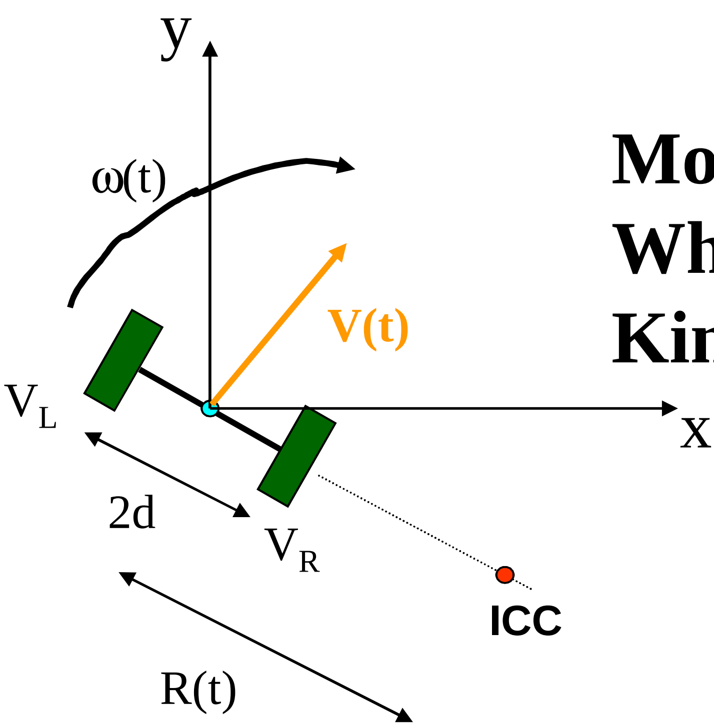


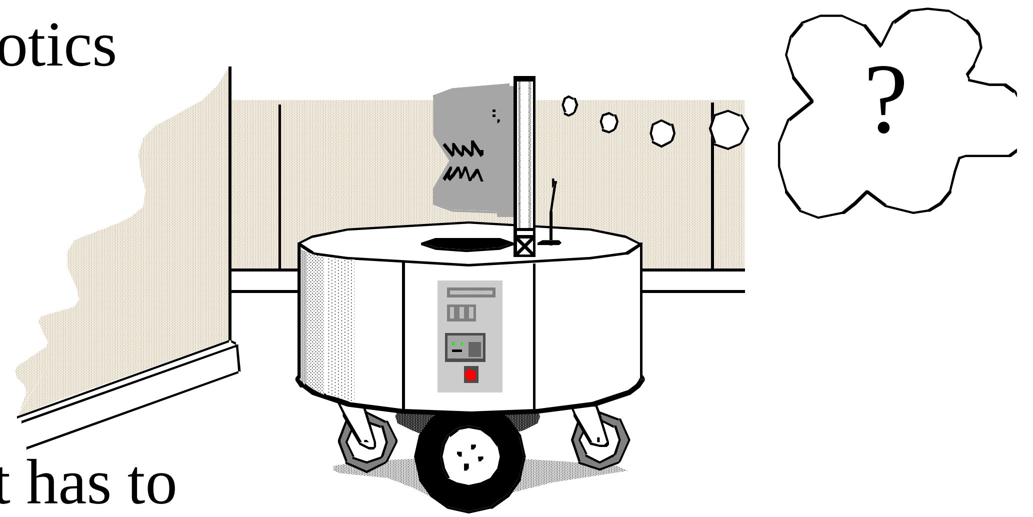
Mobile Robot Wheel Configurations and Kinematics

Martin Jagersand

With slides from Zach Dodds, Roland Siegwart, Sebastian Thrun

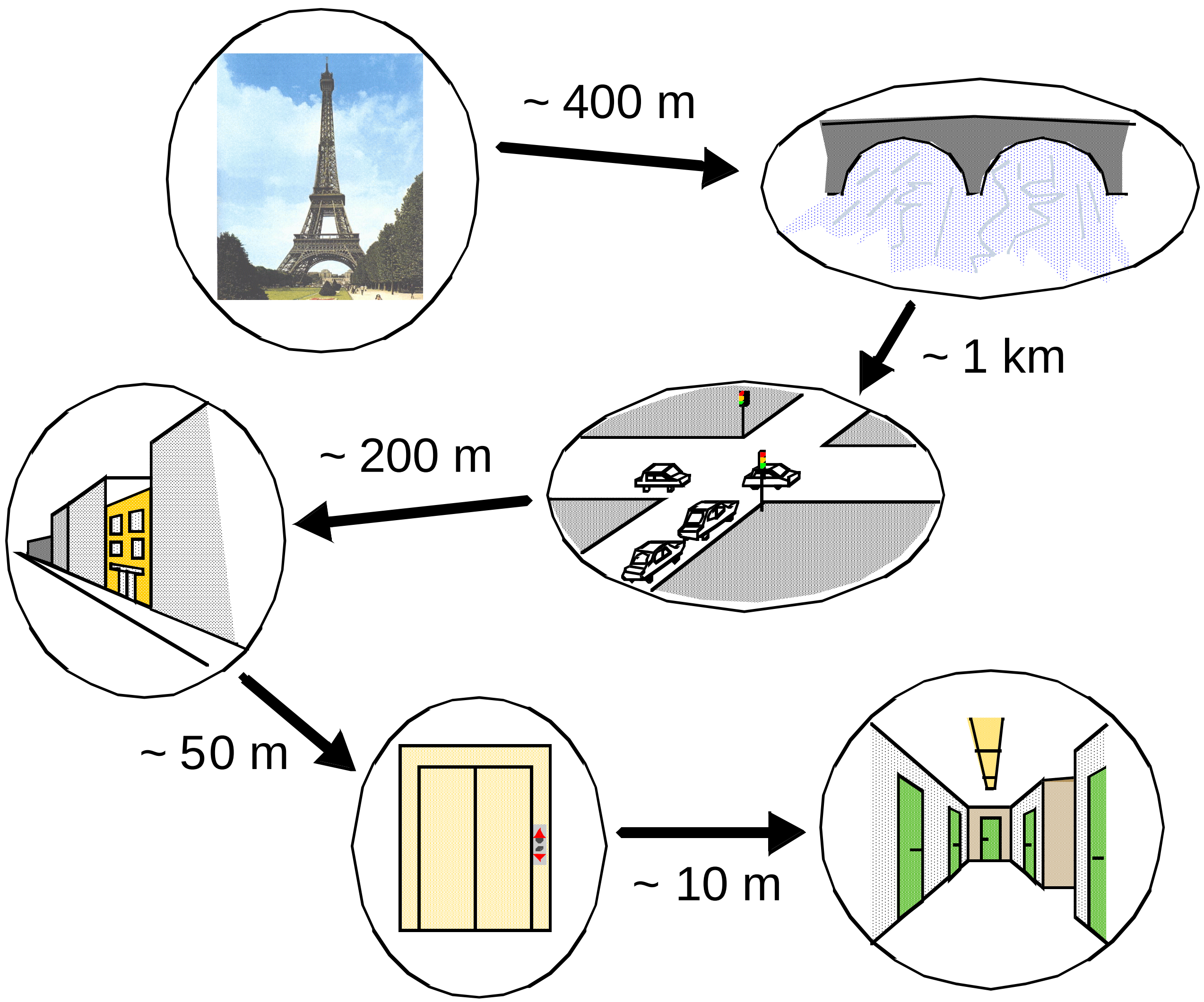
Autonomous Mobile Robots

- Three key questions in Mobile Robotics
  - *Where am I ?*
  - *Where am I going ?*
  - *How do I get there ?*



- To answer these questions the robot has to
  - *have a model of the robot and environment (given or autonomously built)*
  - *perceive and analyze the environment*
  - *find robot position within the environment*
  - *plan and execute the movement*
- Today: Focus on Where am I going ? using wheel encoders + model

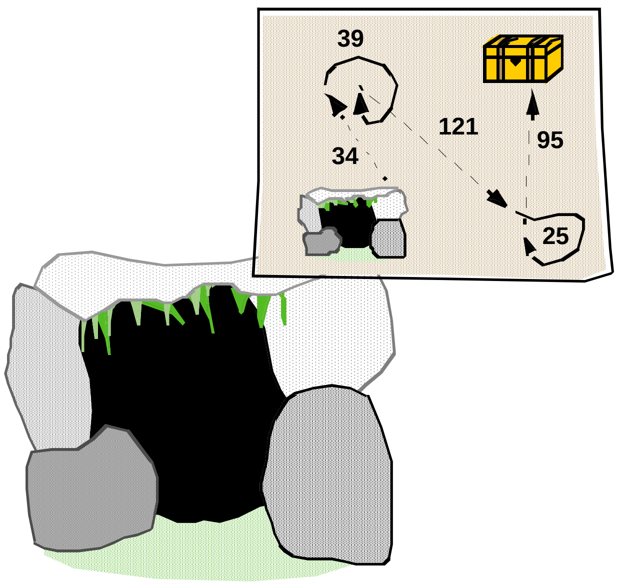
Human Navigation: Topological with imprecise metric information



Courtesy K. Arras

Human Navigation: Topological with imprecise metric information

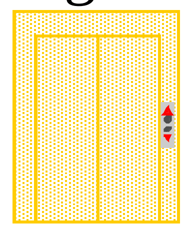
- Odometry



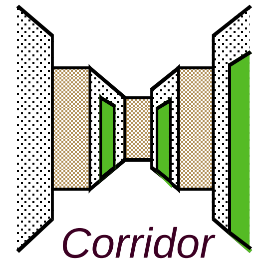
How to find a treasure

(Imprecise)

- Feature-based Navigation

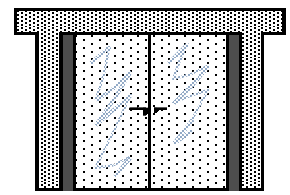


Elevator door



Corridor crossing

Eiffel Tower



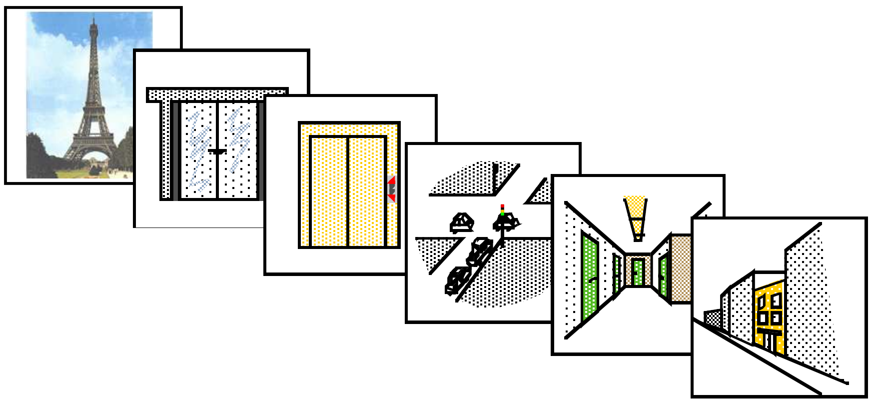
Entrance

Courtesy K. Arras

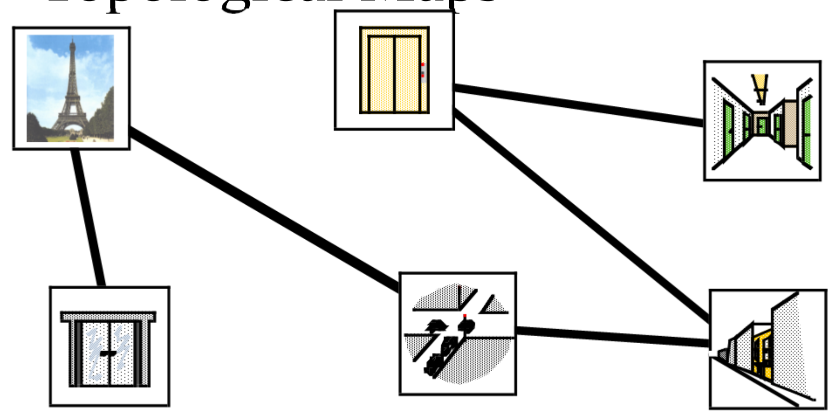
- *still a challenge for artificial systems*

Environment Representation: The Map Categories

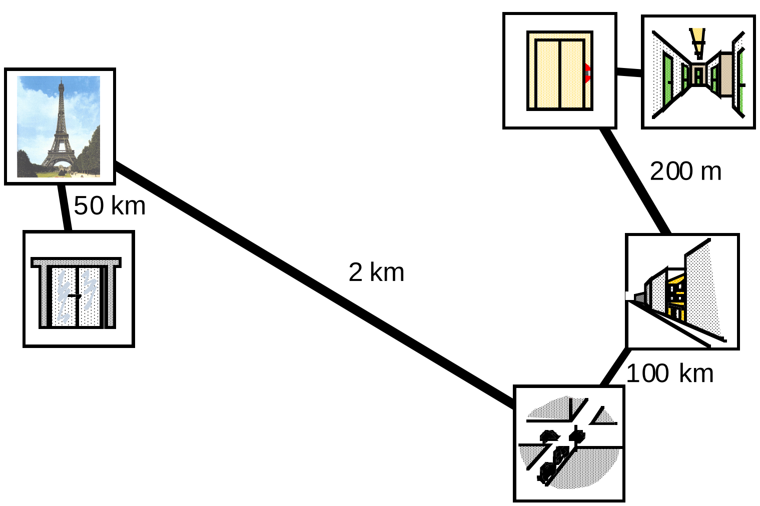
- Recognizable Locations



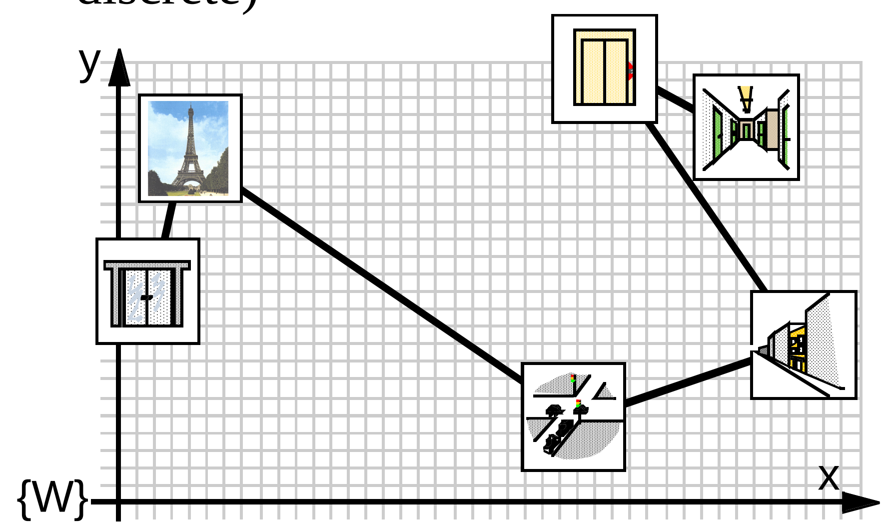
- Topological Maps



- Metric Topological Maps



- Fully Metric Maps (continuos or discrete)

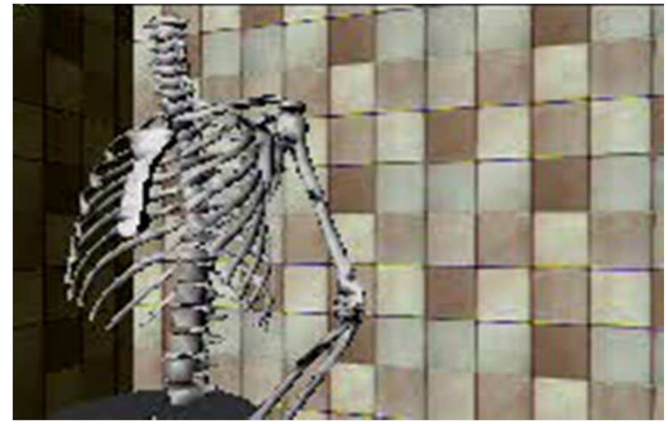


Courtesy K. Arras

Kinematics

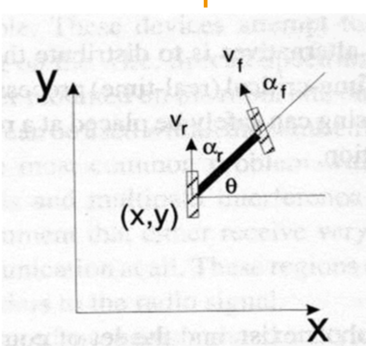
If we move a wheel one radian, how does the robot center move?

Robot geometry and kinematic analysis gives answers!

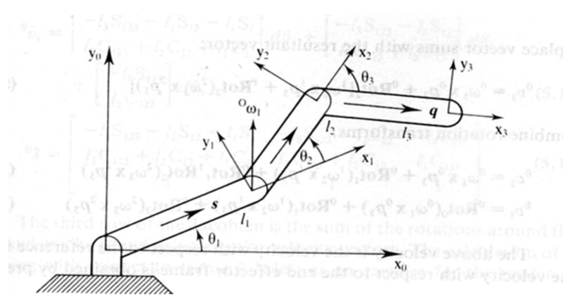


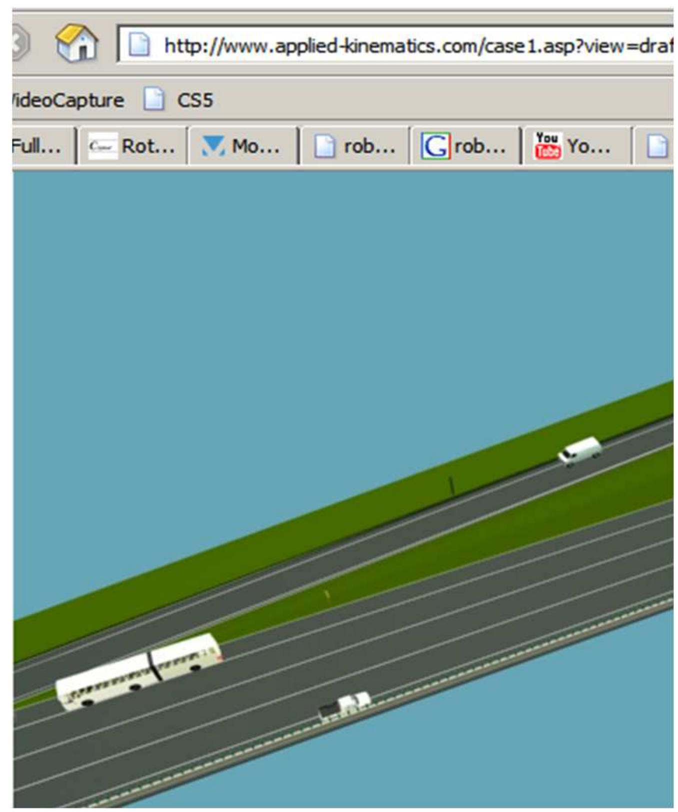
from sort of simple to
sort of complex

wheeled platforms



manipulator modeling





applied-kinematics.com: forensic animation

Kinematics vs. Dynamics

kinematics

The effect of a robot's geometry on its motion.

If the motors move ***this*** much, where will the robot be?

Assumes that we control ***encoder readings***…

dynamics

The effect of all forces (internal and external) on a robot's motion.

If the motors apply ***this*** much force, where will the robot be?

Assumes that we control ***motor current***…

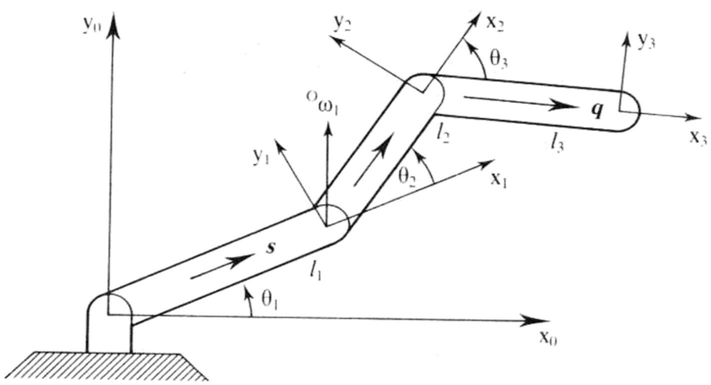
consider the segway…

Kinematics

kinematics

The effect of a robot's geometry on its motion.

from sort of simple to sort of complex



three-link manipulator

represented by a 4x4 matrix

dynamics

The effect of all forces (internal and external) on a robot's motion.

Aaargh!

two-link manipulator

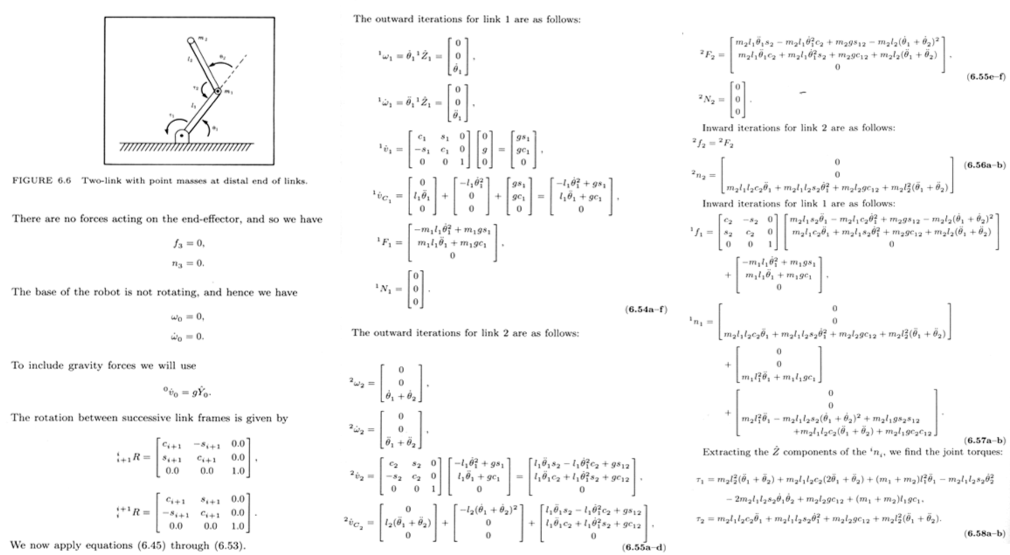


FIGURE 6.6 Two-link with point masses at distal end of links.

There are no forces acting on the end-effector, and so we have

$$f_3 = 0,$$
$$n_3 = 0.$$

The base of the robot is not rotating, and hence we have

$$\omega_0 = 0,$$
$$\dot\omega_0 = 0.$$

To include gravity forces we will use

$$^0\dot v_0 = g\hat Y_0.$$

The rotation between successive link frames is given by

$$^{i}_{i+1}R = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0.0 \\ s_{i+1} & c_{i+1} & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix},$$

$$^{i+1}_{i}R = \begin{bmatrix} c_{i+1} & s_{i+1} & 0.0 \\ -s_{i+1} & c_{i+1} & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}.$$

We now apply equations (6.45) through (6.53).

