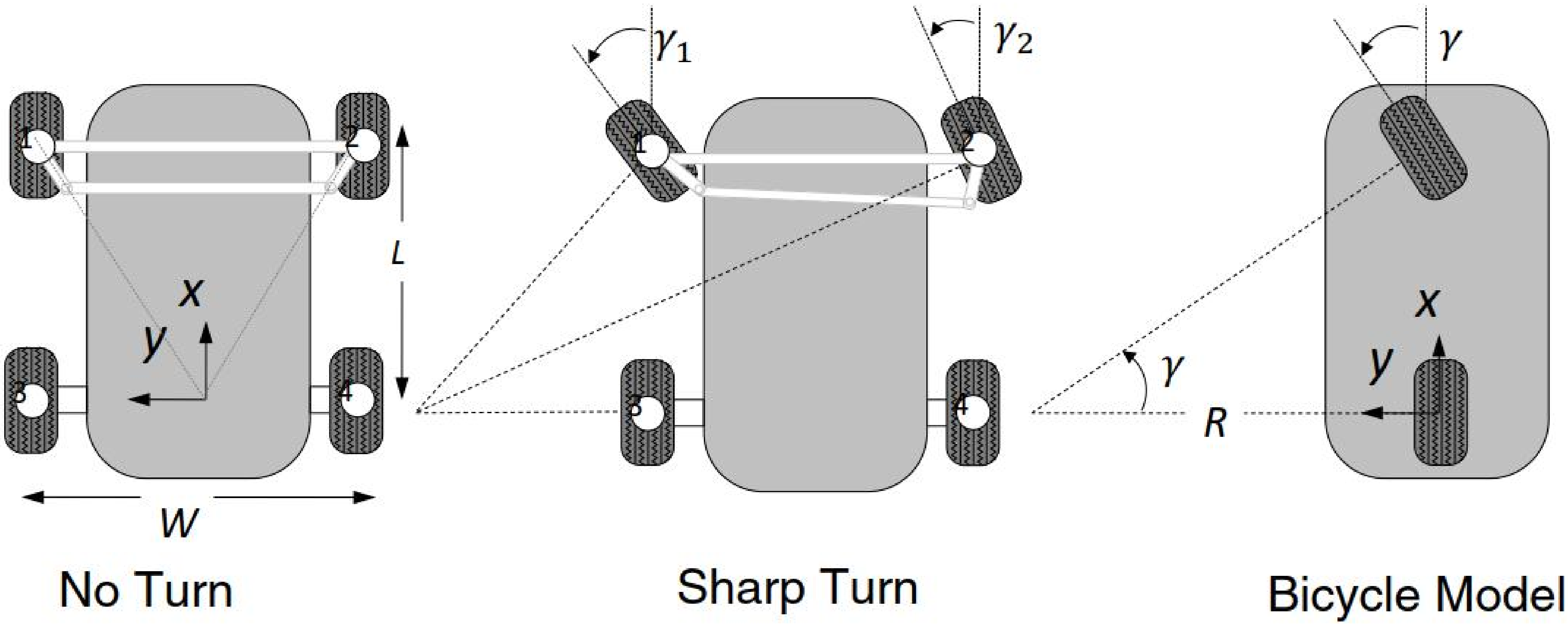
- This is easy to invert:

Again, other dof is zero in body frame due to nonholonomic constraints

$$\begin{bmatrix} v_x \\ \omega \end{bmatrix} = \frac{1}{2W} \begin{bmatrix} W & W \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{W} & -\frac{1}{W} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}$$

Example: Ackerman Steer

- Special mechanism ensures wheels are lined up properly.



Example: Ackerman Steer (Inverse)

- Position vector to front wheel in body (vehicle) frame:

$$\underline{r}_f^{v} = \begin{bmatrix} L & 0 \end{bmatrix}^T$$

- Cross product skew matrix:

$$[\underline{r}_f^{v}]^\times = \begin{bmatrix} 0 & -(\underline{r}_f^{v})_z & (\underline{r}_f^{v})_y \\ (\underline{r}_f^{v})_z & 0 & -(\underline{r}_f^{v})_x \\ -(\underline{r}_f^{v})_y & (\underline{r}_f^{v})_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -L \\ 0 & L & 0 \end{bmatrix}$$

