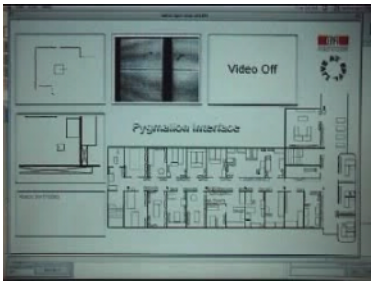
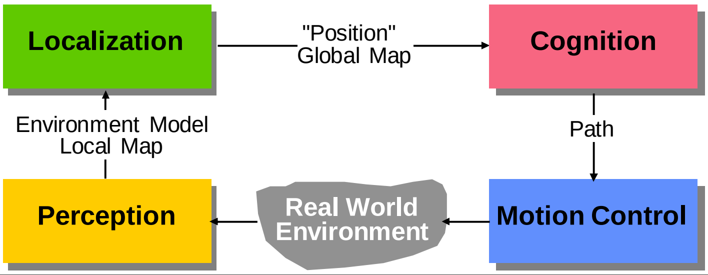
Motion Control (wheeled robots)

- Requirements for Motion Control
  - *Kinematic / dynamic model of the robot*
  - *Model of the interaction between the wheel and the ground*
  - *Definition of required motion -> speed control, position control*
  - *Control law that satisfies the requirements*





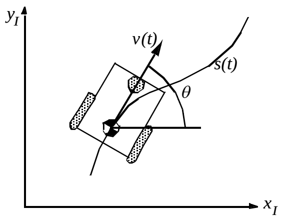
Introduction: Mobile Robot Kinematics (position and velocity)

- Aim
  - *Description of mechanical behavior of the robot for* design *and* control
  - *Similar to robot manipulator kinematics*
  - *However, mobile robots can move unbound with respect to its environment*
    - *there is no direct way to measure the robot's position*
    - *Position must be integrated over time*
    - *Leads to inaccuracies of the position (motion) estimate*
      -> the number 1 challenge in mobile robotics
  - *Understanding mobile robot motion starts with understanding wheel* constraints *placed on the robots mobility*

Introduction: Kinematics Model

- Goal:
  - *establish the robot speed* $\dot{\xi} = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$ *as a function of the wheel speeds* $\dot{\varphi}_i$, *steering angles* β_i, *steering speeds* $\dot{\beta}_i$ *and the geometric parameters of the robot (configuration coordinates).*
  - *forward kinematics*

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_1, \ldots \dot{\varphi}_n, \beta_1, \ldots \beta_m, \dot{\beta}_1, \ldots \dot{\beta}_m)$$



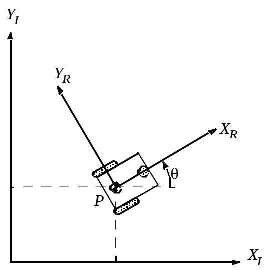
  - *Inverse kinematics*

$$[\dot{\varphi}_1 \quad \cdots \quad \dot{\varphi}_n \quad \beta_1 \quad \ldots \quad \beta_m \quad \dot{\beta}_1 \quad \ldots \quad \dot{\beta}_m]^T = f(\dot{x}, \dot{y}, \dot{\theta})$$

  - *why not* $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = f(\varphi_1, \ldots \varphi_n, \beta_1, \ldots \beta_m)$ -> ***not straight forward***

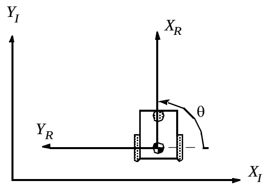
Representing Robot Position

- Representing to robot within an arbitrary initial frame
  - *Initial frame:* $\{X_I, Y_I\}$
  - *Robot frame:* $\{X_R, Y_R\}$
  - *Robot position:* $\xi_I = [x \quad y \quad \theta]^T$
  - *Mapping between the two frames*
  - $\dot{\xi}_R = R(\theta)\dot{\xi}_I = R(\theta) \cdot [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$