The outward iterations for link 1 are as follows:

$$^1\omega_1 = \dot\theta_1 {}^1\hat Z_1 = \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix},$$

$$^1\dot\omega_1 = \ddot\theta_1 {}^1\hat Z_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot\theta_1 \end{bmatrix},$$

$$^1\dot v_1 = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix},$$

$$^1\dot v_{C_1} = \begin{bmatrix} 0 \\ l_1\ddot\theta_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -l_1\dot\theta_1^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1\dot\theta_1^2 + gs_1 \\ l_1\ddot\theta_1 + gc_1 \\ 0 \end{bmatrix},$$

$$^1F_1 = \begin{bmatrix} -m_1 l_1\dot\theta_1^2 + m_1 gs_1 \\ m_1 l_1\ddot\theta_1 + m_1 gc_1 \\ 0 \end{bmatrix},$$

$$^1N_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \qquad \textbf{(6.54a–f)}$$

The outward iterations for link 2 are as follows:

$$^2\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 + \dot\theta_2 \end{bmatrix},$$

$$^2\dot\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \ddot\theta_1 + \ddot\theta_2 \end{bmatrix},$$

$$^2\dot v_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -l_1\dot\theta_1^2 + gs_1 \\ l_1\ddot\theta_1 + gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1\ddot\theta_1 s_2 - l_1\dot\theta_1^2 c_2 + gs_{12} \\ l_1\ddot\theta_1 c_2 + l_1\dot\theta_1^2 s_2 + gc_{12} \\ 0 \end{bmatrix},$$

$$^2\dot v_{C_2} = \begin{bmatrix} 0 \\ l_2(\ddot\theta_1 + \ddot\theta_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -l_2(\dot\theta_1 + \dot\theta_2)^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1\ddot\theta_1 s_2 - l_1\dot\theta_1^2 c_2 + gs_{12} \\ l_1\ddot\theta_1 c_2 + l_1\dot\theta_1^2 s_2 + gc_{12} \\ 0 \end{bmatrix}. \qquad \textbf{(6.55a–d)}$$

$$^2F_2 = \begin{bmatrix} m_2 l_1\ddot\theta_1 s_2 - m_2 l_1\dot\theta_1^2 c_2 + m_2 gs_{12} - m_2 l_2(\dot\theta_1 + \dot\theta_2)^2 \\ m_2 l_1\ddot\theta_1 c_2 + m_2 l_1\dot\theta_1^2 s_2 + m_2 gc_{12} + m_2 l_2(\ddot\theta_1 + \ddot\theta_2) \\ 0 \end{bmatrix},$$

$$^2N_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \qquad \textbf{(6.55e–f)}$$

Inward iterations for link 2 are as follows:

$$^2f_2 = {}^2F_2$$

$$^2n_2 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2\ddot\theta_1 + m_2 l_1 l_2 s_2\dot\theta_1^2 + m_2 l_2 gc_{12} + m_2 l_2^2(\ddot\theta_1 + \ddot\theta_2) \end{bmatrix} \qquad \textbf{(6.56a–b)}$$

Inward iterations for link 1 are as follows:

$$^1f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_2 l_1 s_2\ddot\theta_1 - m_2 l_1 c_2\dot\theta_1^2 + m_2 gs_{12} - m_2 l_2(\dot\theta_1 + \dot\theta_2)^2 \\ m_2 l_1 c_2\ddot\theta_1 + m_2 l_1 s_2\dot\theta_1^2 + m_2 gc_{12} + m_2 l_2(\ddot\theta_1 + \ddot\theta_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -m_1 l_1\dot\theta_1^2 + m_1 gs_1 \\ m_1 l_1\ddot\theta_1 + m_1 gc_1 \\ 0 \end{bmatrix},$$

$$^1n_1 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2\ddot\theta_1 + m_2 l_1 l_2 s_2\dot\theta_1^2 + m_2 l_2 gc_{12} + m_2 l_2^2(\ddot\theta_1 + \ddot\theta_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_1 l_1^2\ddot\theta_1 + m_1 l_1 gc_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_2 l_1^2\ddot\theta_1 - m_2 l_1 l_2 s_2(\dot\theta_1 + \dot\theta_2)^2 + m_2 l_1 gs_2 s_{12} + m_2 l_1 l_2 c_2(\ddot\theta_1 + \ddot\theta_2) + m_2 l_1 gc_2 c_{12} \end{bmatrix}. \qquad \textbf{(6.57a–b)}$$

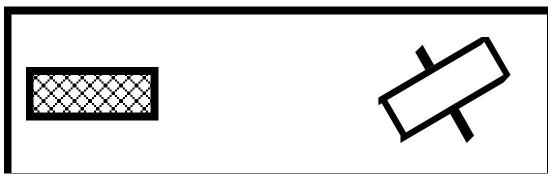
Extracting the $\hat Z$ components of the ^in_i, we find the joint torques:

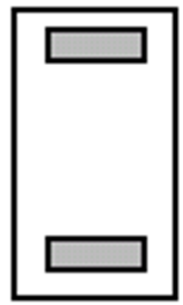
$$\tau_1 = m_2 l_2^2(\ddot\theta_1 + \ddot\theta_2) + m_2 l_1 l_2 c_2(2\ddot\theta_1 + \ddot\theta_2) + (m_1 + m_2) l_1^2\ddot\theta_1 - m_2 l_1 l_2 s_2\dot\theta_2^2 - 2m_2 l_1 l_2 s_2\dot\theta_1\dot\theta_2 + m_2 l_2 gc_{12} + (m_1 + m_2) l_1 gc_1,$$

$$\tau_2 = m_2 l_1 l_2 c_2\ddot\theta_1 + m_2 l_1 l_2 s_2\dot\theta_1^2 + m_2 l_2 gc_{12} + m_2 l_2^2(\ddot\theta_1 + \ddot\theta_2). \qquad \textbf{(6.58a–b)}$$

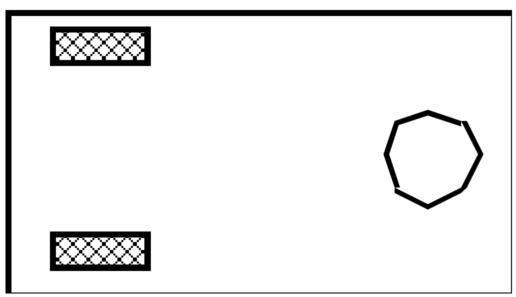
Mobile Robot
Different Arrangements of Wheels

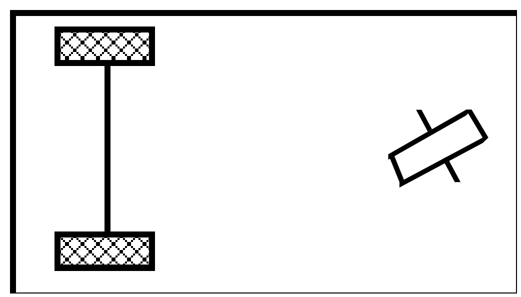
- Two wheels



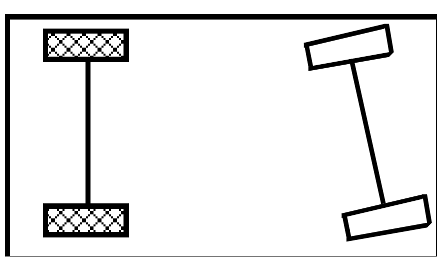


- Three wheels

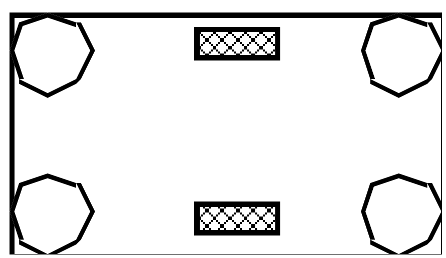




- Four wheels



Six wheels

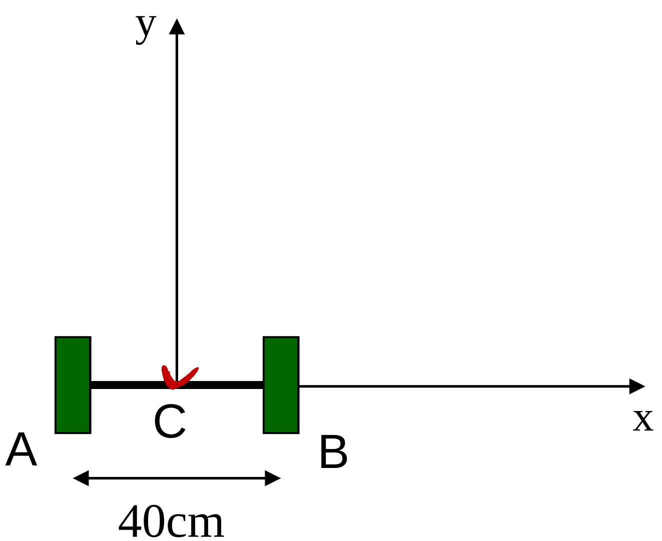


“Quiz” Odometry / Dead Reckoning

or why optical encoders are so useful...

Suppose a robots’ wheels are arranged as shown below. The robot moves its wheels smoothly and evenly forward (initially +y) until Wheel **A** has made 8 full rotations and Wheel **B** has made 6 full rotations. Where is the robot? That is, what are the coordinates of its center, **C**, if **C** was initially at (0,0)?

Assume that the wheels do not slip as they move. (A strong assumption...)



Where will C be?

Wheels **A** and **B** have a diameter of 5 cm.

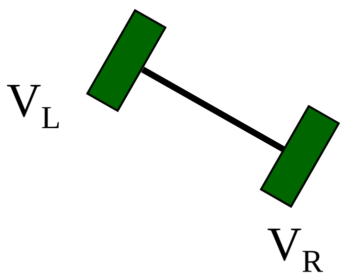
Differential drive

Most common kinematic choice

Most miniature robots…

ER1, Pioneer, Roomba

- difference in wheels' speeds determines its turning angle



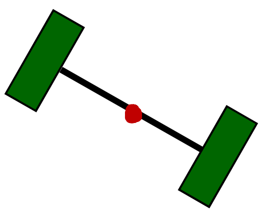
Differential drive

Most common kinematic choice

Most miniature robots…

ER1, Pioneer, Roomba

- difference in wheels' speeds determines its turning angle



Questions (forward kinematics)

Given the **wheel's** velocities or positions, what is the **robot's** velocity/position ?

Are there any inherent system constraints?

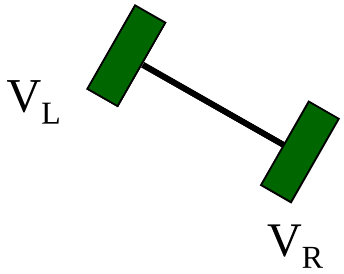
Differential drive

Most common kinematic choice

Most miniature robots…

ER1, Pioneer, Roomba

- difference in wheels' speeds determines its turning angle



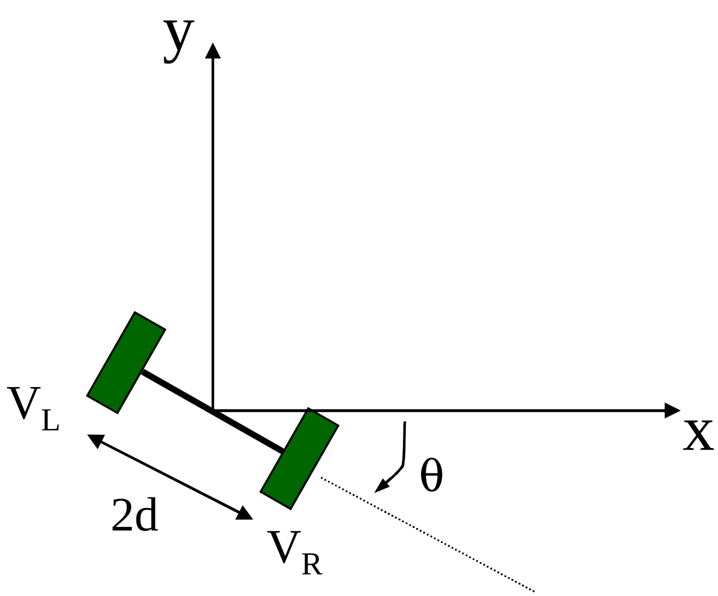
Questions (forward kinematics)

Given the wheel's velocities or positions, what is the robot's velocity/position ?

Are there any inherent system constraints?

1) Specify system measurements
2) Determine the point (the radius) around which the robot is turning.
3) Determine the speed at which the robot is turning to obtain the robot velocity.
4) Integrate to find position.

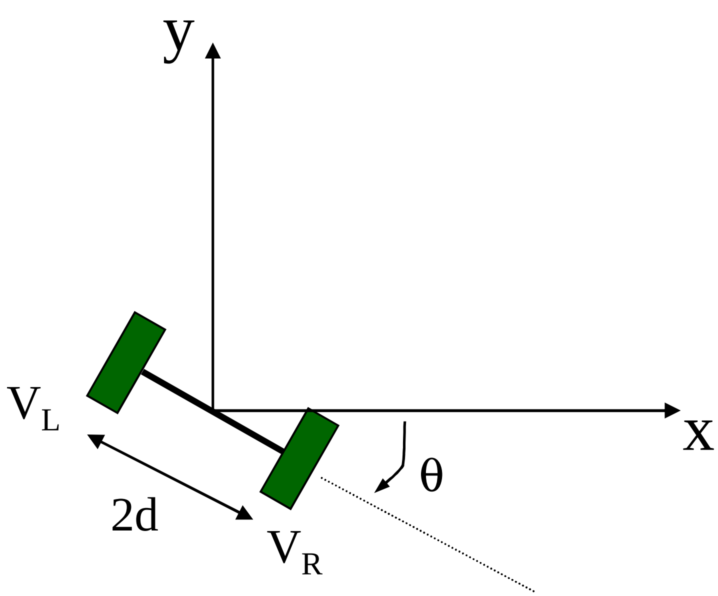
Differential drive



1) Specify system measurements

- consider possible coordinate systems

Differential drive

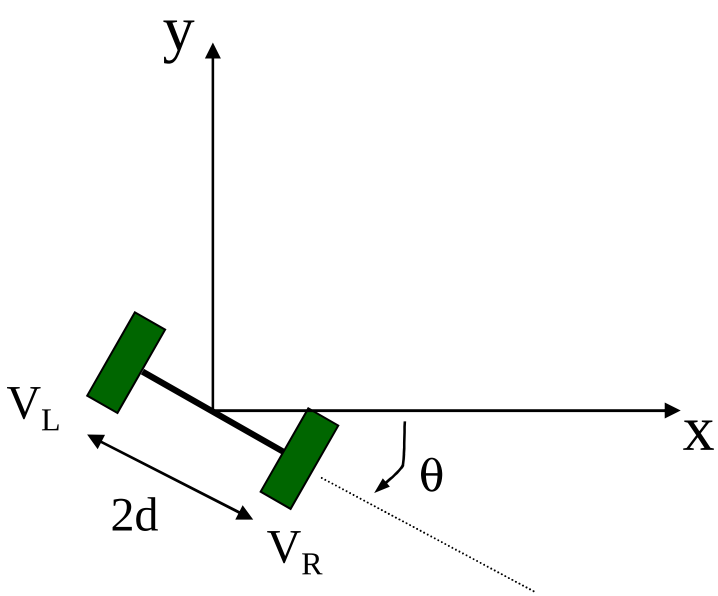


1) Specify system measurements
   - consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning.

Is there always a point around which the robot is rotating?

Differential drive



1) Specify system measurements
   - consider possible coordinate systems

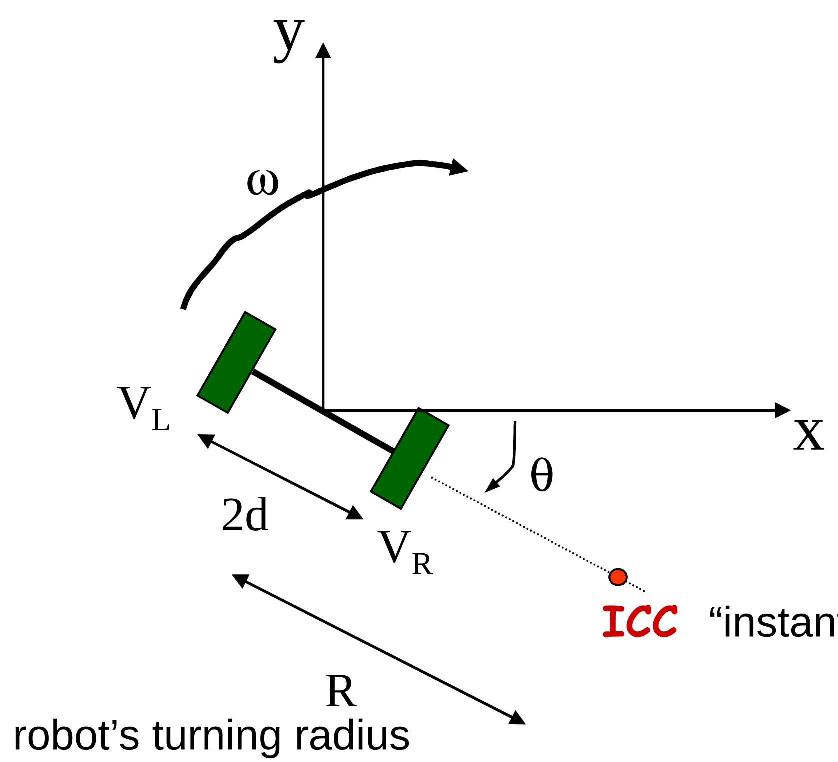
2) Determine the point (the radius) around which the robot is turning.

- to minimize wheel slippage, the instantaneous center of curvature (the **ICC**) must lie at the intersection of the wheels' axles

- each wheel must be traveling at the same angular velocity

same angular velocity??

Differential drive



1) Specify system measurements
- consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning.

- to minimize wheel slippage, this point (the **ICC**) must lie at the intersection of the wheels' axles

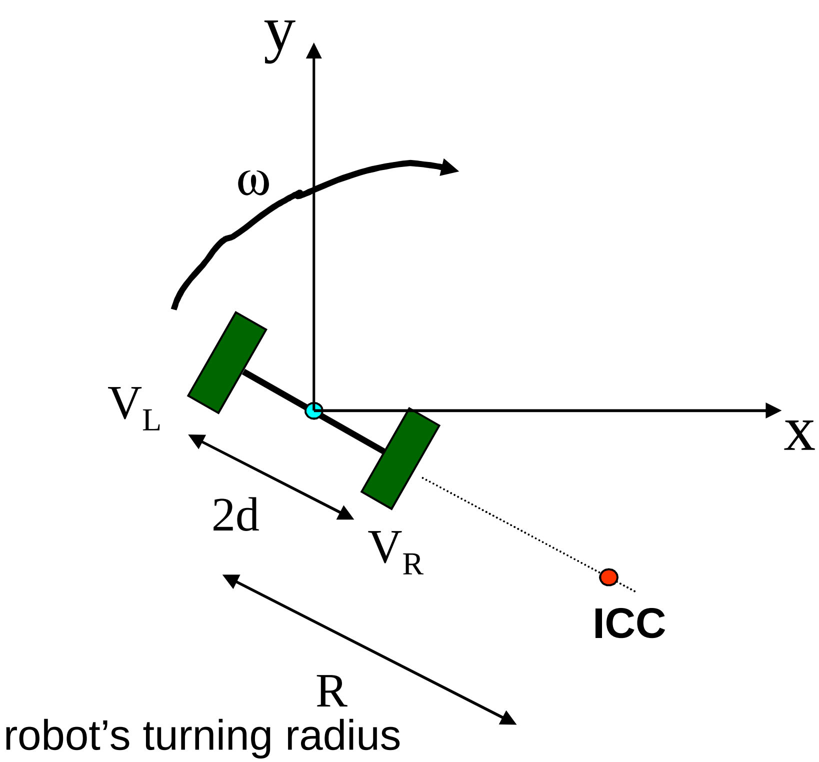
- each wheel must be traveling at the same angular velocity **around the ICC**

ICC "instantaneous center of curvature"

How are these values related?

(the wheel diameter is already accounted for in V_L and V_R)

Differential drive



1) Specify system measurements
   - consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning.

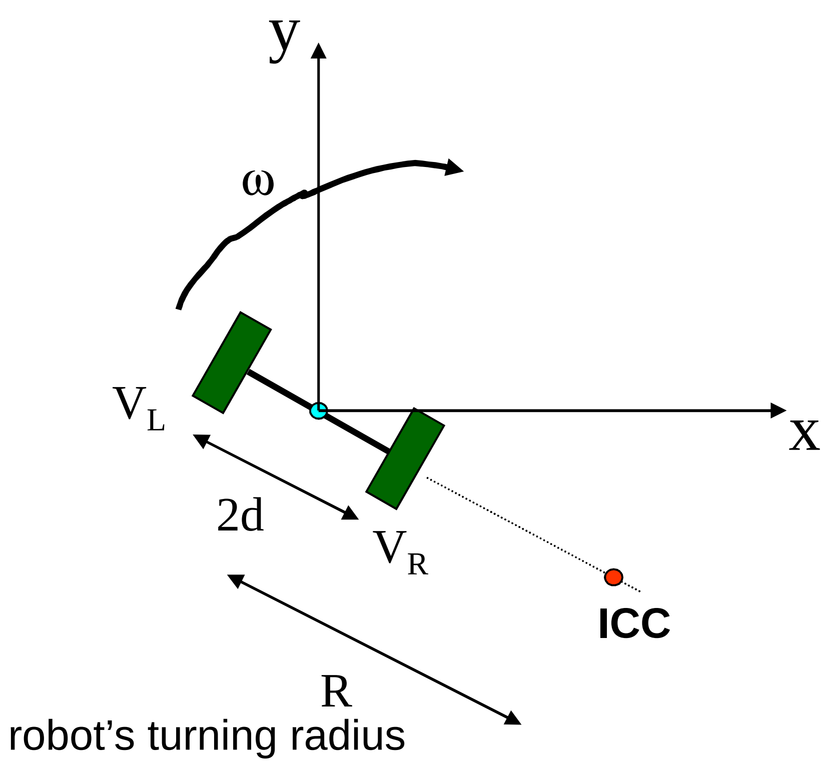
   - each wheel must be traveling at the same angular velocity **around the ICC**

3) Determine the robot's speed around the ICC and its linear velocity

$$\omega(R+d) = V_L$$

$$\omega(R-d) = V_R$$

Differential drive



1) Specify system measurements
   - consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning.

   - each wheel must be traveling at the same angular velocity **around the ICC**

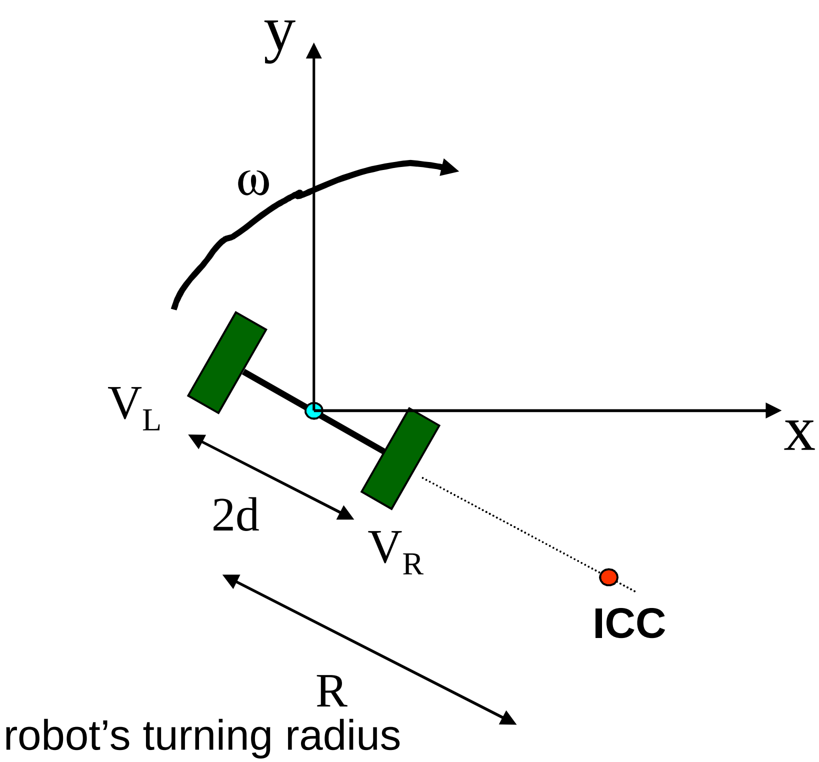
3) Determine the robot's speed around the ICC and its linear velocity

$$\omega(R+d) = V_L$$

$$\omega(R-d) = V_R$$

of these five, what's known & what's not?

Differential drive



are there interesting cases?

1) Specify system measurements
  - consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning.

- each wheel must be traveling at the same angular velocity **around the ICC**

3) Determine the robot's speed around the ICC and its linear velocity

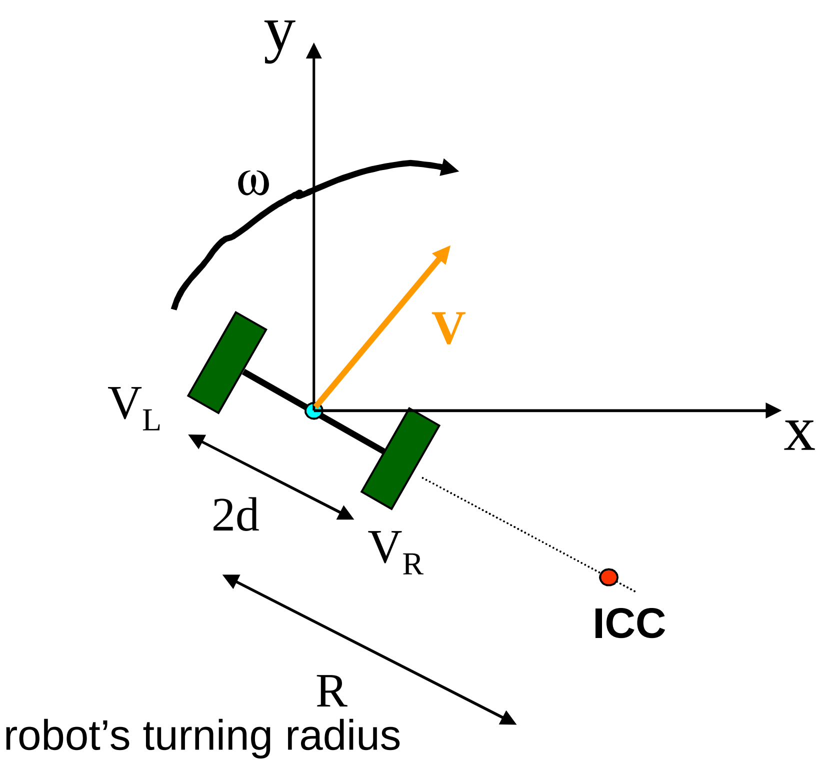
$$\omega(R+d) = V_L$$
$$\omega(R-d) = V_R$$

Thus,

$$\omega = (V_R - V_L) / 2d$$

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

Differential drive



1) Specify system measurements
- consider possible coordinate systems

---

2) Determine the point (the radius) around which the robot is turning.