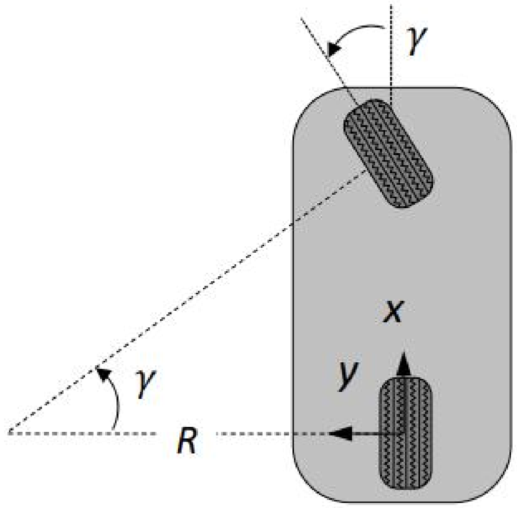
- Wheel equation in body frame reduces to:

$$^{v}\underline{v}_f^{w} = {}^{v}\underline{v}_v^{w} - [\underline{r}_c^{v}]^\times\, {}^{v}\underline{\omega}_v^{w} \Rightarrow v_f = \begin{bmatrix} v_x & \omega L \end{bmatrix}^T$$

BTW rear wheel velocity is trivial

$$\underline{v}_c^{w} = \underline{v}_v^{w} - [\underline{r}_c^{v}]^\times \underline{\omega}_v^{w}$$

Wheel Equation



Bicycle Model

Example: Ackerman Steer (Inverse)

- Last result is of the form:

$$ {}^{v}\underline{v}_f^{w} = {}^{v}H_c \; {}^{w}\underline{\dot{x}}_v $$

Jacobian does not depend on the steer angle

$$ v_f = \begin{bmatrix} v_x & \omega L \end{bmatrix}^T $$

From Last Page

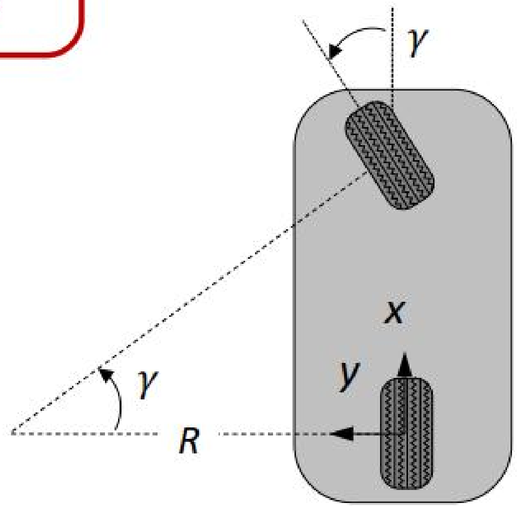
- Written out:

$$ \begin{bmatrix} v_{fx} \\ v_{fy} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} $$

- So, the angle of the front wheel can be computed:

$$ tan(\gamma) = \frac{\omega L}{v_x} = \kappa L = \frac{L}{R} $$

The "Car" Equation



Bicycle Model

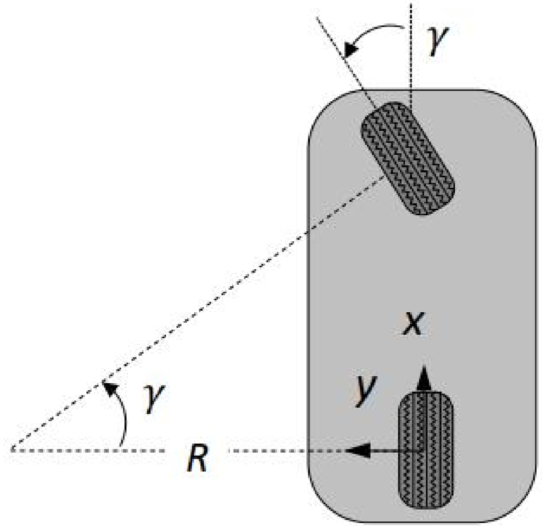
Example: Ackerman Steer (Fwd)

- Inverse kinematics again are:

$$\begin{bmatrix} v_{fx} \\ v_{fy} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix}$$

- This is easy to invert:

$$\begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & L \end{bmatrix}^{-1} \begin{bmatrix} v_{fx} \\ v_{fy} \end{bmatrix} = \begin{bmatrix} v_{fx} \\ v_{fy}/L \end{bmatrix}$$



Bicycle Model

Summary

- The kinematic equations governing the motion of wheeled vehicles are those of planar rigid bodies.
  - Its all about the ICR.
- Rate kinematics for wheeled mobile robots are pretty straightforward (!)
  - In the general case in 3D.
- The inverse problem is often overdetermined.
  - This is solved like any overdetermined system.

As noted in the beginning this general theory is not required.

Out of the scope

- Dynamics important at higher speeds and difficult terrain
- Other types of land locomotion, e.g. tracks
- Aerial and waterborne robots etc.

Material

- Mobile Robotics - Mathematics, Models, and Methods Alonzo Kelly, CMU, CHAPTER 4, Dynamics