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

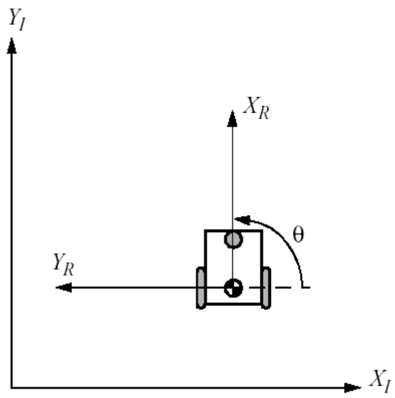
  - *Example: Robot aligned with* Y_I



3.2.1

Example

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



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

Robot Motion models

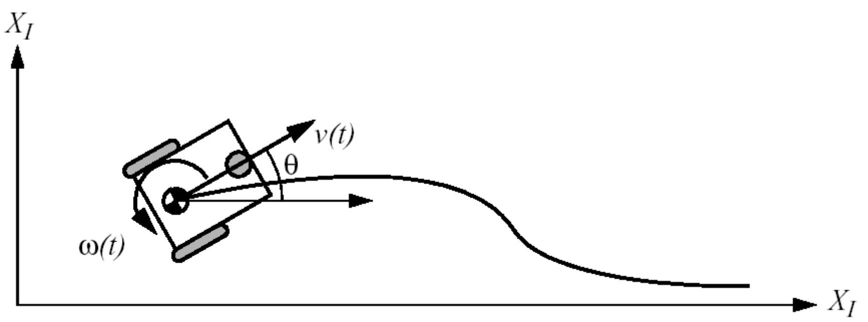
- Velocity motion model
  - *the simplest one, assumes y p , that the control is given as a velocity command to the motors; velocity remain constant in the sampling interval [t-1, t)*
- Odometry motion model
  - *assumes the accessibility to odometric information, usually provided by wheel sensors, but often also by other means (i.e., visual odometry)*
  - *Odometric models are usually more accurate than velocity models, but odometry is available only after the motion command has been executed, while velocity commands are available before performing the actual motion*
  - *Odometric models are good for estimation, while velocity models are better suited for path planning*

Odometry

- Differential Drive robot (Wang paper)
  - *Uncertainty: pose (x, y, θ), Δd, Δθ*
- Differential Drive robot (Textbook)
  - *Uncertainty: pose (x, y, θ), Δr, Δl)*
- Differential Drive robot (Exercise)
  - *Uncertainty: pose (x, y, θ), Δr, Δl, b)*
- Steer Drive robot (Exercise)
  - *Uncertainty: pose (x, y, θ), Δv, Δα, T)*

Odometry: The Differential Drive Robot (1)

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}$$



Odometry: The Differential Drive Robot (2)

- Kinematics

$$\Delta x = \Delta s\cos(\theta + \Delta\theta/2)$$

$$\Delta y = \Delta s\sin(\theta + \Delta\theta/2)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2}\cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2}\sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

Odometry: The Differential Drive Robot (3)

- Error model

$$\Sigma_\Delta = covar(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix}$$

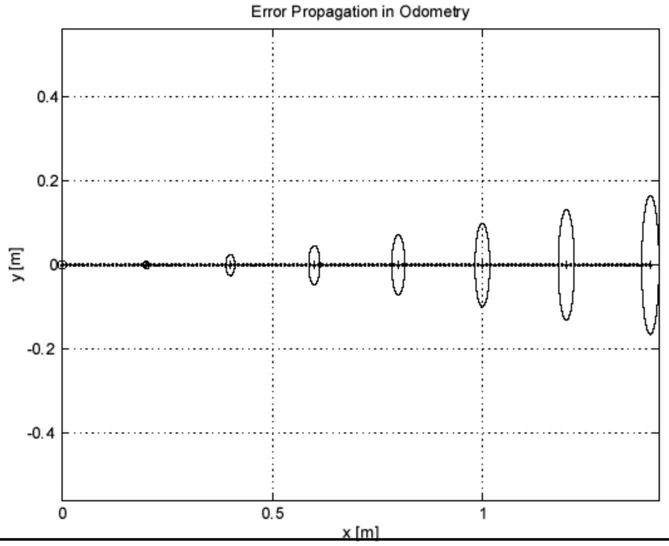
$$\Sigma_{p'} = \nabla_p f \cdot \Sigma_p \cdot \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \Sigma_\Delta \cdot \nabla_{\Delta_{rl}} f^T$$

$$F_p = \nabla_p f = \nabla_p(f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s\sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s\cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$F_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$