- each wheel must be traveling at the same angular velocity **around the ICC**

---

3) Determine the robot's speed around the ICC and its linear velocity

$$\omega(R+d) = V_L$$
$$\omega(R-d) = V_R$$

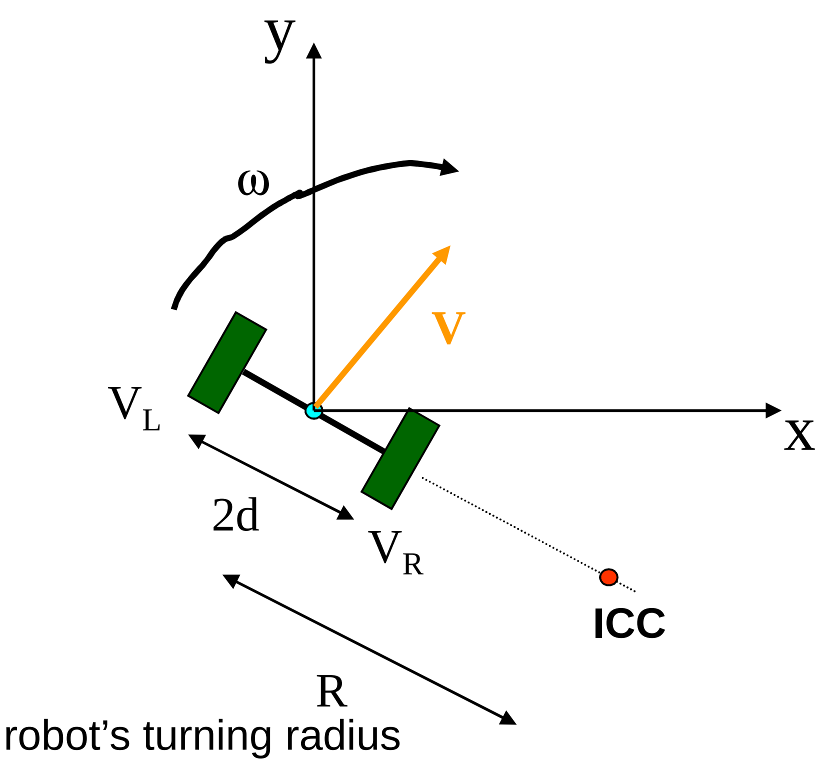
Thus,

$$\omega = (V_R - V_L) / 2d$$

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

So, what is the robot's velocity?

Differential drive



1) Specify system measurements
   - consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning.

   - each wheel must be traveling at the same angular velocity **around the ICC**

3) Determine the robot's speed around the ICC and its linear velocity

$$\omega(R+d) = V_L$$
$$\omega(R-d) = V_R$$

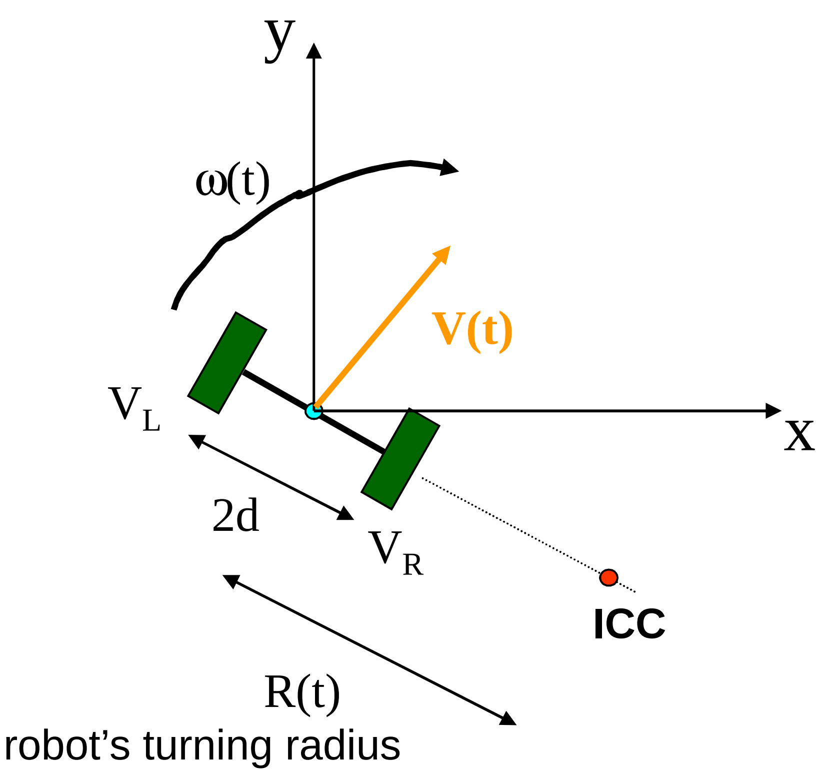
Thus, $\omega = (V_R - V_L) / 2d$

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

So, where's the robot?

So, the robot's velocity is $V = \omega R = (V_R + V_L) / 2$

Differential drive



4) Integrate to obtain position

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

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

Thus,

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

Kinematics

with

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

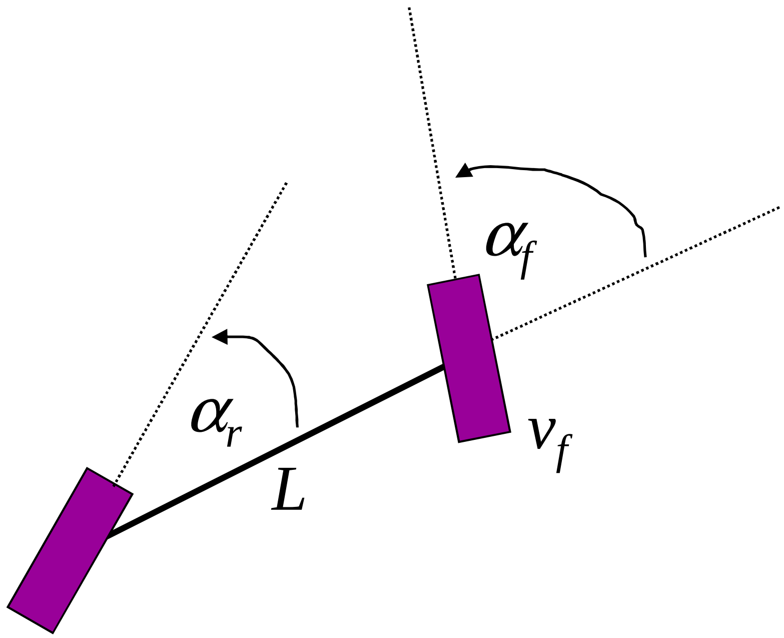
What has to happen to change the ICC ?

things have to change over time, **t**

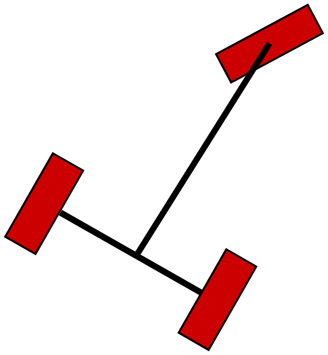
Q: Two-steered-wheel bicycle

Unusual kinematics:

- *powered front wheel (velocity = v_f)*
- *both wheels (in orientation) can be* ***steered*** *independently:* $\alpha_f \quad \alpha_r$
- *there is a rigid frame between the wheels with length L*



Hw #2: What are the *velocity kinematics* of this vehicle?



Q: Tricycle drive

• front wheel is powered and steerable
• back wheels tag along (without slipping...)

Mecos tricycle-drive robot

A Lego Mindstorms example!

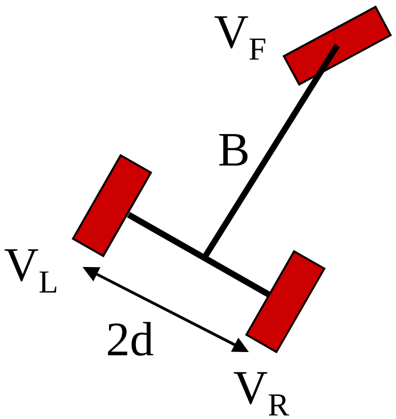
Mecos tricycle-drive robot

- The velocity of the front wheel is V_F

  (i.e., the conversion from angular velocity with wheel diameter is already done...)

- We know the distances 2d and B

Q: ?

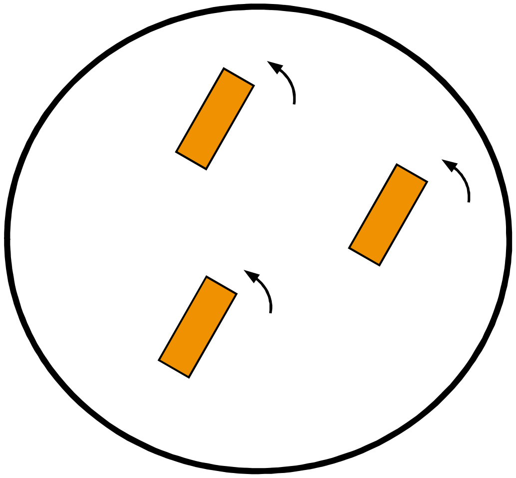


- What else do we need to know?
- Where is the ICC?
- How fast is the trikebot rotating around it?
- **What are V_L and V_R ?**

Synchro drive

Nomad 200

wheels rotate in tandem and remain parallel
all of the wheels are driven at the same speed



Where is the ICC ?

Questions (forward kinematics)

Given the wheel's velocities or positions, what is the robot's velocity/position ?

Are there any inherent system constraints?

1) Specify system measurements

2) Determine the point (the radius) around which the robot is turning.

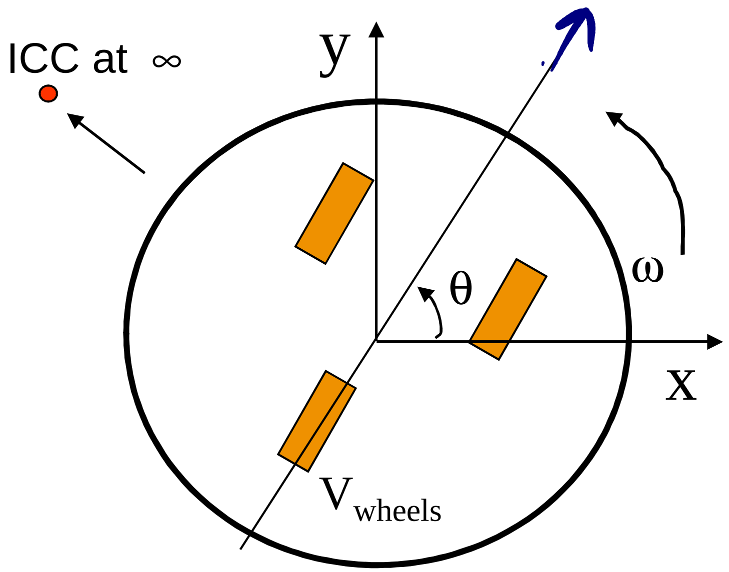
3) Determine the speed at which the robot is turning to obtain the robot velocity.

4) Integrate to find position.

Synchro drive

Nomad 200

wheels rotate in tandem and remain parallel
all of the wheels are driven at the same speed



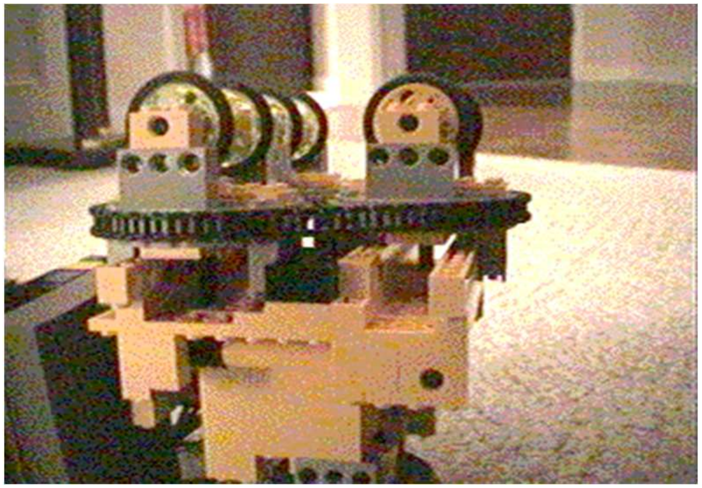
$$\left.\begin{aligned} V_{robot} &= V_{wheels} \\ \omega_{robot} &= \omega_{wheels} \end{aligned}\right\} \text{velocity}$$

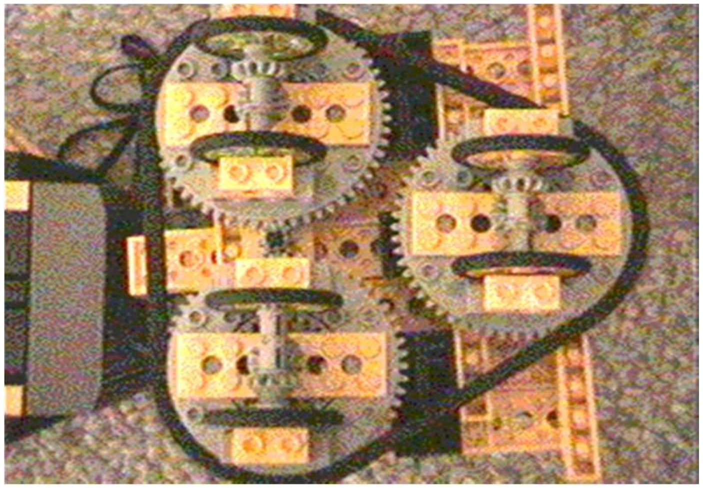
Kinematics

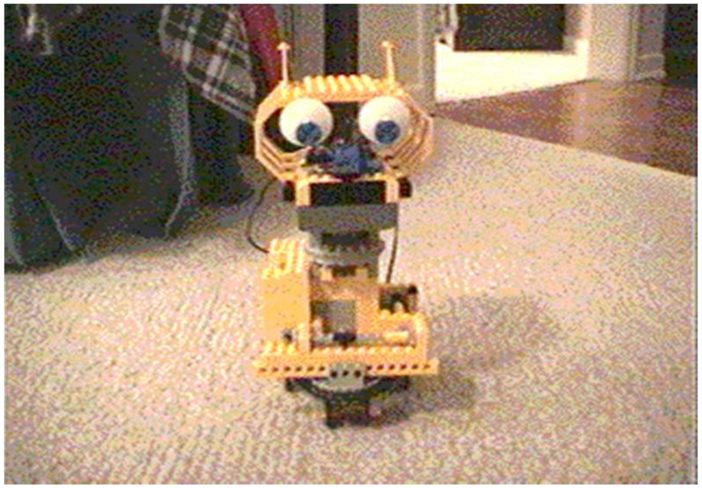
$$\left.\begin{aligned} \theta(t) &= \int \omega(t)\, dt \\ x(t) &= \int V_{wheels}(t) \cos(\theta(t))\, dt \\ y(t) &= \int V_{wheels}(t) \sin(\theta(t))\, dt \end{aligned}\right\} \text{position}$$

simpler to control, but ...

Lego Synchro







http://www.robotthoughts.com/?q=node/2589

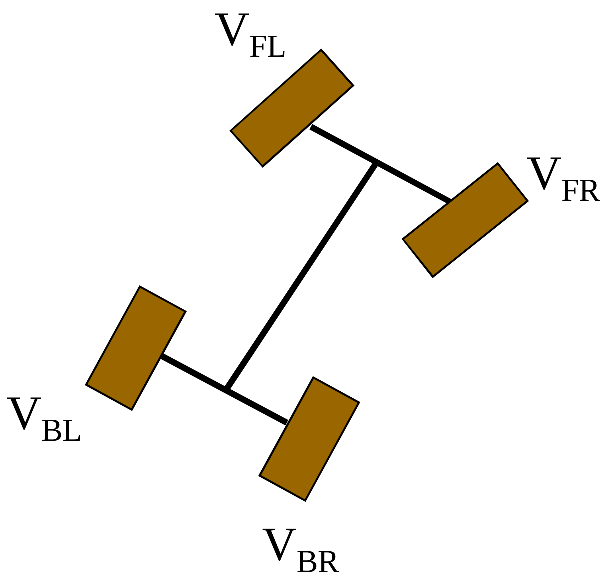
Anything that can be done, can be done with Lego

the lego motto…

Four-wheel Steering

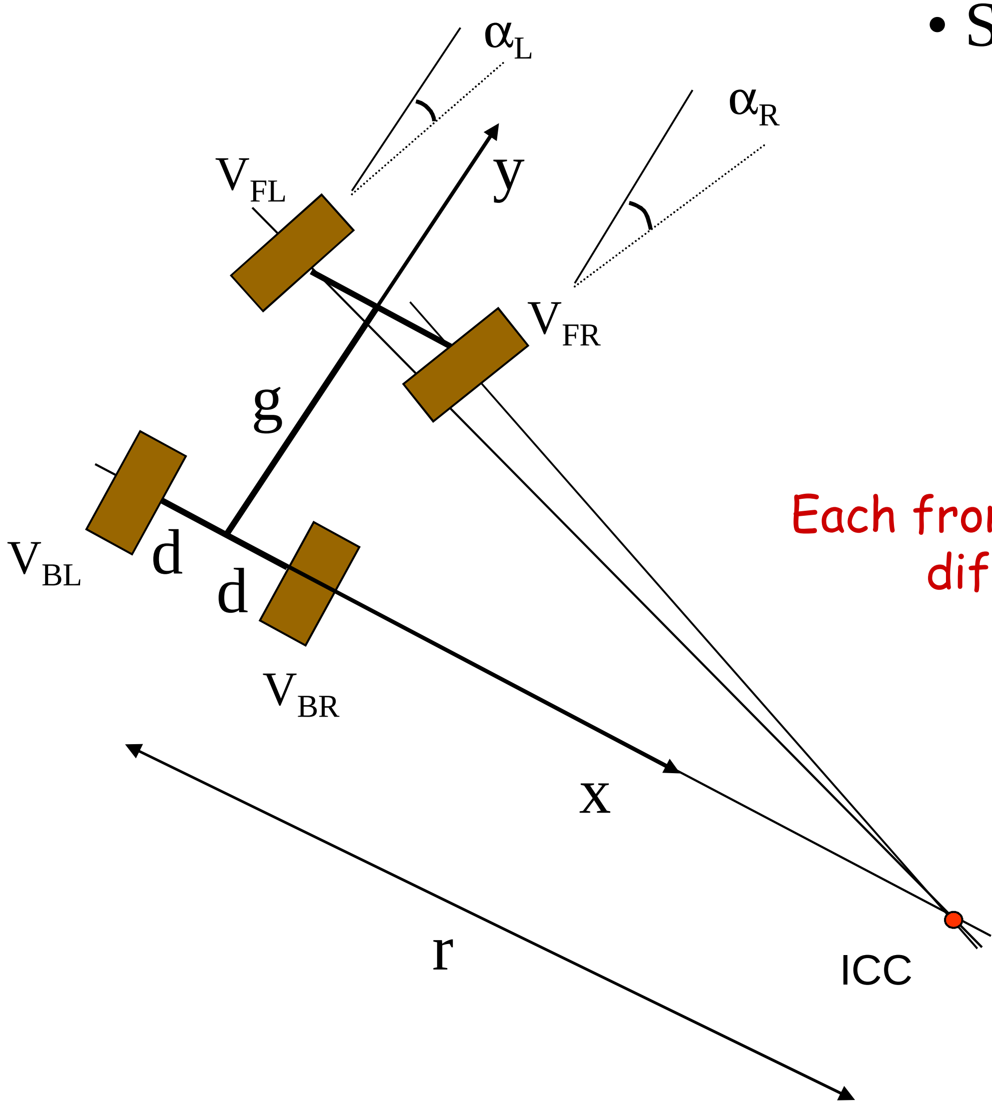
The kinematic challenges of parallel parking:

- wheels have limited turning angles
- no in-place rotation
- lots of SUVs around



What has to happen in order for a car's wheels not to slip while turning, i.e., for there to be a single ICC?

Ackerman Steering



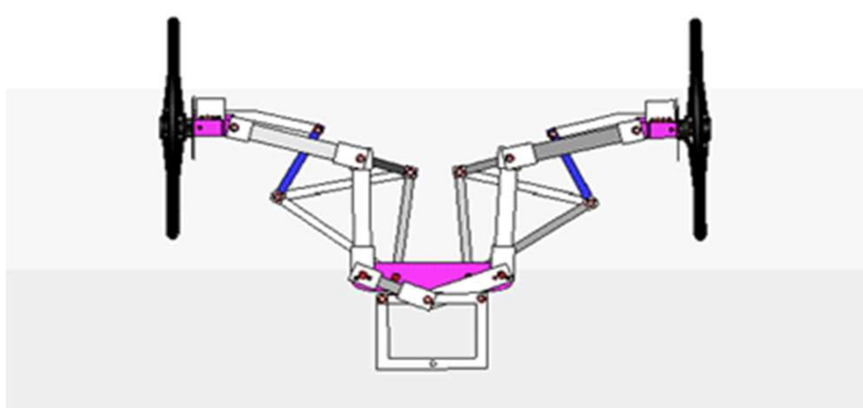
- Similar to a tricycle-drive robot

$$r = \frac{g}{\tan(\alpha_R)} + d$$

$$\frac{\omega g}{\sin(\alpha_R)} = V_{FR}$$

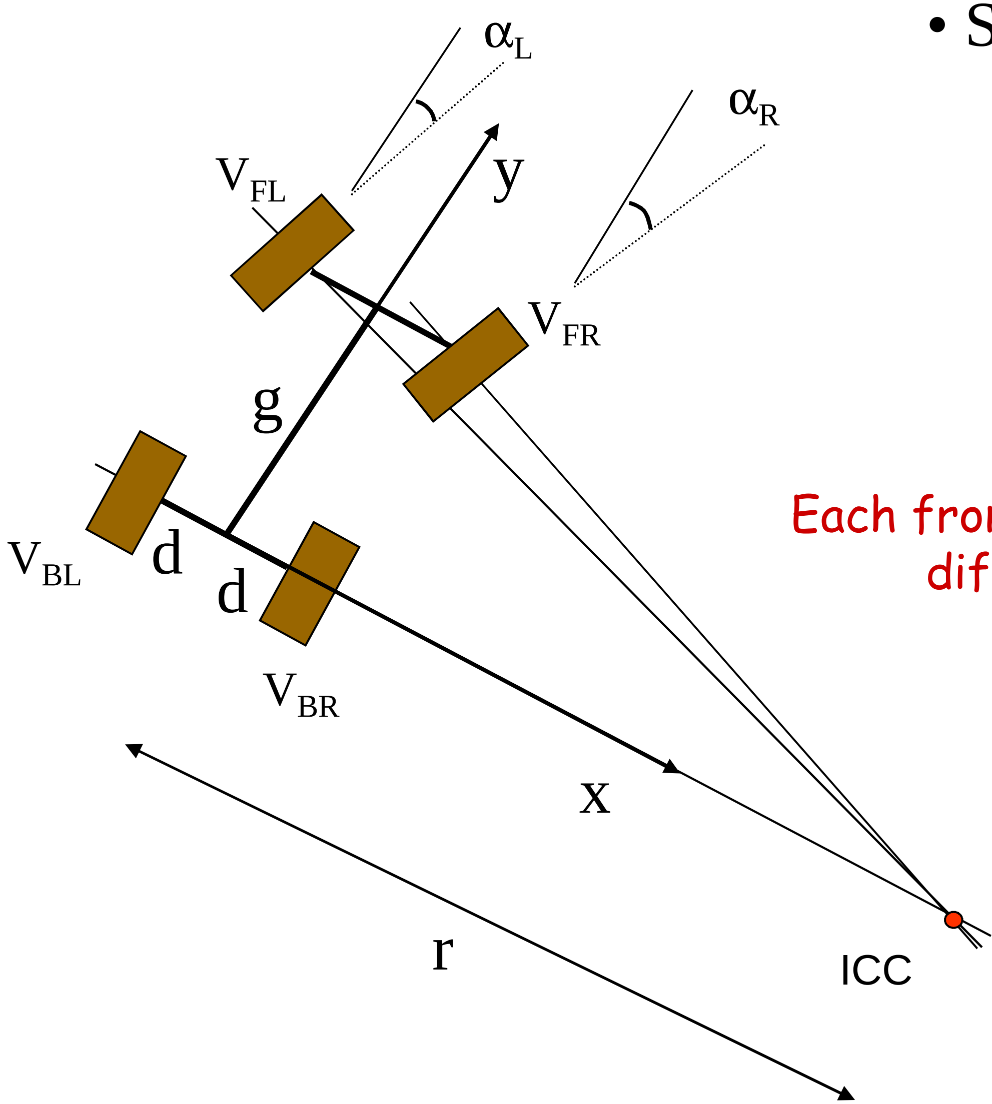
determines ω

Each front wheel has to turn a different amount !



6 bar design @ the Mechanisms lab at UCI

Ackerman Steering



- Similar to a tricycle-drive robot

$$r = \frac{g}{\tan(\alpha_R)} + d$$

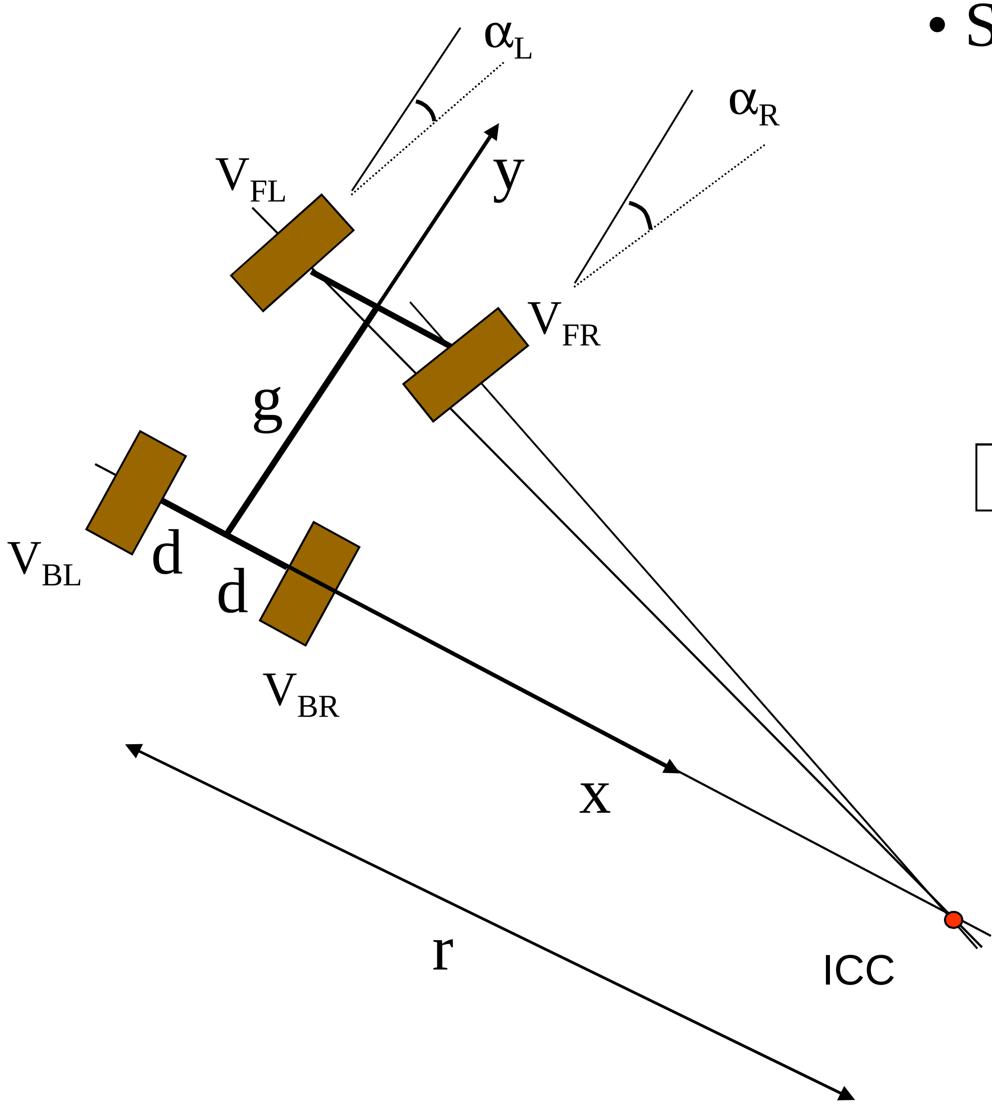
$$\frac{\omega g}{\sin(\alpha_R)} = V_{FR}$$

determines ω

Each front wheel has to turn a different amount !

but not in the Barbie Jeep...

Ackerman Steering



- Similar to a tricycle-drive robot

$$r = \frac{g}{\tan(\alpha_R)} + d$$

$$\frac{\omega g}{\sin(\alpha_R)} = V_{FR}$$ determines ω

The other wheel velocities are now fixed!

$$\frac{\omega g}{\sin(\alpha_L)} = V_{FL}$$

$$\alpha_L = \tan^{-1}(g / (r + d))$$

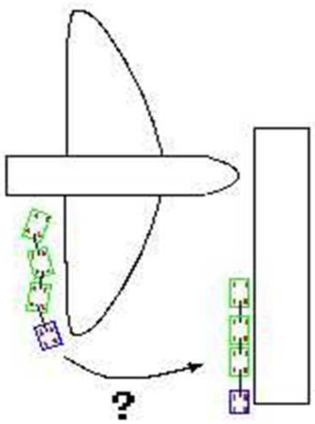
$$\omega(r - d) = V_{BR}$$

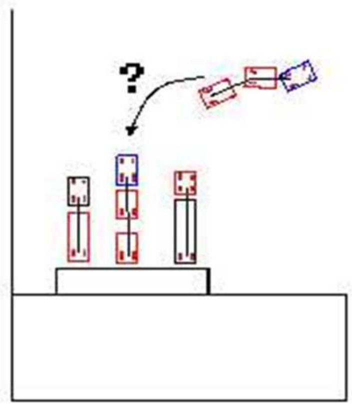
$$\omega(r + d) = V_{BL}$$

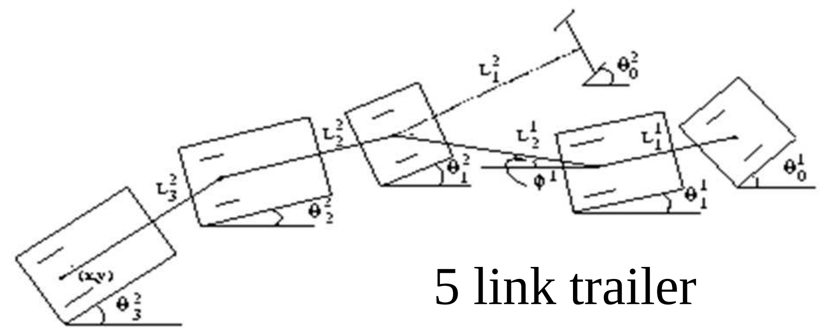
This is just the cab...

The Big Rigs

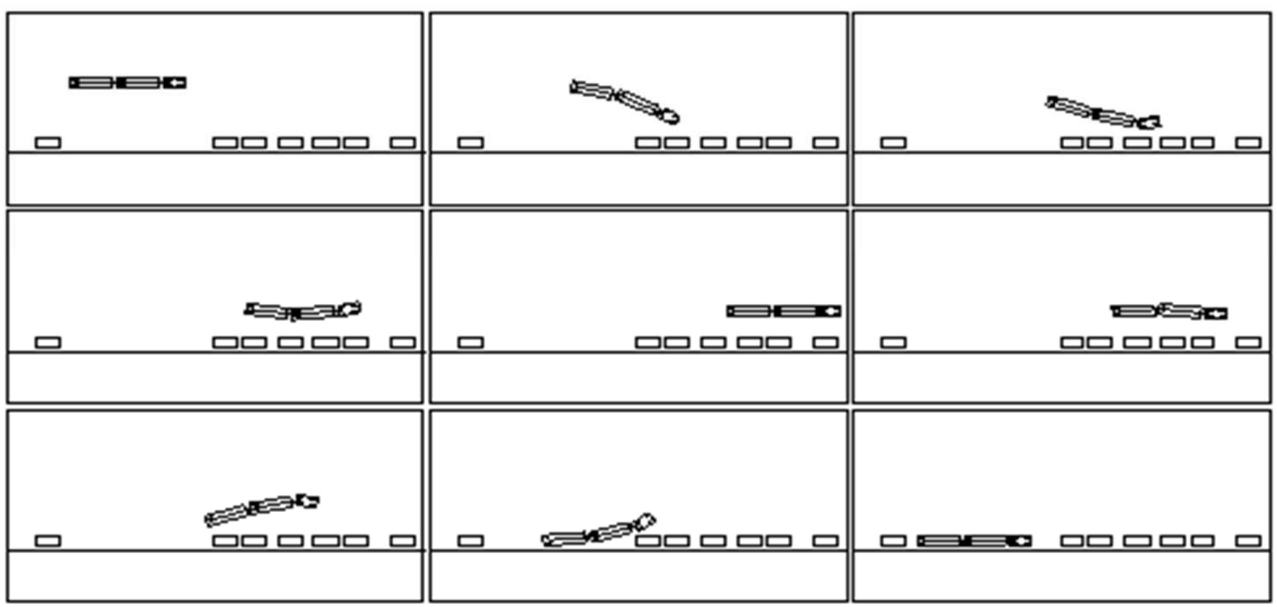
Applications:







5 link trailer
2 controlled angles

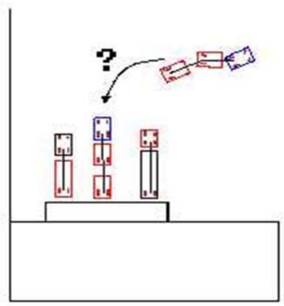


Parking two trailers

scary stuff...

Multiple trailers ?

Double trailers?



"Double-trailer vehicles, including both western doubles and LCVs, had a **threefold increased risk of crash** involvement compared to single-trailer vehicles, even after adjusting for other variables significantly affecting crash risk, including empty or loaded travel, driver age, hours driving, type of carrier, and whether the carrier operated intrastate or interstate."

http://www.usroads.com/journals/rmej/9902/rm990201.htm

automatic control / robotic applications:

"pavement-loading" application (no parking, I suppose)

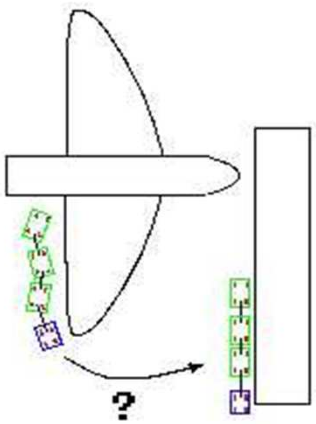
http://www.westrack.com/wt_03.htm

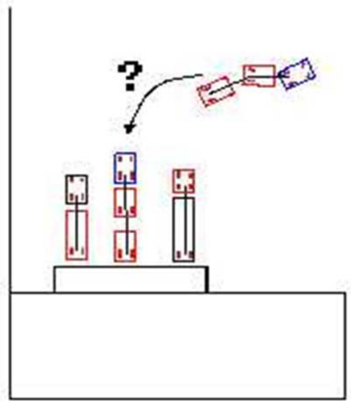
see www.knottlab.com/animations.aspx

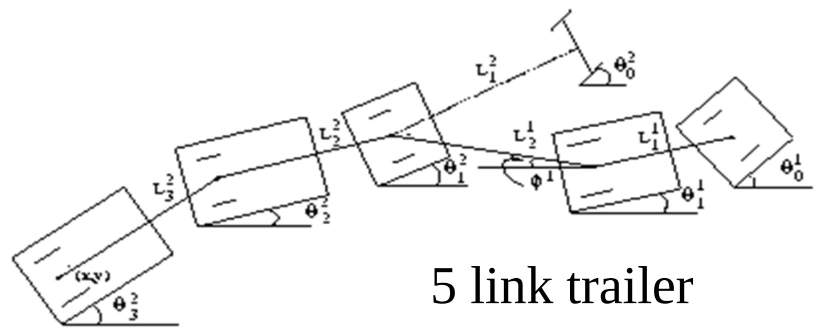
safety factor requirements?

The Big Rigs

Applications:

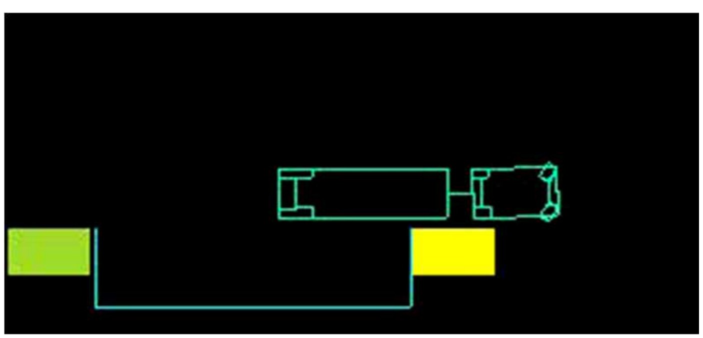






5 link trailer
2 controlled angles

Parking/reversing with trailers



Nonholonomicity

All of the robots mentioned thus far share an important (if frustrating) property: they are **nonholonomic** .

- makes it more difficult to navigate between two arbitrary points

- need to resort to techniques like parallel parking

Nonholonomicity

All of the robots mentioned thus far share an important (if frustrating) property: they are **nonholonomic** .

- makes it more difficult to navigate between two arbitrary points

- need to resort to techniques like parallel parking

By definition, a robot is **nonholonomic** if it *can not* move to change its pose instantaneously in all available directions.

Nonholonomicity

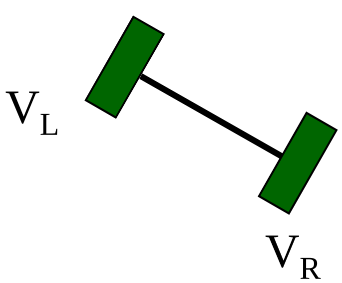
All of the robots mentioned thus far share an important (if frustrating) property: they are **nonholonomic** .

- makes it more difficult to navigate between two arbitrary points

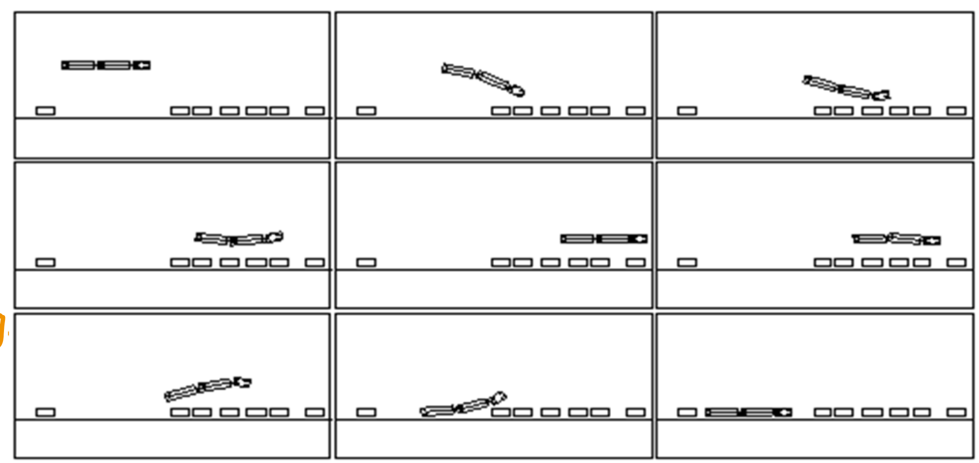
- need to resort to techniques like parallel parking

By definition, a robot is **nonholonomic** if it *can not* move to change its pose instantaneously in all available directions.

differential-drive robots are nonholonomic



multiple-trailer rigs are "very" nonholonomic



Nonholonomicity

All of the robots mentioned thus far share an important (if frustrating) property: they are **nonholonomic** .

- makes it more difficult to navigate between two arbitrary points

- need to resort to techniques like parallel parking

By definition, a robot is **holonomic** if it *can* move to change its pose instantaneously in all available directions.

Nonholonomicity

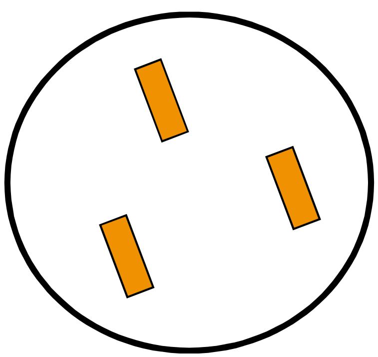
All of the robots mentioned thus far share an important (if frustrating) property: they are **nonholonomic** .

- makes it more difficult to navigate between two arbitrary points

- need to resort to techniques like parallel parking

By definition, a robot is **holonomic** if it *can* move to change its pose instantaneously in all available directions.

i.e., the robot's *differential motion* is unconstrained.



Synchro Drive

is the Nomad holonomic?

Nonholonomicity

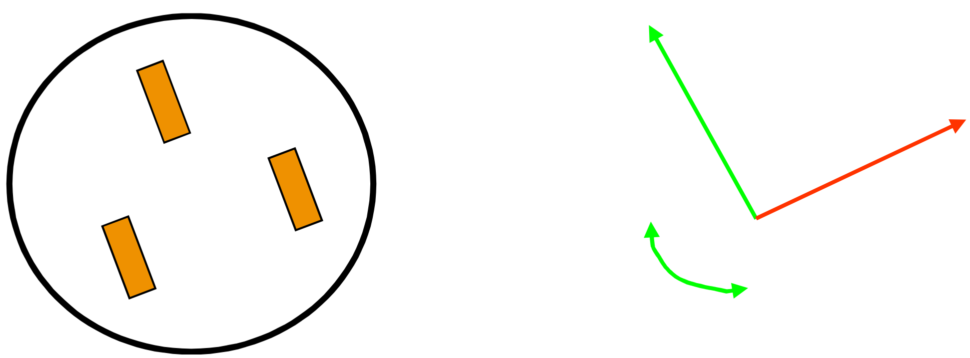
All of the robots mentioned thus far share an important (if frustrating) property: they are **nonholonomic** .

- makes it more difficult to navigate between two arbitrary points

- need to resort to techniques like parallel parking

By definition, a robot is **holonomic** if it *can* move to change its pose instantaneously in all available directions.

But the Nomad's differential motion ***is*** constrained.



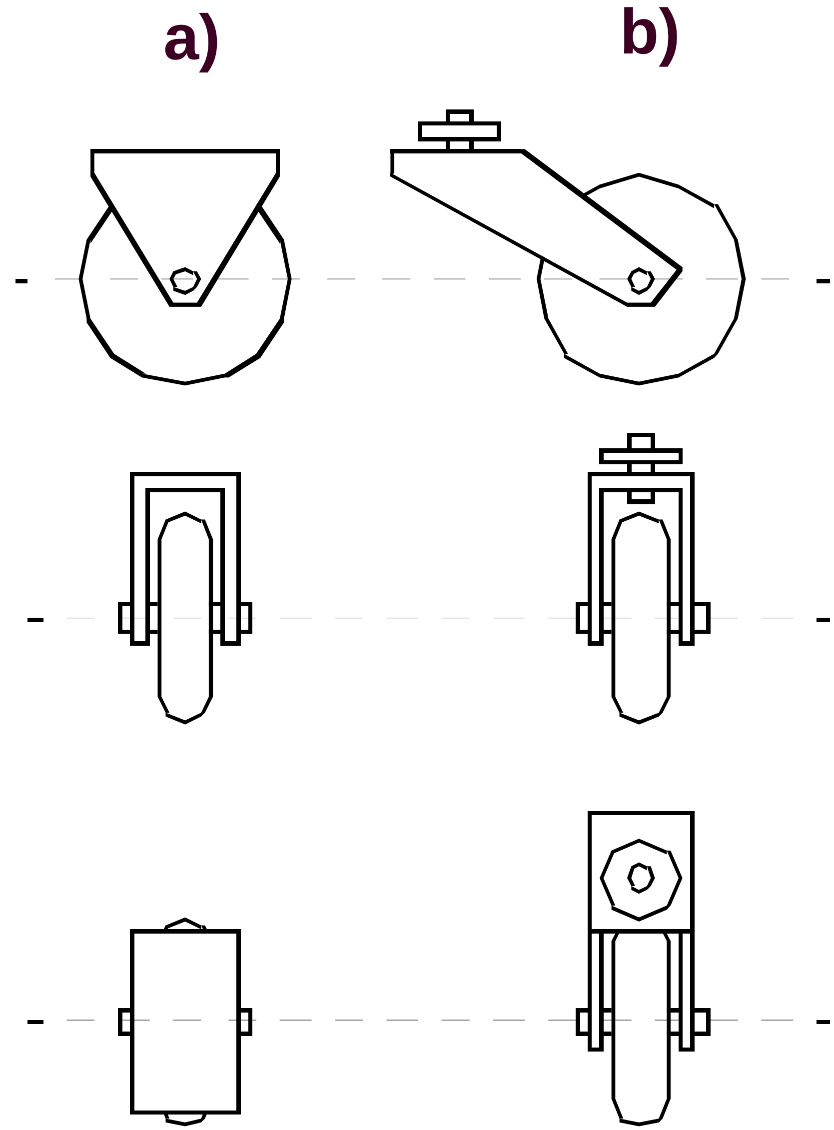
Synchro Drive

two DOF are freely controllable; the third is only indirectly accessible

The Four Basic Wheels Types

- a) Standard wheel: Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point

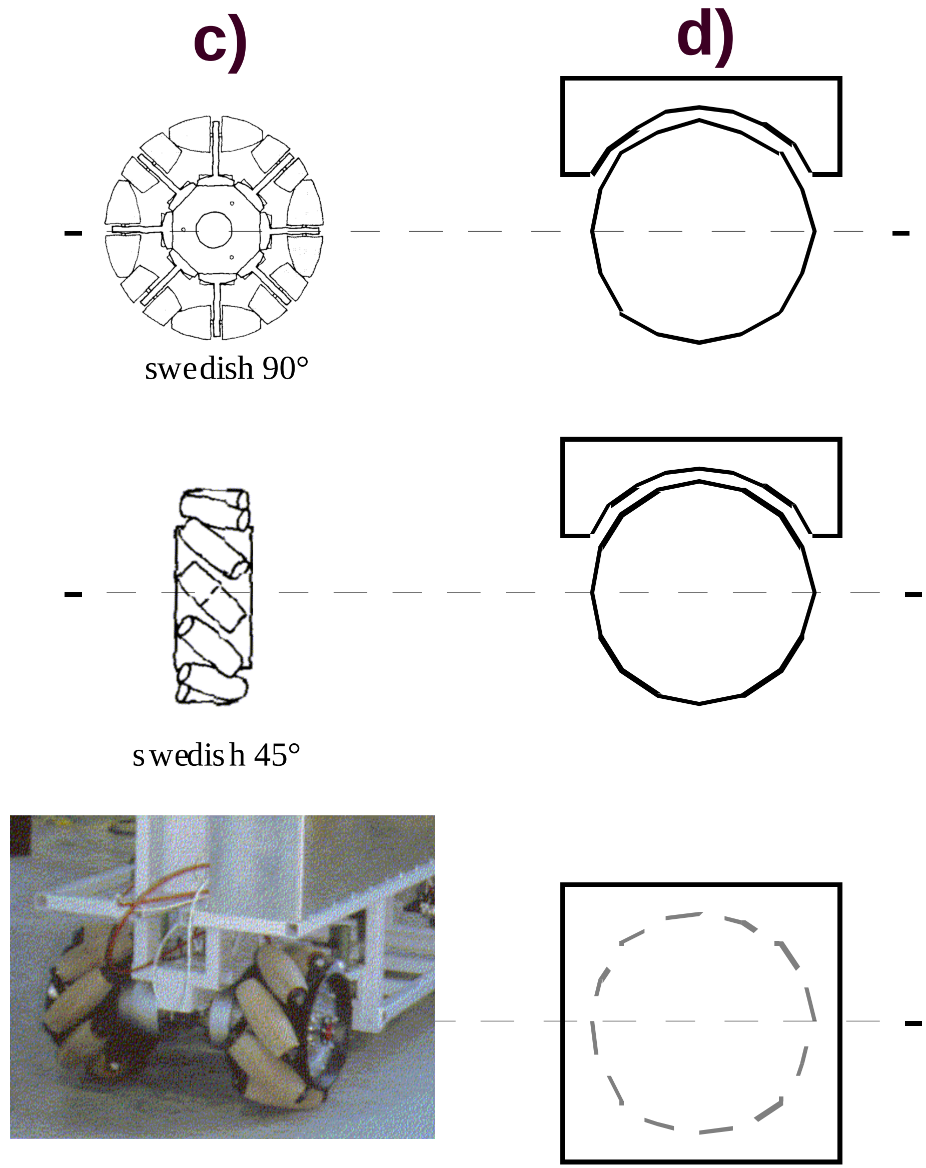
- b) Castor wheel: Three degrees of freedom; rotation around the wheel axle, the contact point and the castor axle



The Four Basic Wheels Types

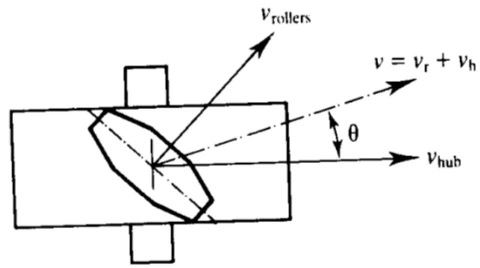
- c) Swedish wheel: Three degrees of freedom; rotation around the (motorized) wheel axle, around the rollers and around the contact point

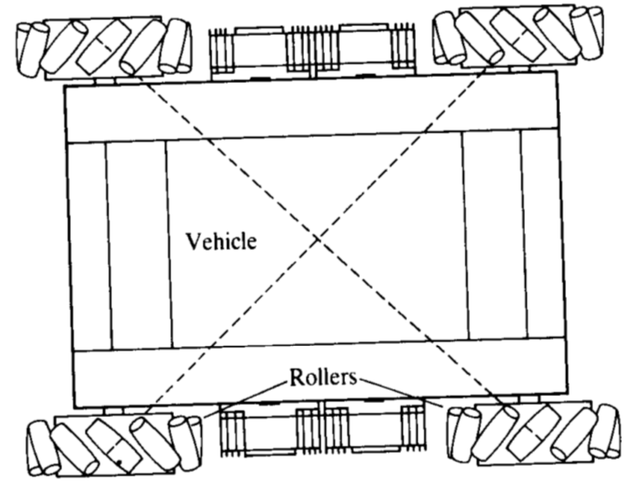
- d) Ball or spherical wheel: Suspension technically not solved

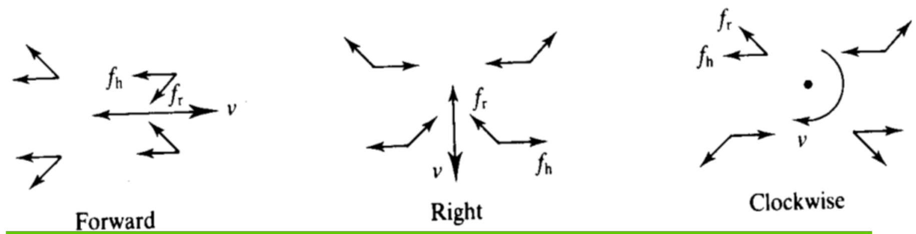


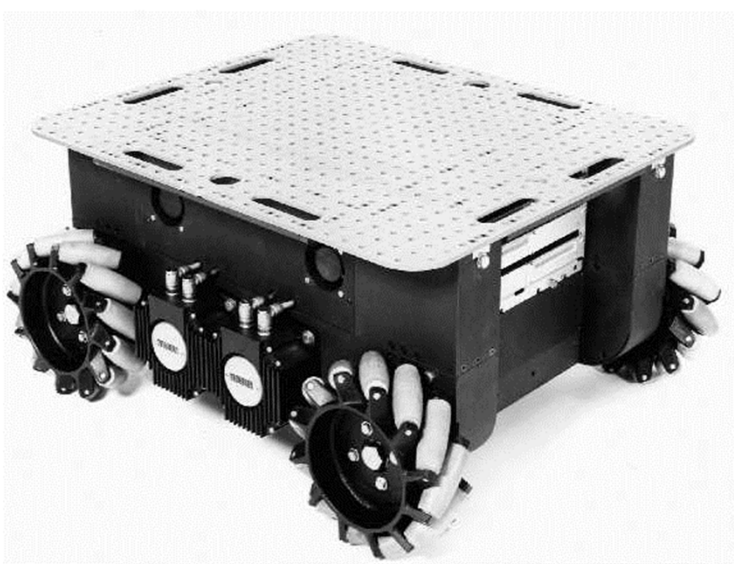
Uranus, CMU: Omnidirectional Drive with 4 Wheels

- Movement in the plane has 3 DOF
  - *thus only three wheels can be independently controlled*
  - *It might be better to arrange three swedish wheels in a triangle*









$$v_y = (v_0 + v_1 + v_2 + v_3)/4$$

$$v_x = (v_0 - v_1 + v_2 - v_3)/4$$

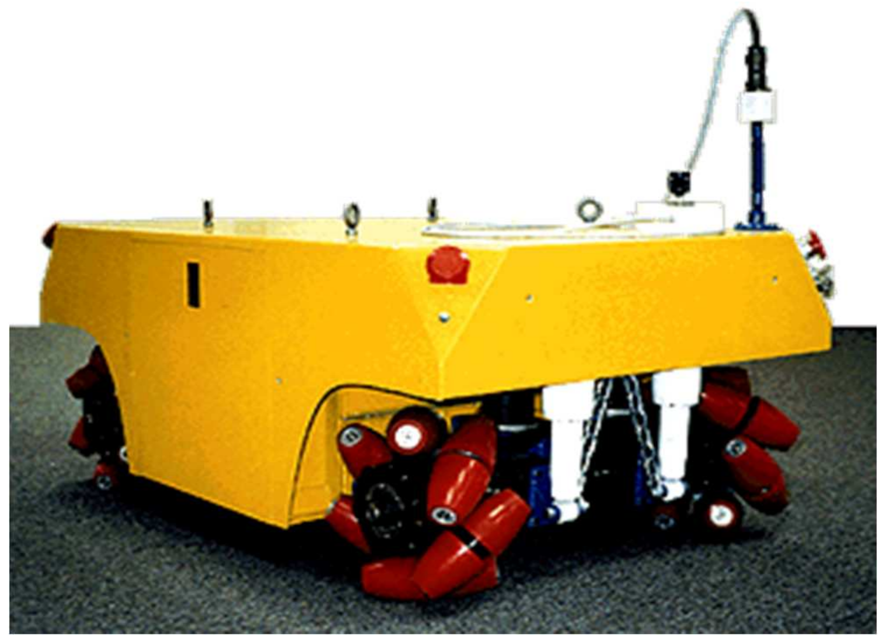
$$v_\theta = (v_0 + v_1 - v_2 - v_3)/4$$

$$v_{error} = (v_0 - v_1 - v_2 + v_3)/4$$

Holonomic Robots

Navigation is simplified considerably if a robot *can* move instantaneously in any direction, i.e., is **holonomic**.

Omniwheels



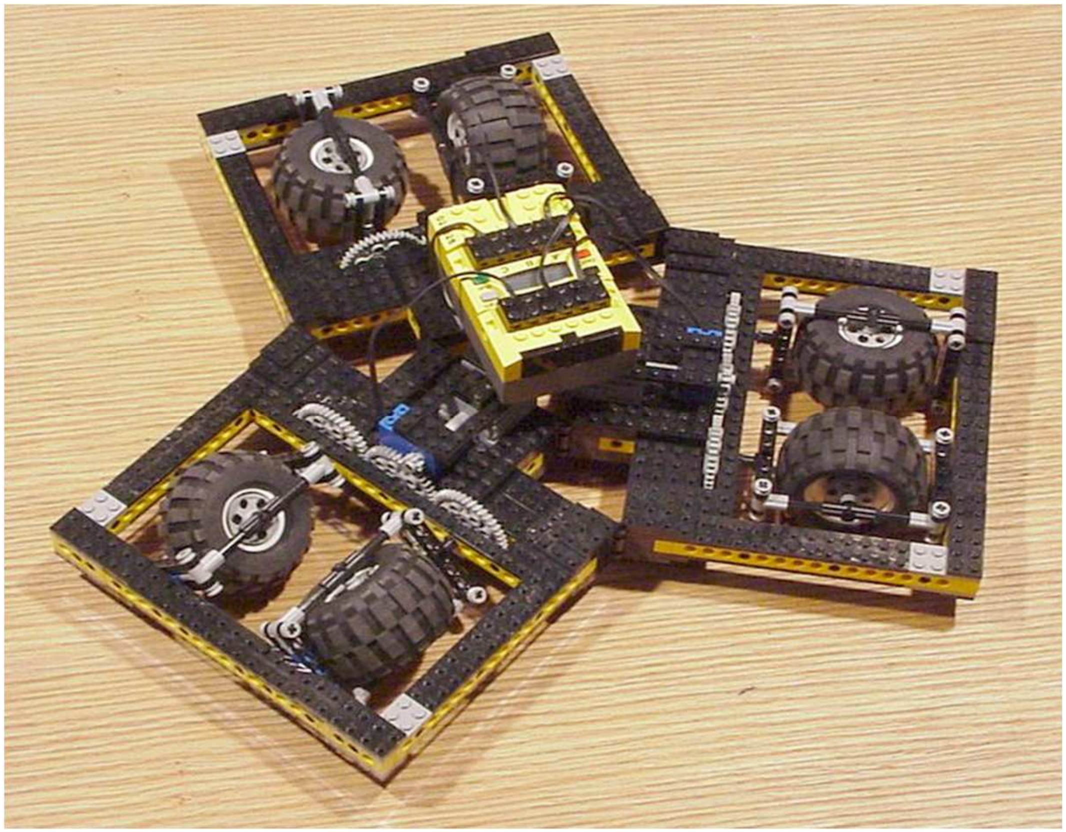
Mecanum wheels

tradeoffs in locomotion/wheel design

show examples…

if it can be done at all ...

Holonomic Designs



Killough Platform

in action and “frisbeeing”

lego logo…

Holonomic hype

"The PeopleBot is a *highly holonomic* platform, able to navigate in the tightest of spaces…"

www.activmedia.com

Robot of the Day

Rotundus, Inc. of Sweden – spherical robot

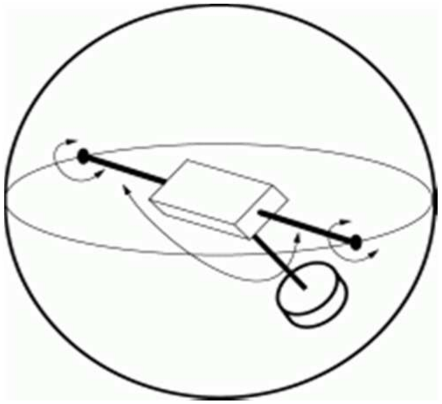
How?

all internal (*proprioceptive) sensing*

Robot of the Day

Rotundus, Inc. of Sweden – spherical robot

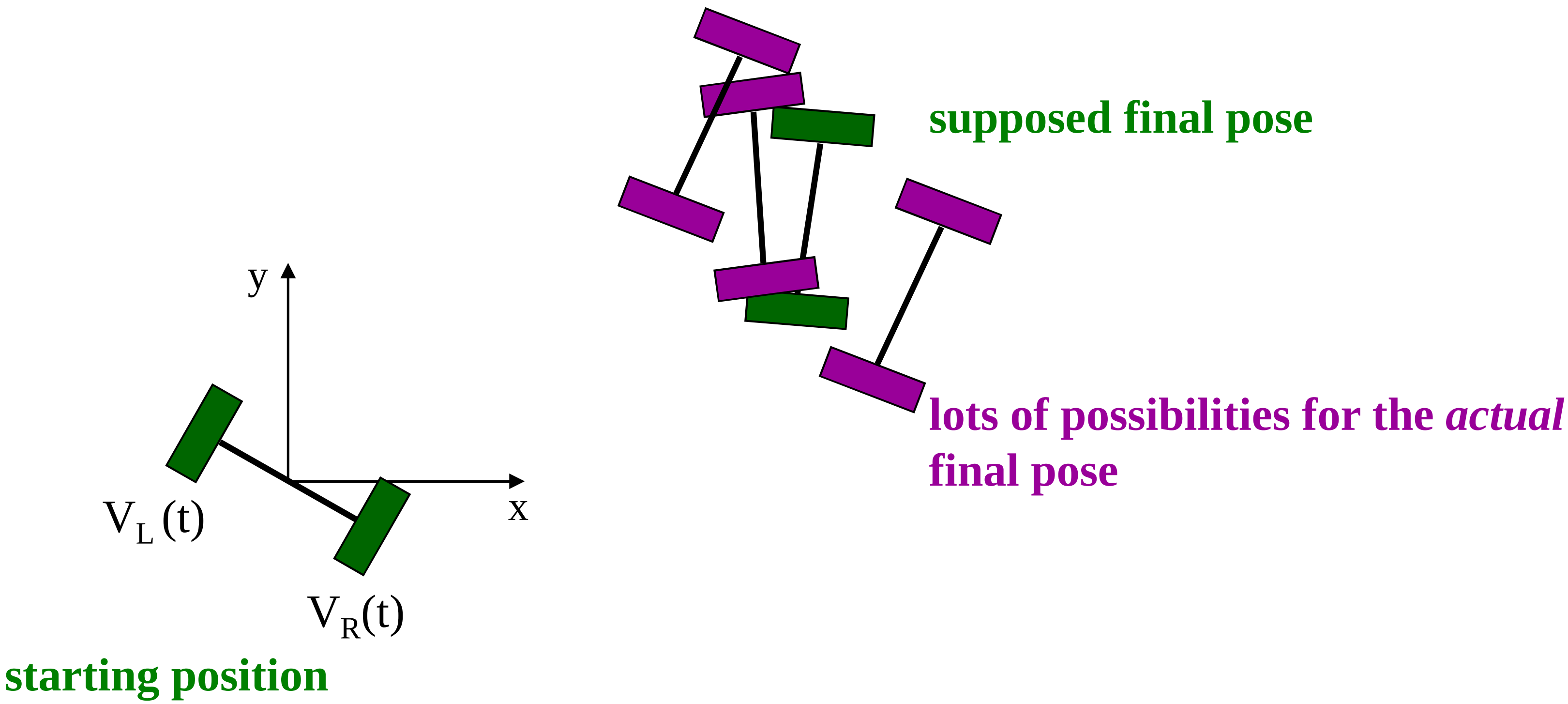
Internal, motor-driven pendulum for shifting the center of mass…



Probabilistic Kinematics

Key question:

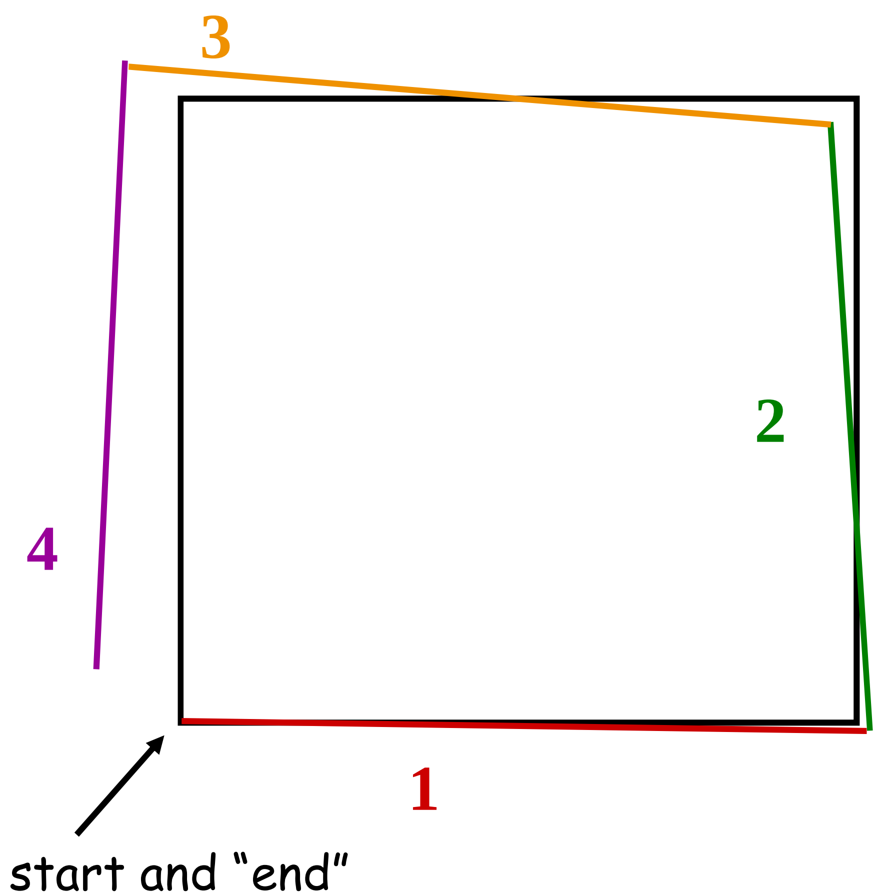
We may know where our robot is *supposed to be*, but in reality it might be somewhere else…



What should we do?

MODEL the error in order to reason about it!

Running around in squares



• Create a program that will run your robot in a square (~2m to a side), pausing after each side before turning and proceeding.

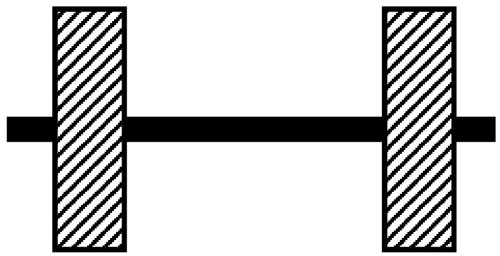
• For 10 runs, collect both the odometric estimates of where the robot thinks it is and where the robot ***actually is*** after each side.

• You should end up with two sets of 30 angle measurements and 40 length measurements: one set from odometry and one from "ground-truth."

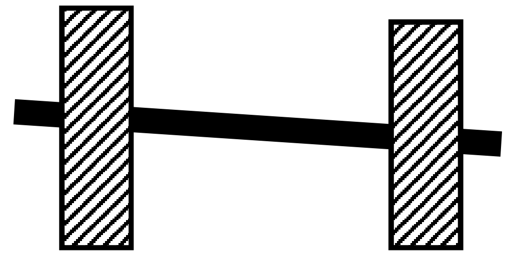
• Find the **mean** and the **standard deviation** of the ***differences*** between odometry and ground truth for the angles and for the lengths – this is the robot's ***motion uncertainty model***.

This provides a *probabilistic kinematic* model.

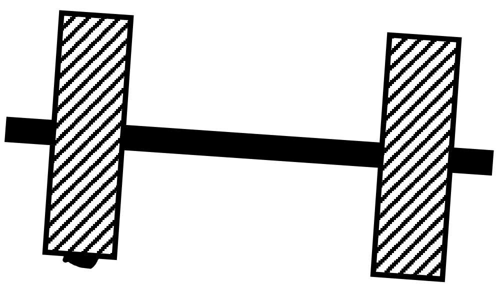
Reasons for Motion Errors



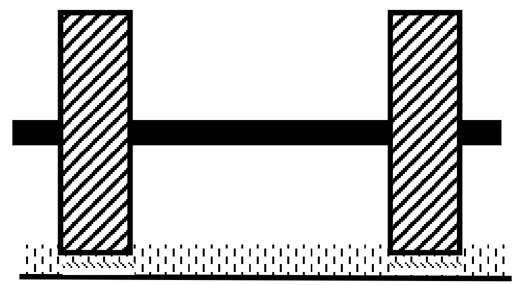
ideal case



different wheel diameters



bump

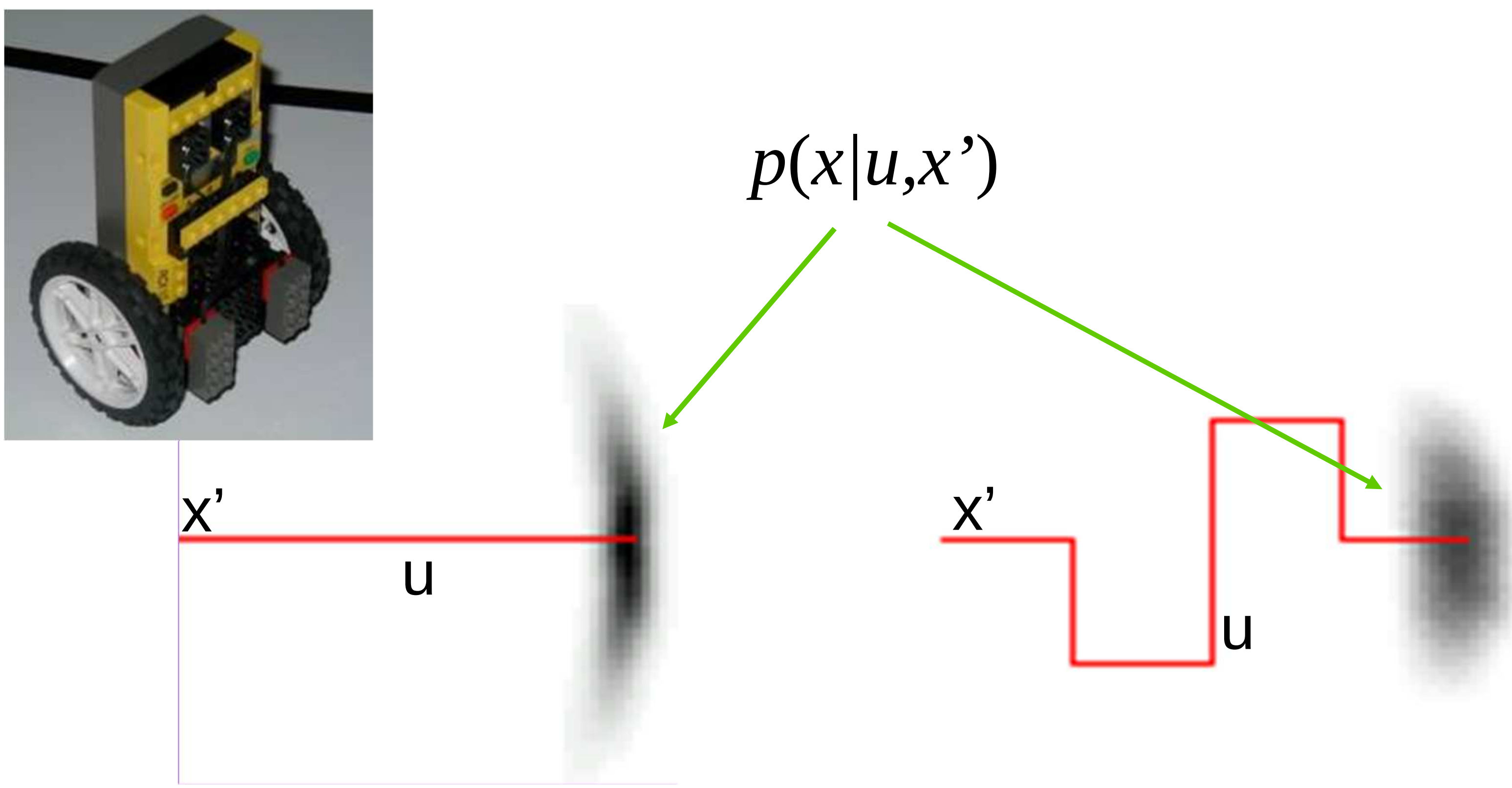


carpet

and many more …

Uncertain motion: Probabilistic kinematics

- Repeated application of the sensor model for short movements.
- For a motion command **u**, and starting position **x'**, what is the probability to end at **x**?



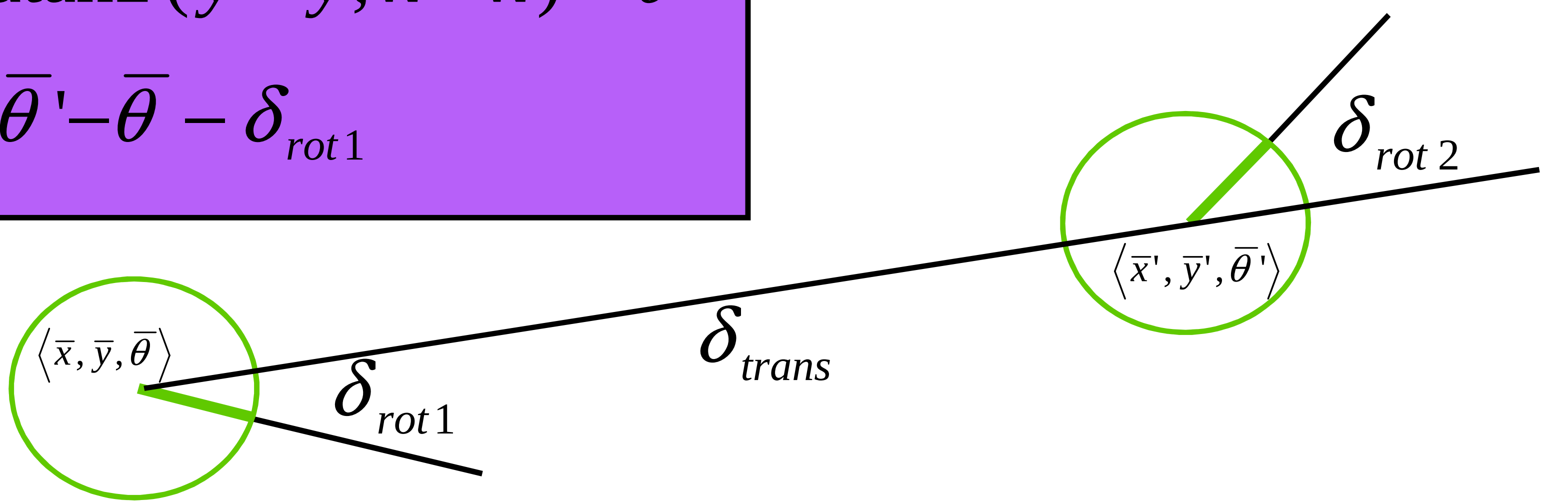
Odometry Model

- Robot moves from $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$
- Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



The atan2 Function

- Extends the inverse tangent and correctly copes with the signs of x and y.

$$\mathsf{atan2}(y,x) \;=\; \begin{cases} \mathsf{atan}(y/x) & \mathsf{if}\ x > 0 \\ \mathsf{sign}(y)\ (\pi - \mathsf{atan}(|y/x|)) & \mathsf{if}\ x < 0 \\ 0 & \mathsf{if}\ x = y = 0 \\ \mathsf{sign}(y)\ \pi/2 & \mathsf{if}\ x = 0, y \neq 0 \end{cases}$$

Noise Model for Odometry

- The measured motion is given by the true motion corrupted with noise.

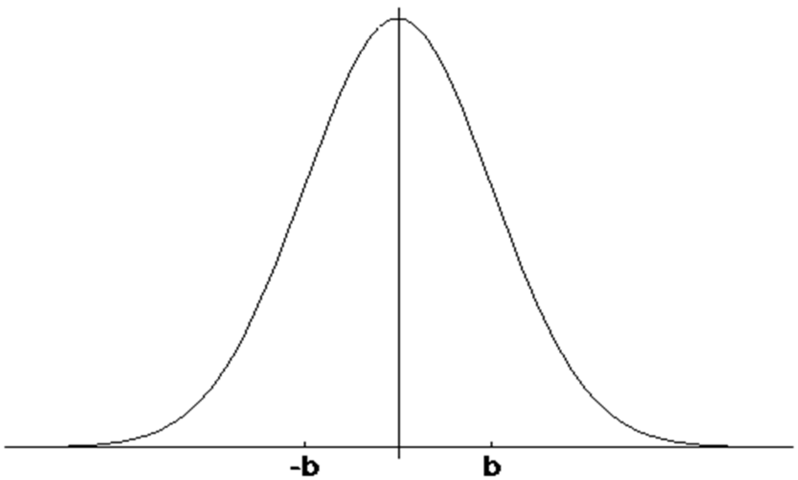
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot1} + \delta_{rot2}|}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|}$$

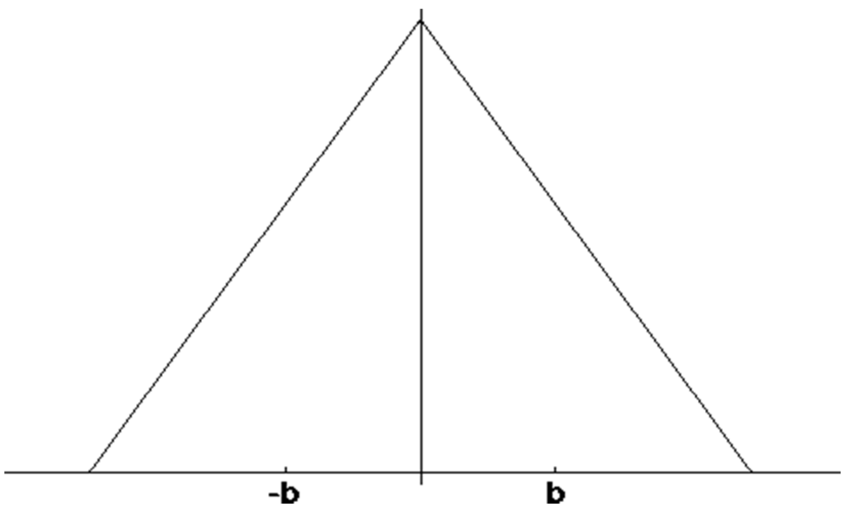
Typical Distributions for Probabilistic Motion Models

Normal distribution



$$\varepsilon_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}$$

Triangular distribution



$$\varepsilon_{\sigma^2}(x) = \begin{cases} 0 \text{ if } |x| > \sqrt{6\sigma^2} \\ \dfrac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} \end{cases}$$

Calculating the Probability (zero-centered)

- For a normal distribution

1. Algorithm **prob_normal_distribution**(a,b):
2. return $\frac{1}{\sqrt{2\pi\, b^2}} \exp\left\{-\frac{1}{2}\frac{a^2}{b^2}\right\}$

- For a triangular distribution

1. Algorithm **prob_triangular_distribution**(a,b):
2. return $\max\left\{0, \frac{1}{\sqrt{6}\, b} - \frac{|a|}{6\, b^2}\right\}$

Calculating the Posterior Given x, x', and u

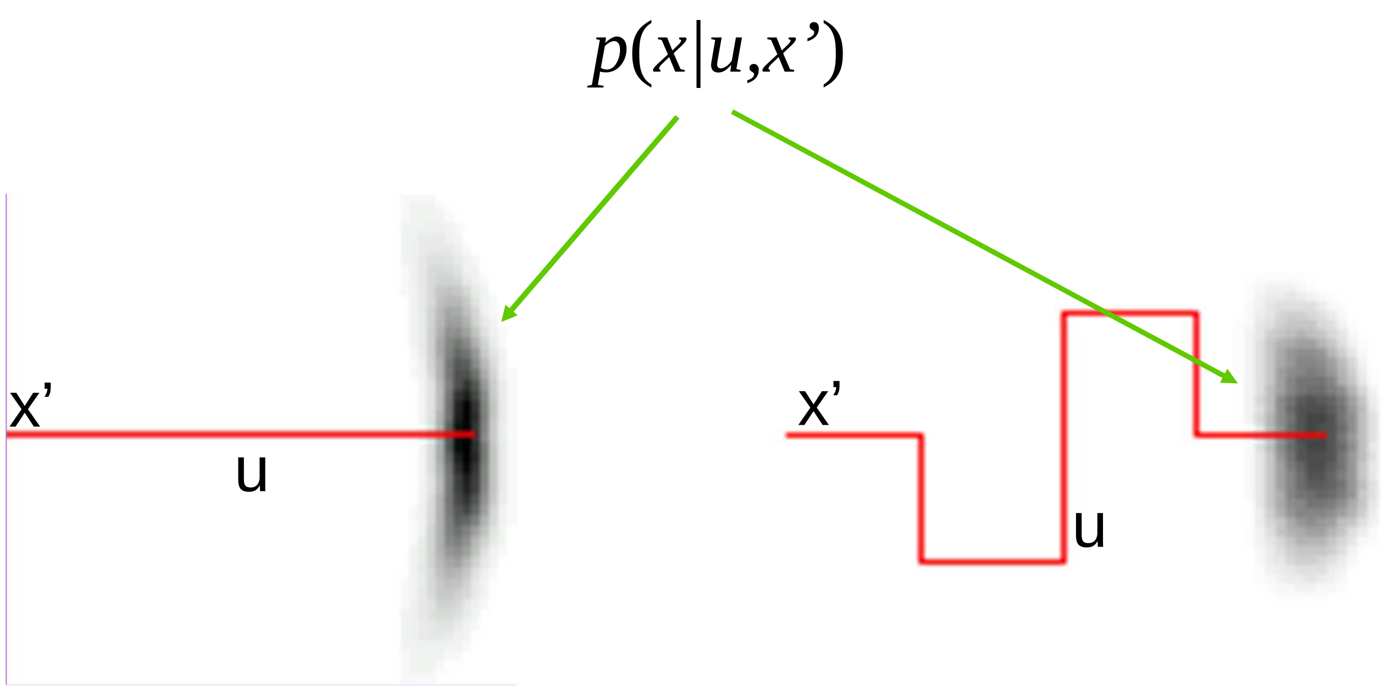
1. Algorithm **motion_model_odometry(x,x',u)**
2. $\delta_{trans} = \sqrt{(\bar{x}'-\bar{x})^2 + (\bar{y}'-\bar{y})^2}$
3. $\delta_{rot1} = \mathrm{atan2}(\bar{y}'-\bar{y}, \bar{x}'-\bar{x}) - \bar{\theta}$
4. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
5. $\hat{\delta}_{trans} = \sqrt{(x'-x)^2 + (y'-y)^2}$
6. $\hat{\delta}_{rot1} = \mathrm{atan2}(y'-y, x'-x) - \bar{\theta}$
7. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
8. $p_1 = \mathrm{prob}(\delta_{\mathrm{rot}1} - \hat{\delta}_{\mathrm{rot}1}, \alpha_1 |\hat{\delta}_{\mathrm{rot}1}| + \alpha_2 \hat{\delta}_{\mathrm{trans}})$
9. $p_2 = \mathrm{prob}(\delta_{\mathrm{trans}} - \hat{\delta}_{\mathrm{trans}}, \alpha_3 \hat{\delta}_{\mathrm{trans}} + \alpha_4 (|\hat{\delta}_{\mathrm{rot}1}| + |\hat{\delta}_{\mathrm{rot}2}|))$
10. $p_3 = \mathrm{prob}(\delta_{\mathrm{rot}2} - \hat{\delta}_{\mathrm{rot}2}, \alpha_1 |\hat{\delta}_{\mathrm{rot}2}| + \alpha_2 \hat{\delta}_{\mathrm{trans}})$
11. return $p_1 \cdot p_2 \cdot p_3$

odometry values (u)

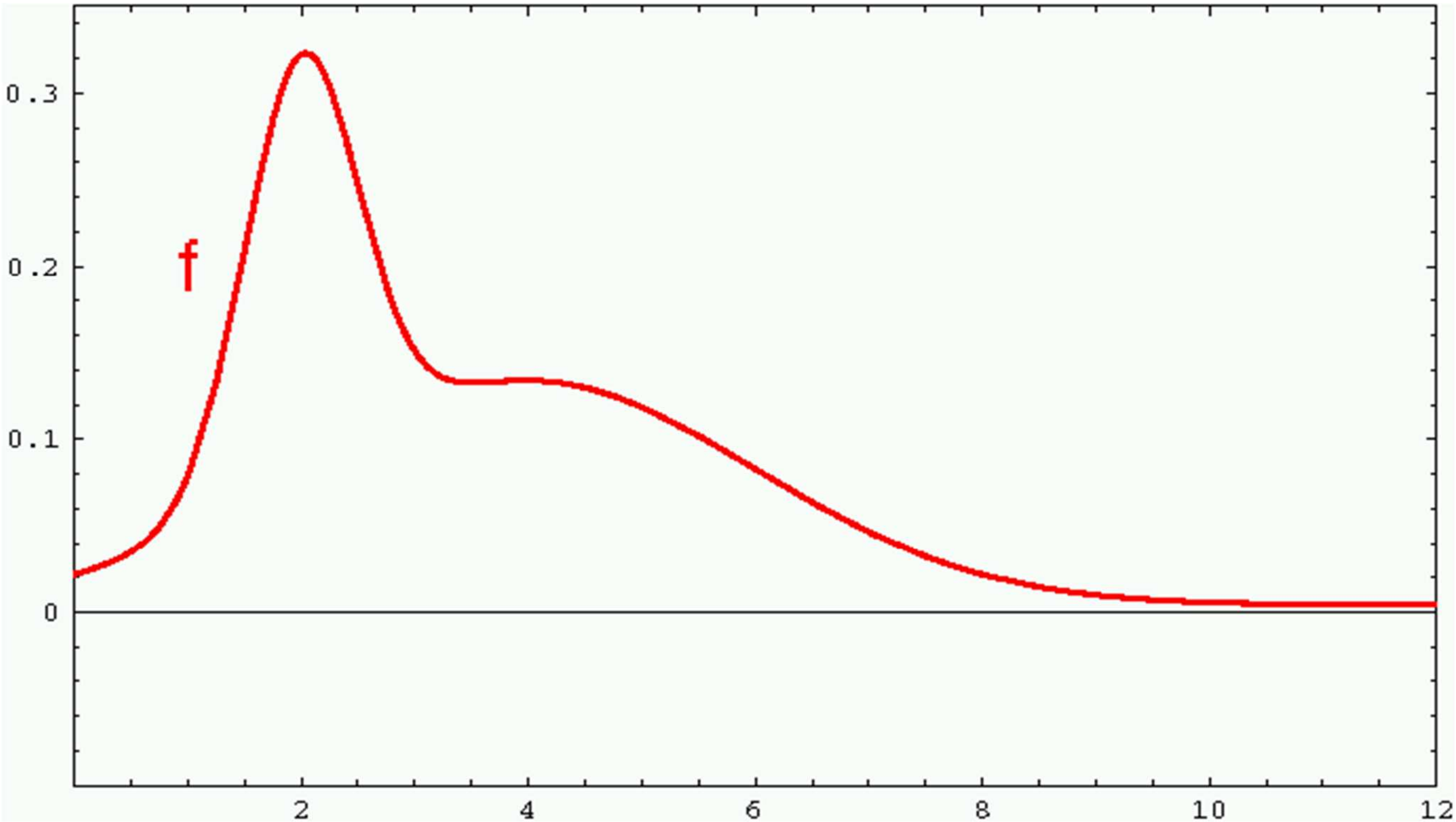
values of interest (x,x')

Application

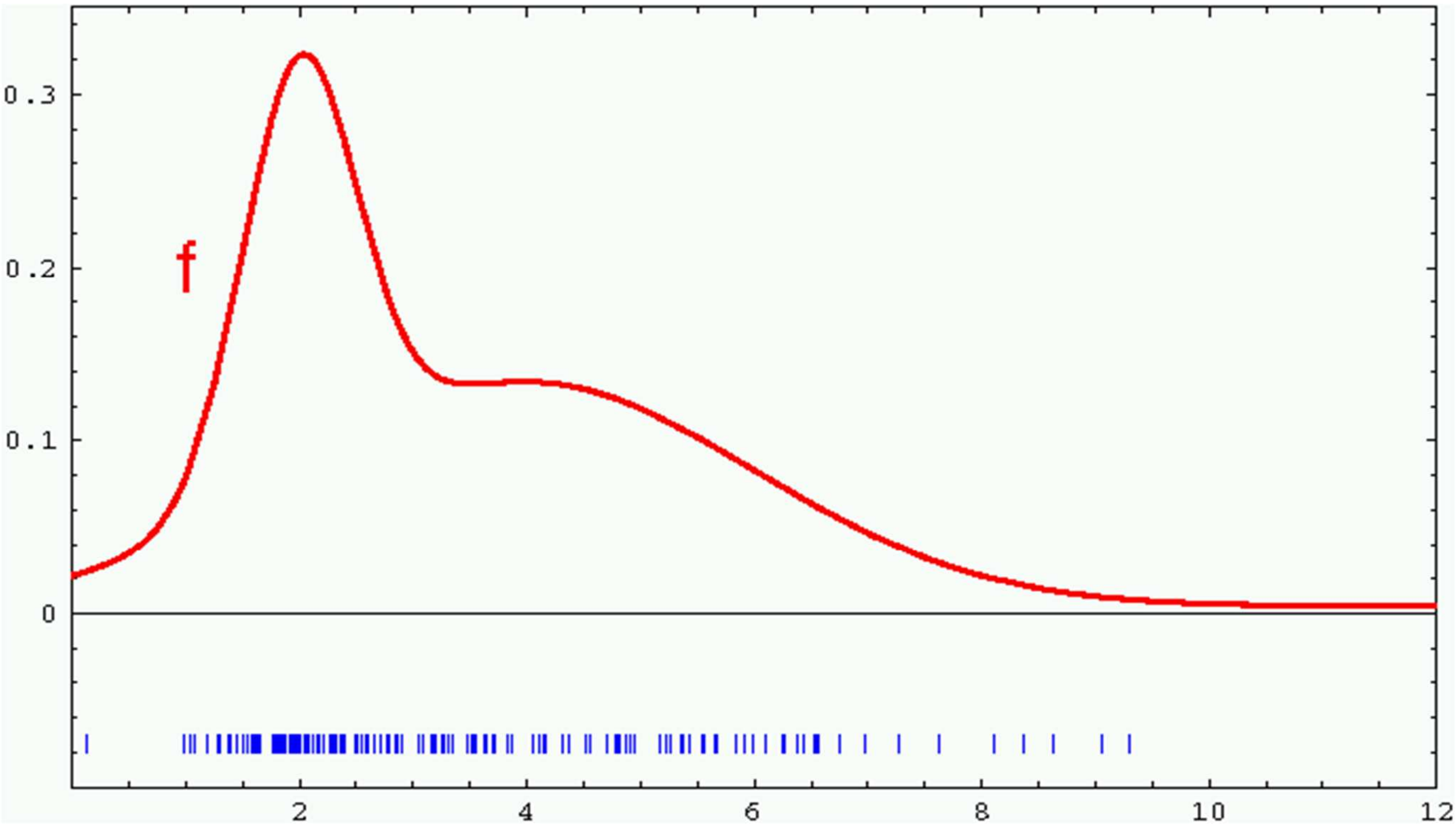
- Repeated application of the sensor model for short movements.
- Typical banana-shaped distributions obtained for 2d-projection of 3d posterior.



Sample-based Density Representation



Sample-based Density Representation



How to Sample from Normal or Triangular Distributions?

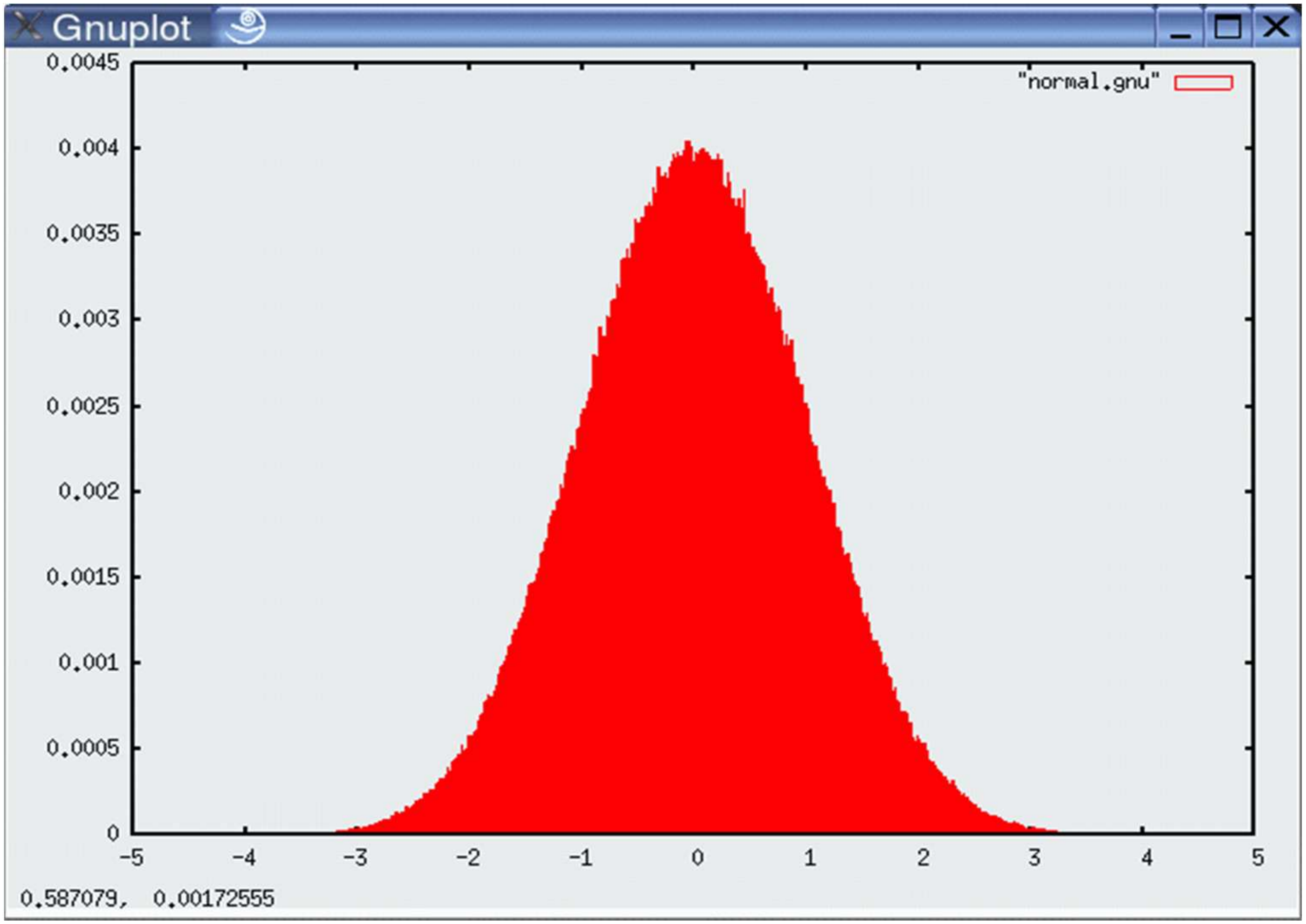
- Sampling from a normal distribution

1. Algorithm **sample_normal_distribution**(b):
2. return $$\frac{1}{2}\sum_{i=1}^{12} rand(-b, b)$$

- Sampling from a triangular distribution

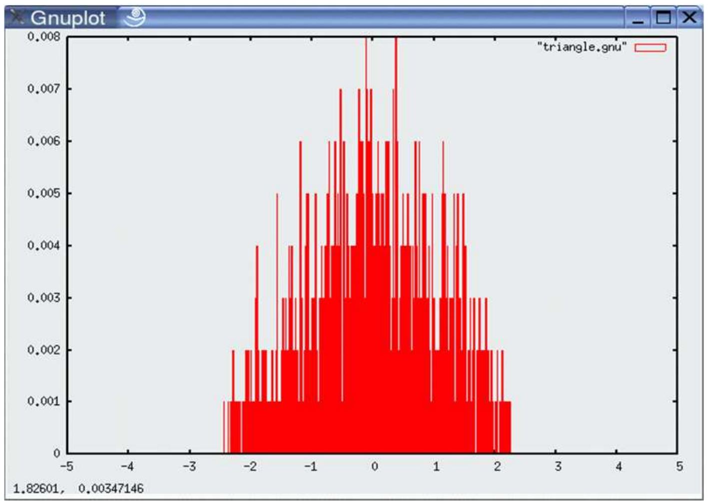
1. Algorithm **sample_triangular_distribution**(b):
2. return $$\frac{\sqrt{6}}{2}\left[\mathbf{rand}(-b, b) + \mathbf{rand}(-b, b)\right]$$

Normally Distributed Samples

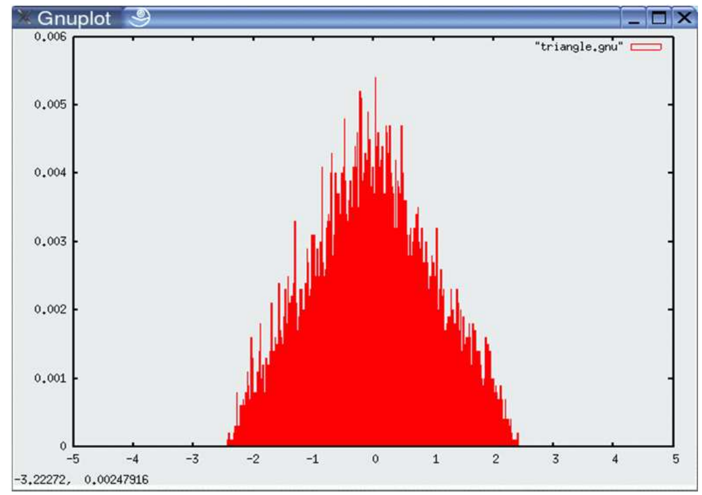


10^6 samples

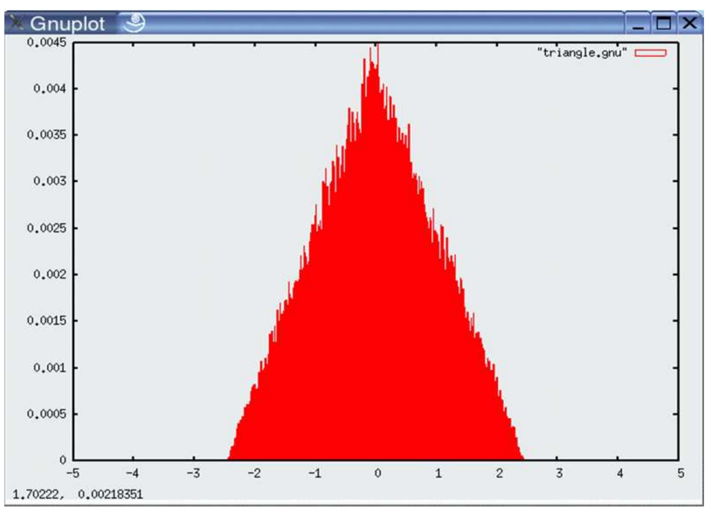
For Triangular Distribution



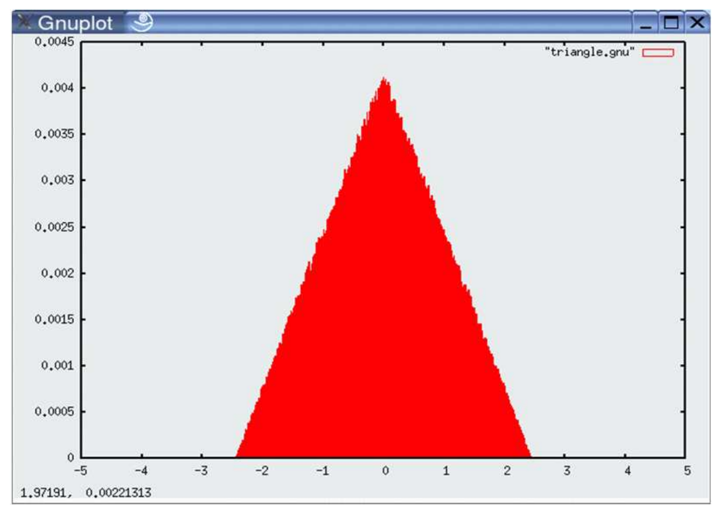
10^3 samples



10^4 samples



10^5 samples



10^6 samples

Rejection Sampling

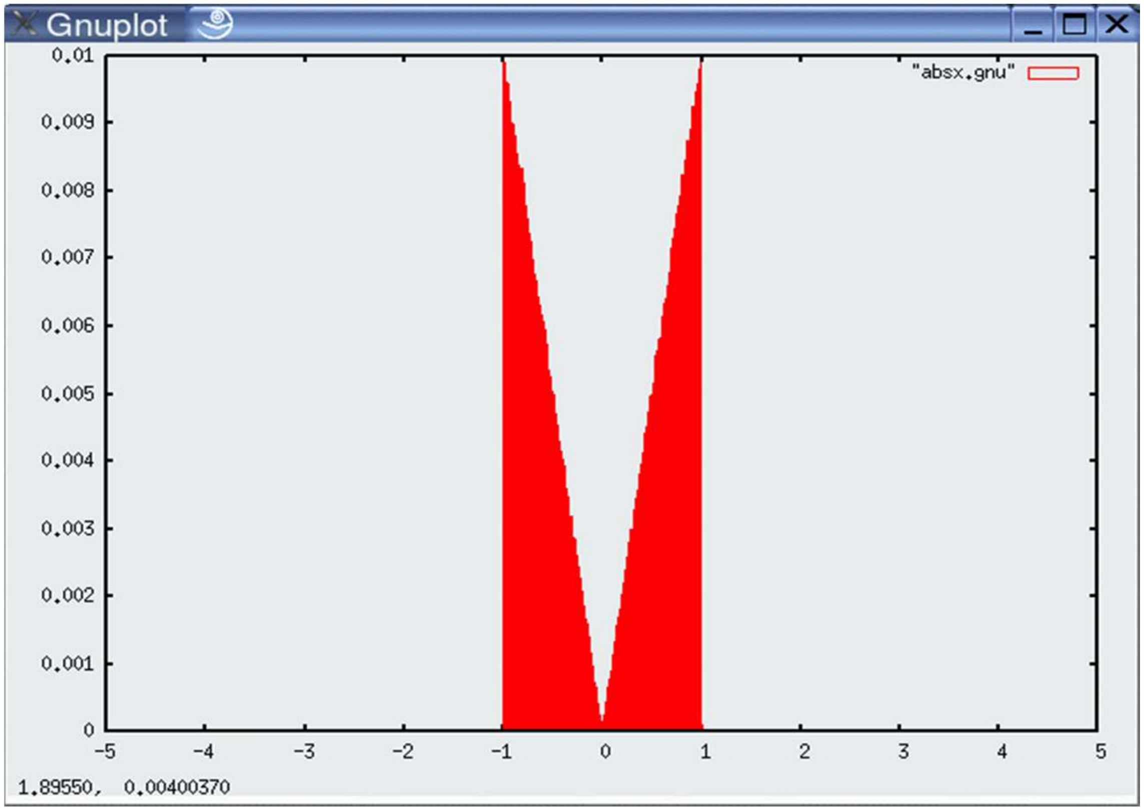
- Sampling from arbitrary distributions

Algorithm **sample_distribution**(f,b):
repeat
    $x = \mathbf{rand}(-b, b)$
    $y = \mathbf{rand}(0, \max\{f(x) \mid x \in (-b, b)\})$
until ($y \leq f(x)$
return x

Example

- Sampling from

$$f(x) = \begin{cases} \mathsf{abs}(x) & x \in [-1;1] \\ 0 & \mathsf{otherwise} \end{cases}$$



Sample Odometry Motion Model

1. Algorithm **sample_motion_model**(u, x):

$$u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$$

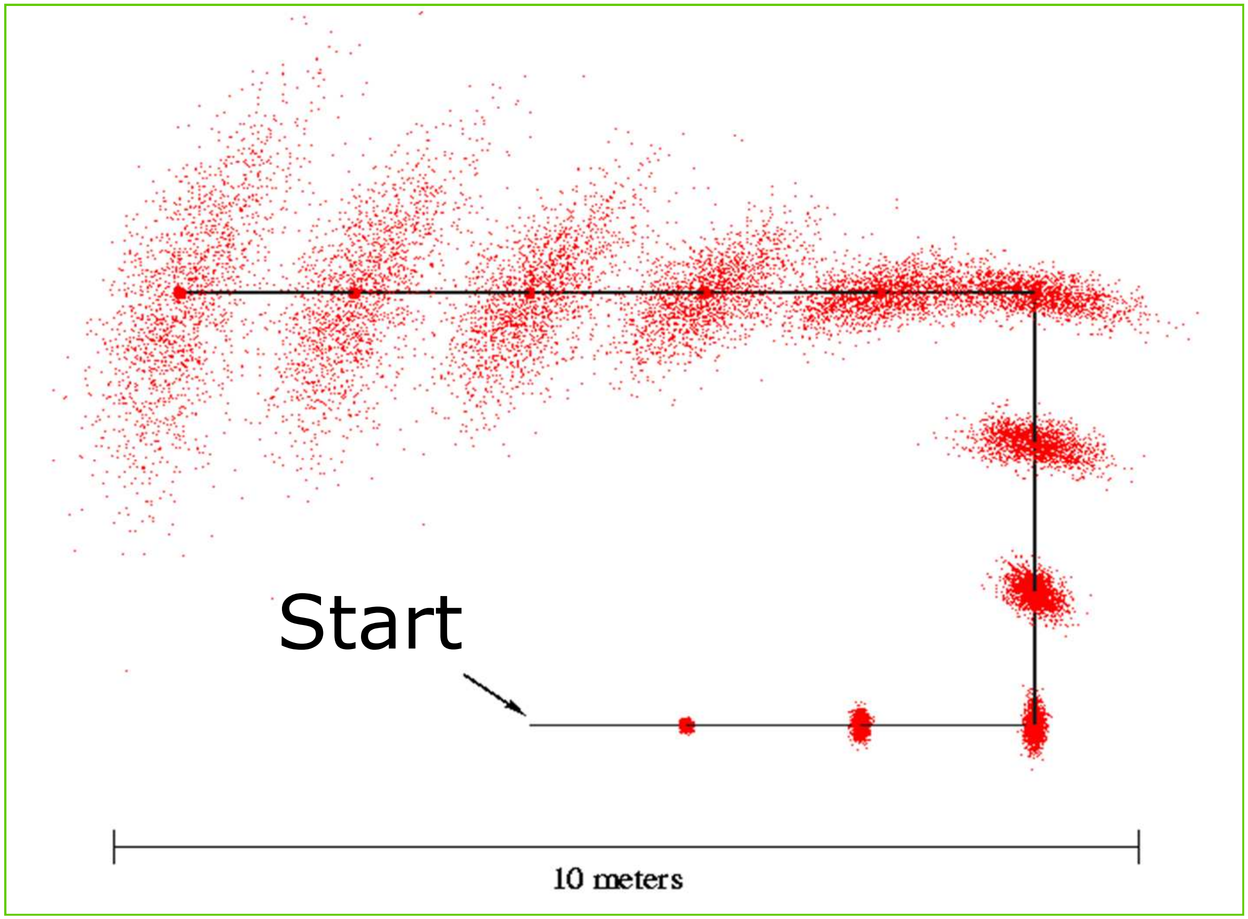
1. $\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 |\delta_{rot1}| + \alpha_2\ \delta_{trans})$
2. $\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_3\ \delta_{trans} + \alpha_4\ (|\delta_{rot1}| + |\delta_{rot2}|))$
3. $\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 |\delta_{rot2}| + \alpha_2\ \delta_{trans})$

4. $x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$
5. $y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$
6. $\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$

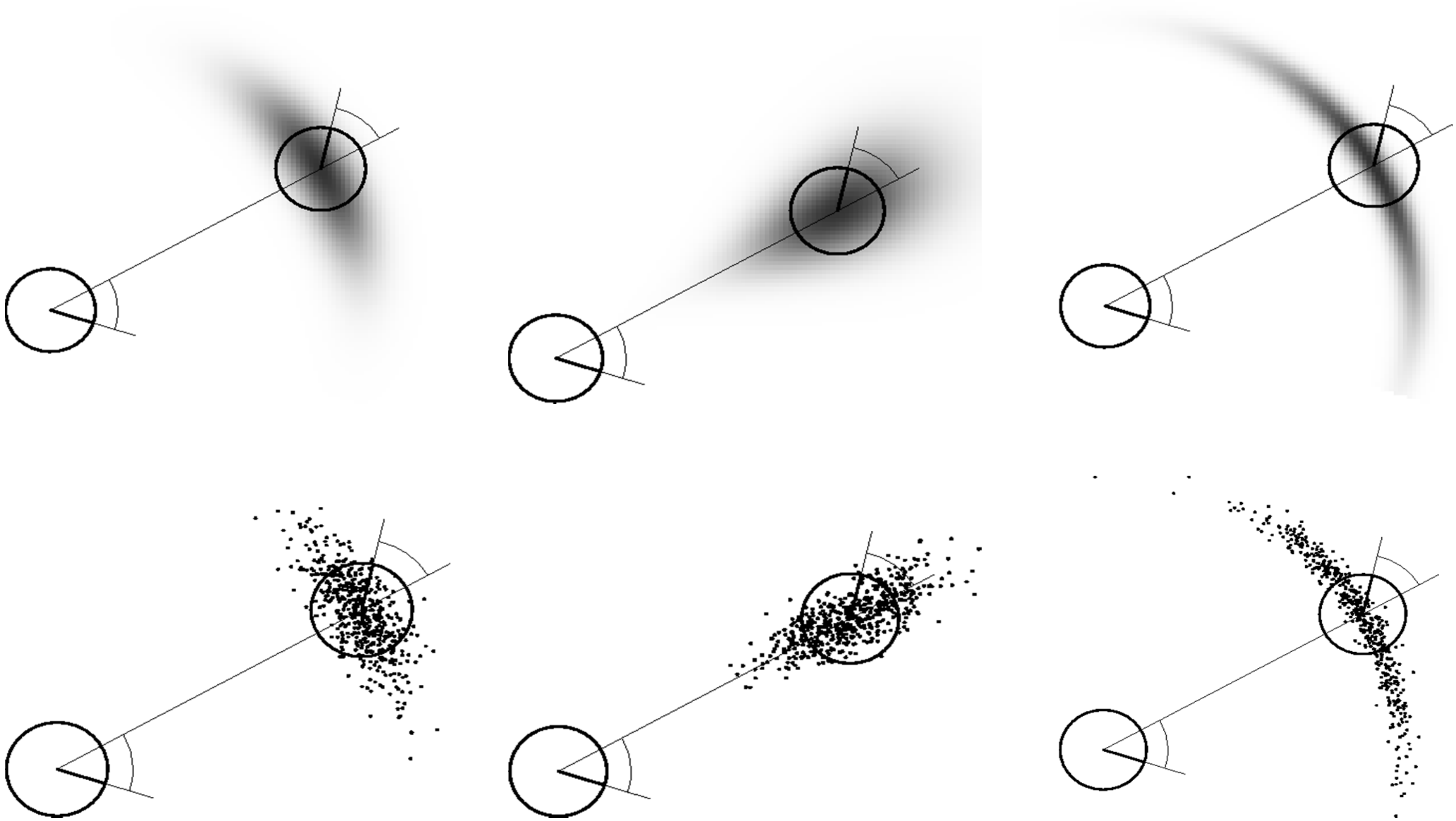
7. Return $\langle x', y', \theta' \rangle$

sample_normal_distribution

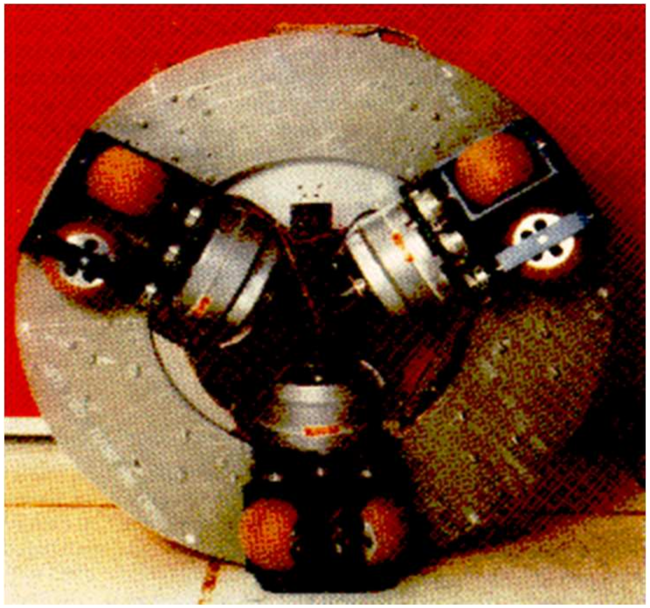
Sampling from Our Motion Model



Examples (Odometry-Based)



Holonomic reality

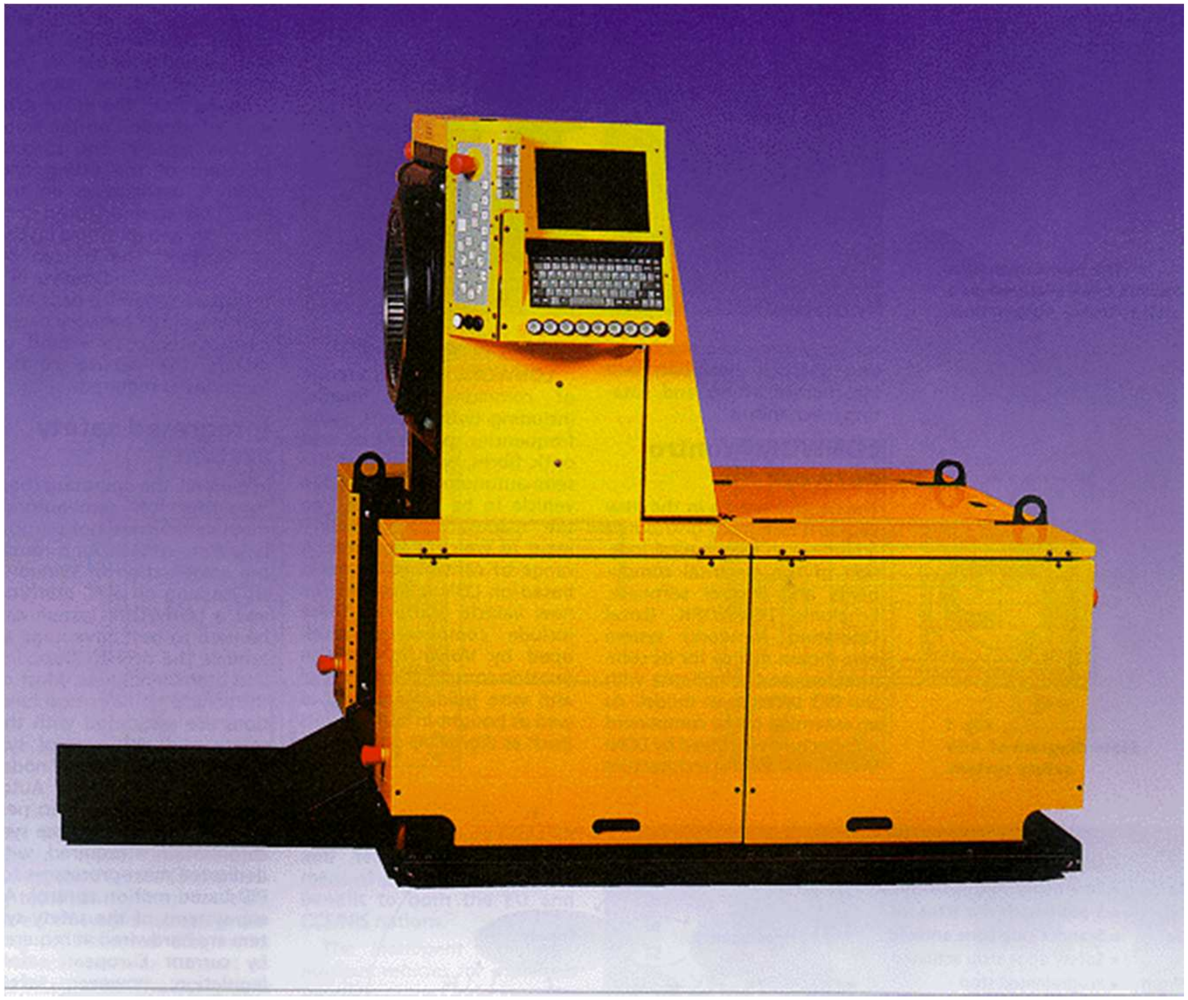


Discover Magazine -- Top 10 Innovation

Sage -- a museum tour guide

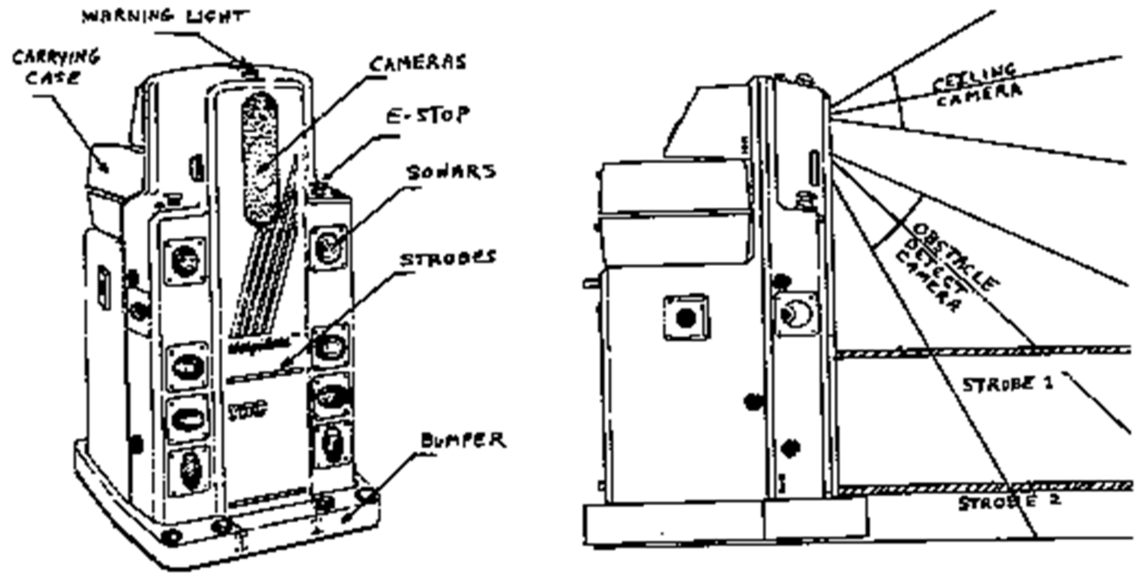
easier to define than to determine…

Mobile Robot Examples
Automatic Guided Vehicles



- Newest generation of Automatic Guided Vehicle of VOLVO used to transport motor blocks from on assembly station to an other. It is guided by an electrical wire installed in the floor but it is also able to leave the wire to avoid obstacles. There are over 4000 AGV only at VOLVO’s plants.

Helpmate



- HELPMATE is a mobile robot used in hospitals for transportation tasks. It has various on board sensors for autonomous navigation in the corridors. The main sensor for localization is a camera looking to the ceiling. It can detect the lamps on the ceiling as reference (landmark). http://www.ntplx.net/~helpmate/

BR700 Cleaning Robot

- BR 700 cleaning robot developed and sold by Kärcher Inc., Germany. Its navigation system is based on a very sophisticated sonar system and a gyro. http://www.kaercher.de

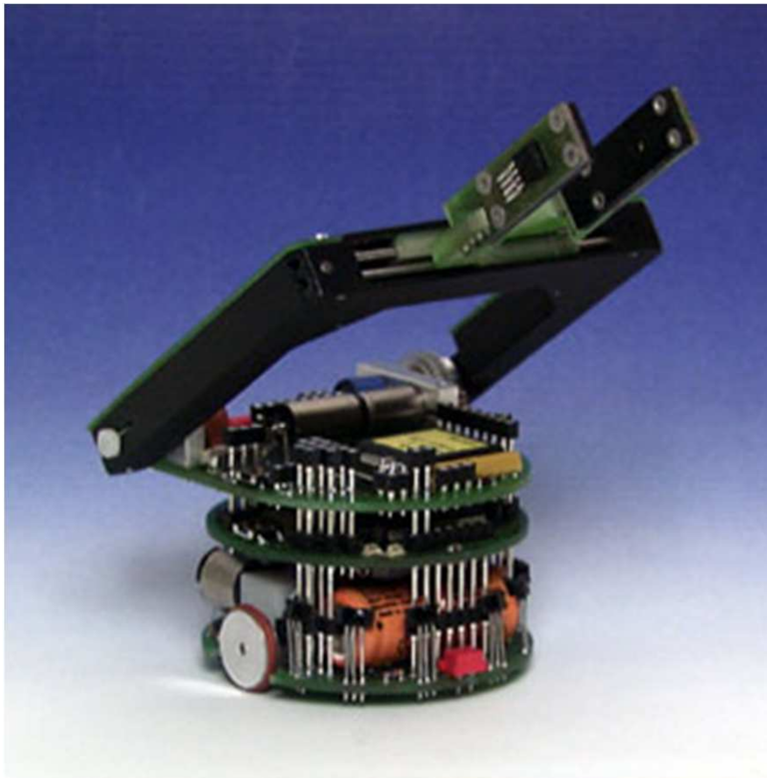
The Pioneer

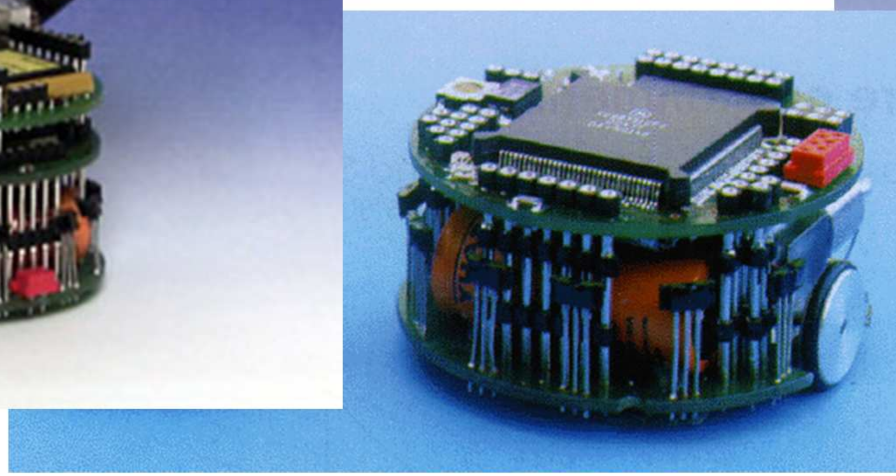
- Picture of Pioneer, the teleoperated robot that is supposed to explore the Sarcophagus at Chernobyl

The Pioneer

- PIONEER 1 is a modular mobile robot offering various options like a gripper or an on board camera. It is equipped with a sophisticated navigation library developed at Stanford Research Institute (SRI). http://www.activmedia.com/robots

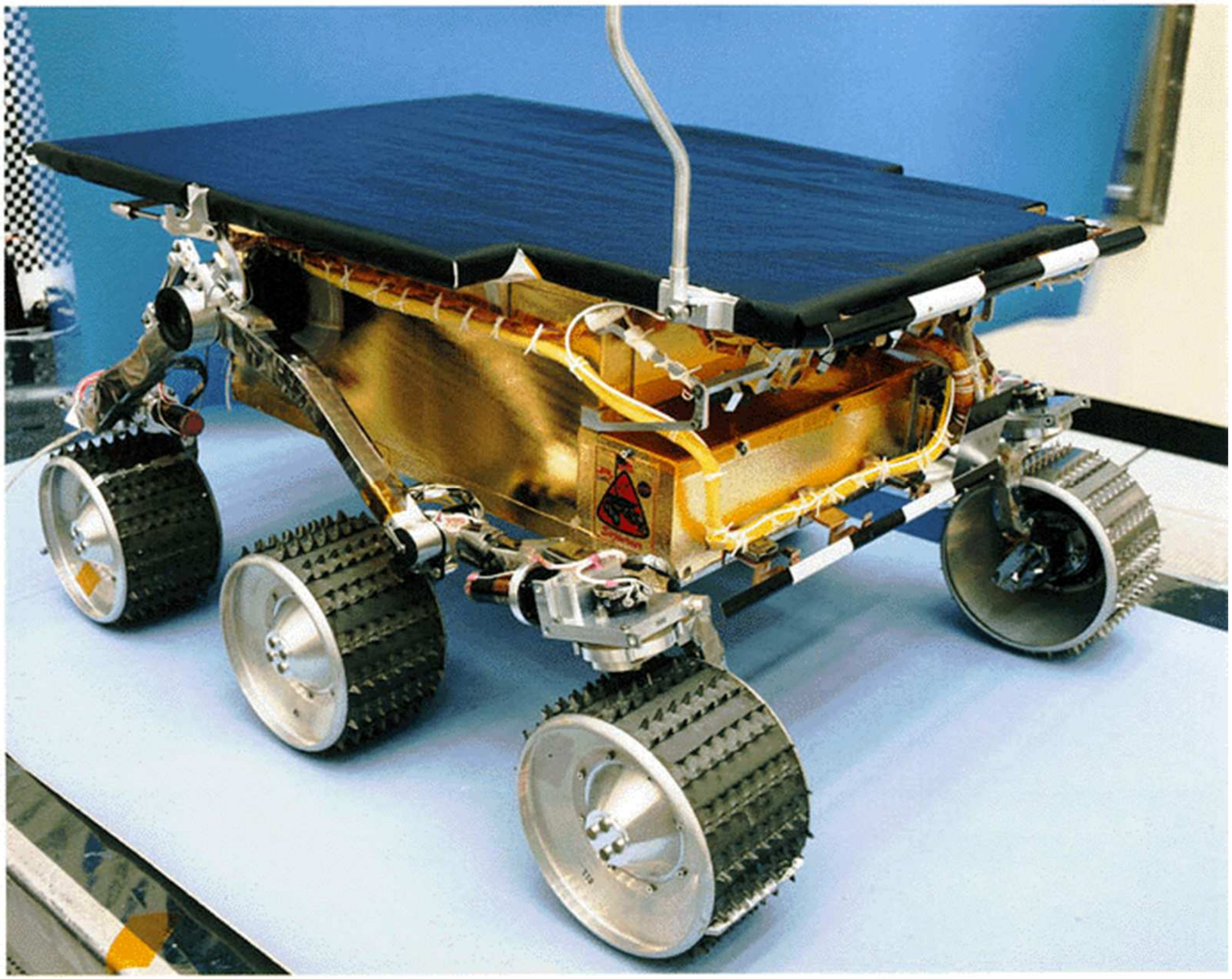
The Khepera Robot





- KHEPERA is a small mobile robot for research and education. It sizes only about 60 mm in diameter. Additional modules with cameras, grippers and much more are available. More then 700 units have already been sold (end of 1998). http://diwww.epfl.ch/lami/robots/K-family/ K-Team.html

Sojourner, First Robot on Mars

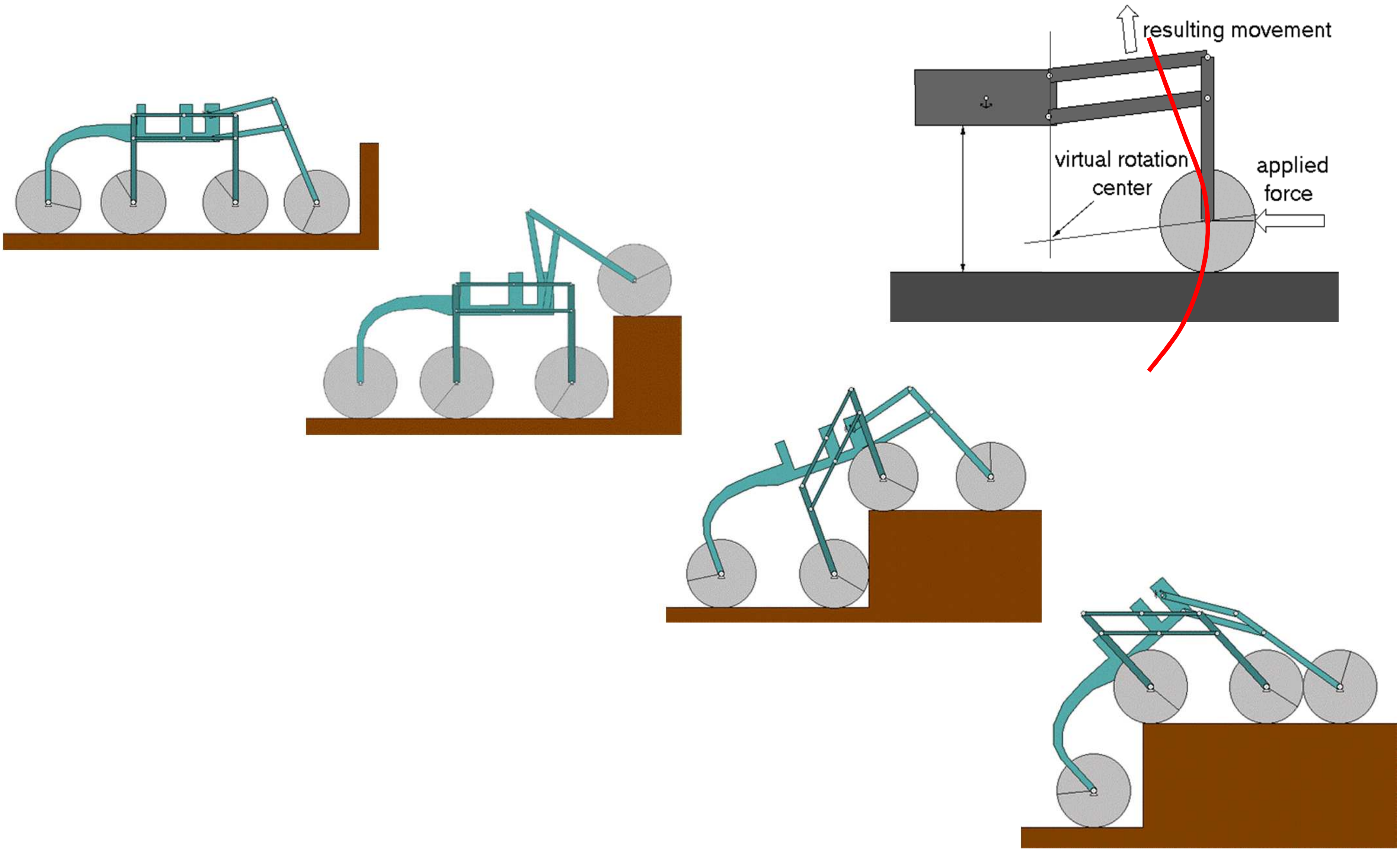


- The mobile robot Sojourner was used during the Pathfinder mission to explore the mars in summer 1997. It was nearly fully teleoperated from earth. However, some on board sensors allowed for obstacle detection. http://ranier.oact.hq.nasa.gov/telerobotics_page/telerobotics.shtm

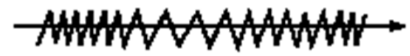
SHRIMP, a Mobile Robot with Excellent Climbing Abilities

- Objective
  - *Passive locomotion concept for rough terrain*
- Results: The Shrimp
  - *6 wheels*
    - *one fixed wheel in the rear*
    - *two boogies on each side*
    - *one front wheel with spring suspension*
  - *robot sizing around 60 cm in length and 20 cm in height*
  - *highly stable in rough terrain*
  - *overcomes obstacles up to 2 times its wheel diameter*

The SHRIMP Adapts Optimally to Rough Terrain

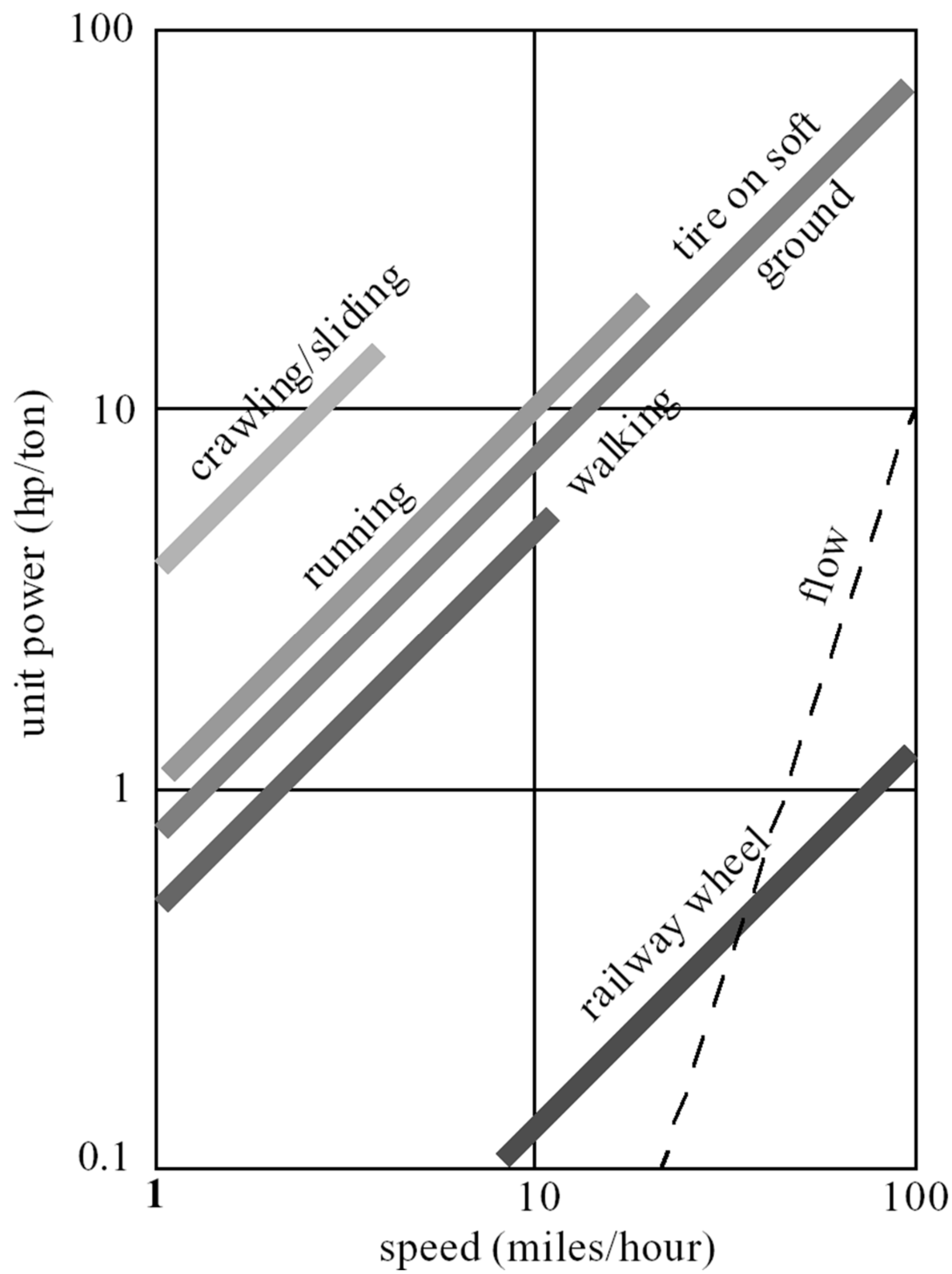


Locomotion Concepts: Principles Found in Nature

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) 

Walking or rolling?

- number of actuators
- structural complexity
- control expense
- energy efficient
  - *terrain (flat ground, soft ground, climbing..)*
- movement of the involved masses
  - *walking / running includes up and down movement of COG*
  - *some extra losses*

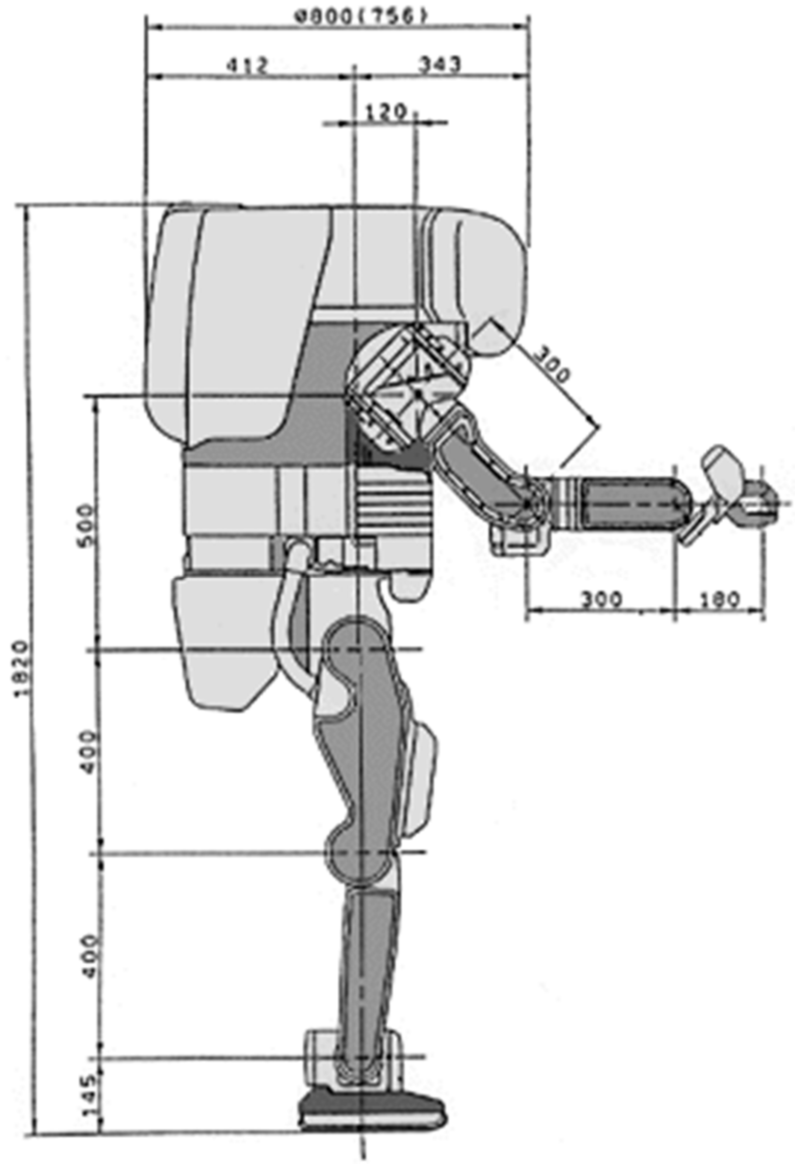


Forester Robot

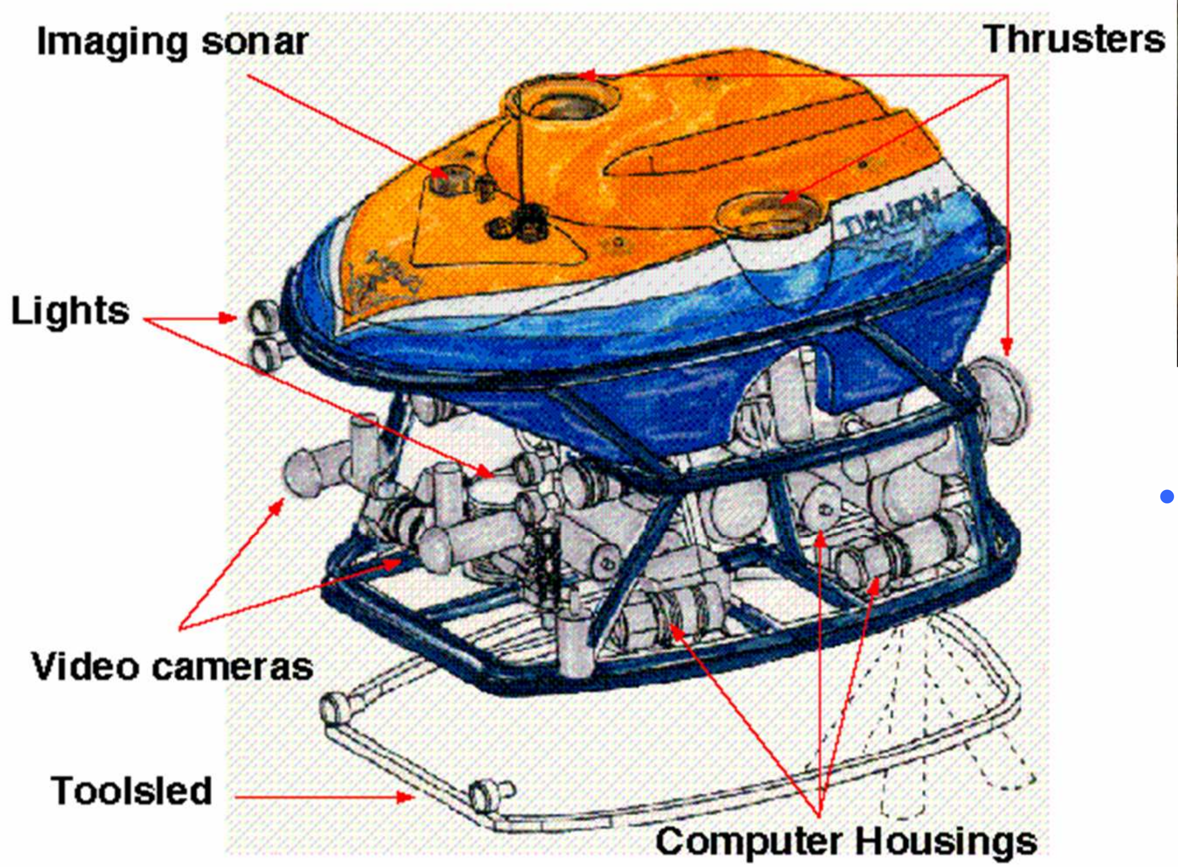
- Pulstech developed the first ‘industrial like’ walking robot. It is designed moving wood out of the forest. The leg coordination is automated, but navigation is still done by the human operator on the robot. http://www.plustech.fi/

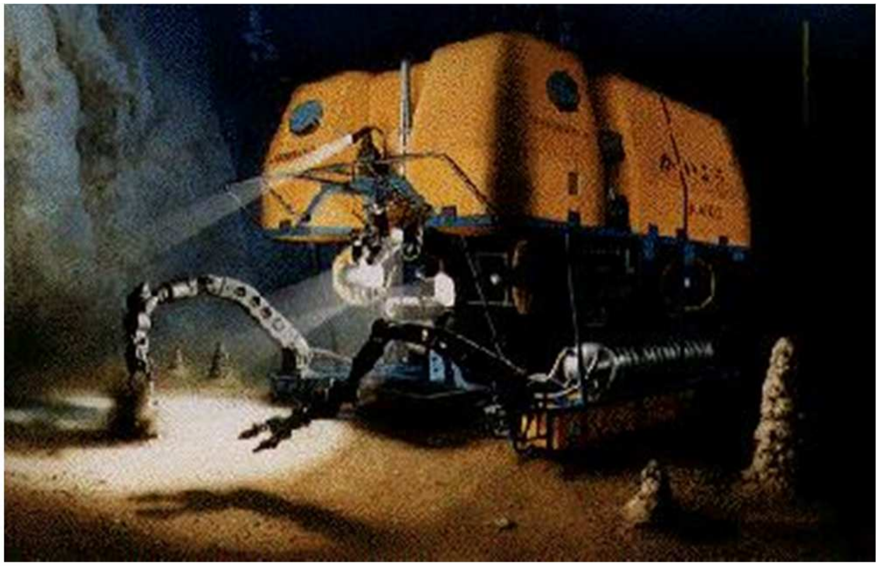
The Honda Walking Robot *http://www.honda.co.jp/tech/other/robot.html*

Image of Honda Robot



ROV Tiburon Underwater Robot





- Picture of robot ROV Tiburon for underwater archaeology (teleoperated)- used by MBARI for deep-sea research, this UAV provides autonomous hovering capabilities for the human operator.

Kinematics lives!

C. DiLeo and **X. Deng***, “Dragonfly Robots: Design, Kinematics and Force Measurements ”,
Journal of Advanced Robotics, May, 2009.

i.e., designing winged robots

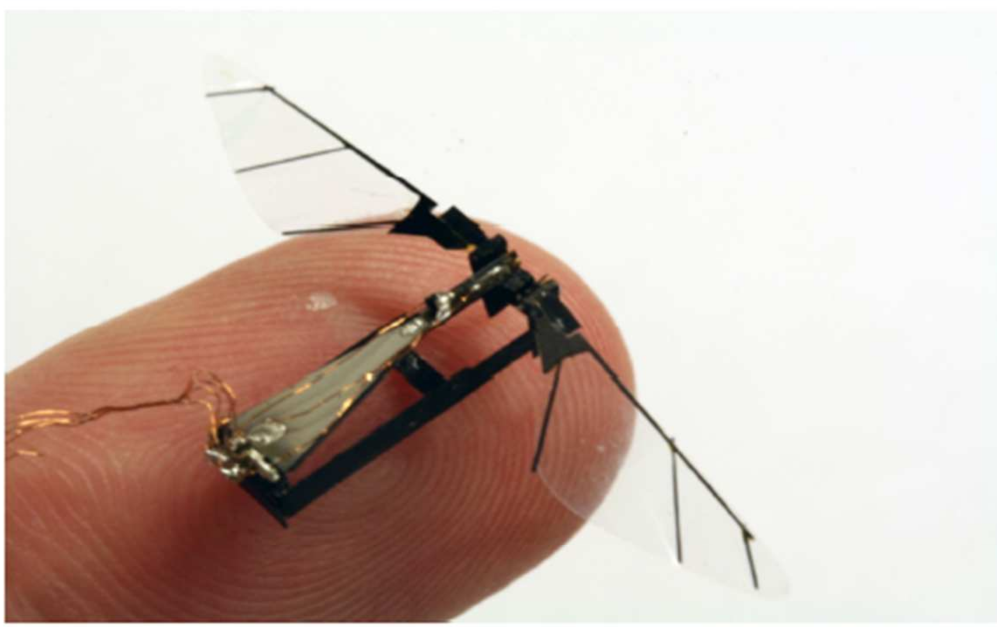
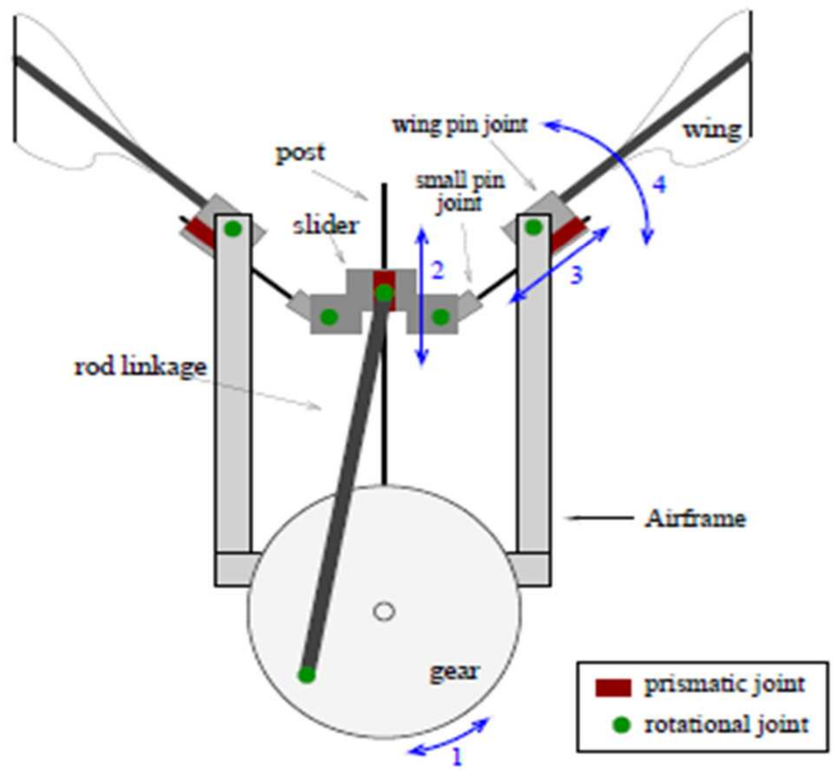


Fig. 18. A 3D CAD model of a flightweight design utilizing the hybrid actuator structure (top), and a next-generation prototype (bottom).

Harvard’s Flying Microrobots



Univ. of Delaware

Kinematics lives!

Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Inverse Kinematics of Gel Robots made of Electro-Active Polymer Gel

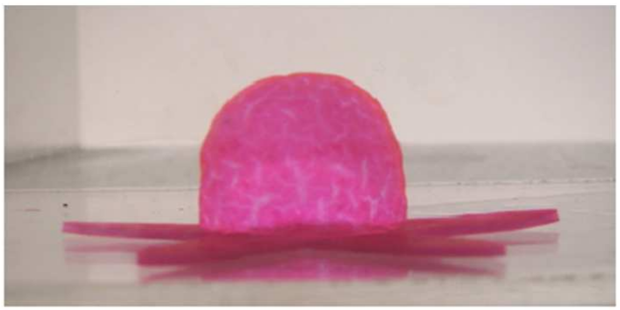
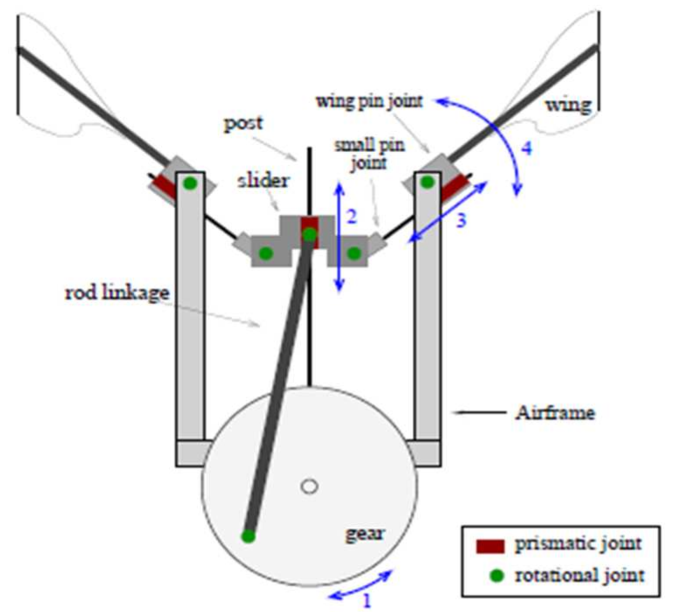


Figure 1: Prototype of octopus-shaped gel robot

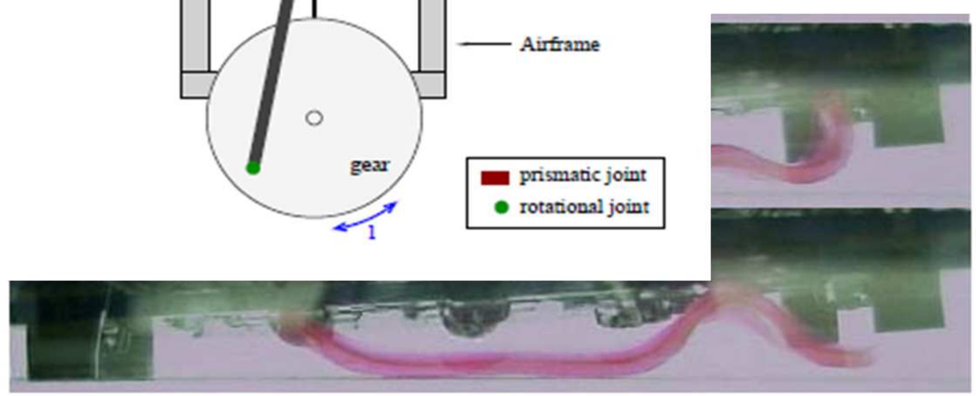
i.e., getting a robotic octopus to move along a path

A single tentacle…



Modeling interactions

The result



Robot Manipulators

Is this robot holonomic ?