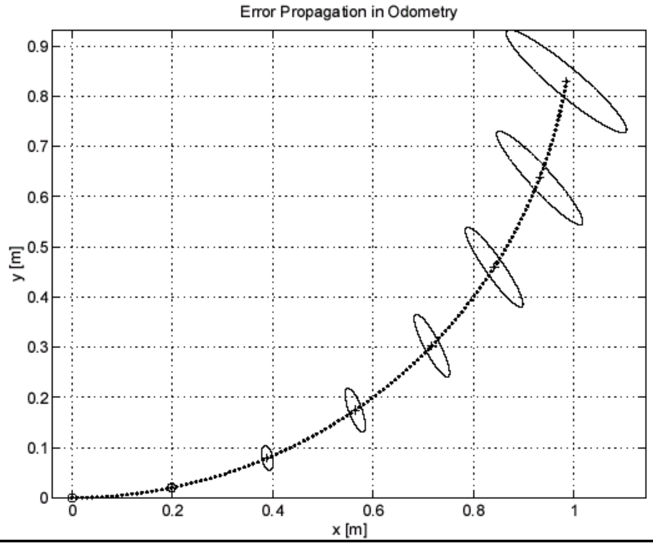
Odometry: Growth of Pose uncertainty for Straight Line Movement

- Note: Errors perpendicular to the direction of movement are growing much faster!



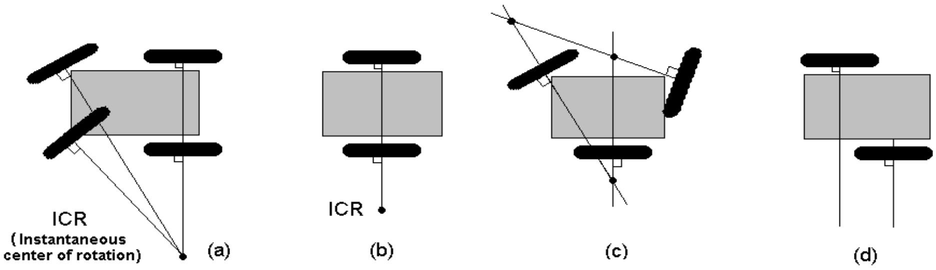
Odometry: Growth of Pose uncertainty for Movement on a Circle

- Note: Errors ellipse in does not remain perpendicular to the direction of movement!



Mobile Robot Locomotion

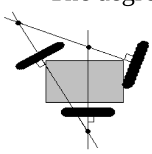
- Instantaneous center of rotation (ICR) or Instantaneous center of curvature (ICC)
  - A cross point of all axes of the wheels



Degree of Mobility

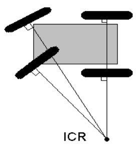
- **Degree of mobility**

  The degree of freedom of the robot motion



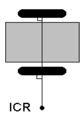
Cannot move anywhere
(No ICR)

- Degree of mobility : 0



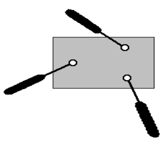
Fixed arc motion
(Only one ICR)

- Degree of mobility : 1



Variable arc motion
(line of ICRs)

- Degree of mobility : 2



Fully free motion

(ICR can be located at any position)

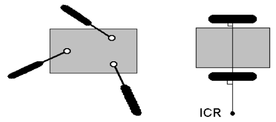
- Degree of mobility : 3

The *degree of mobility* qualities the degrees of controllable freedom based on **changes to wheel velocities**!

Degree of Steerability

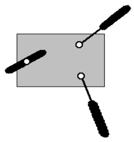
- **Degree of steerability**

  The number of centered orientable wheels that can be steered independently in order to steer the robot

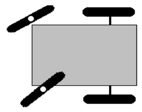


No centered orientable wheels

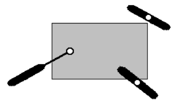
- Degree of steerability : 0



One centered orientable wheel



Two mutually dependent centered orientable wheels



Two mutually independent centered orientable wheels

- Degree of steerability : 1
- Degree of steerability : 2

15

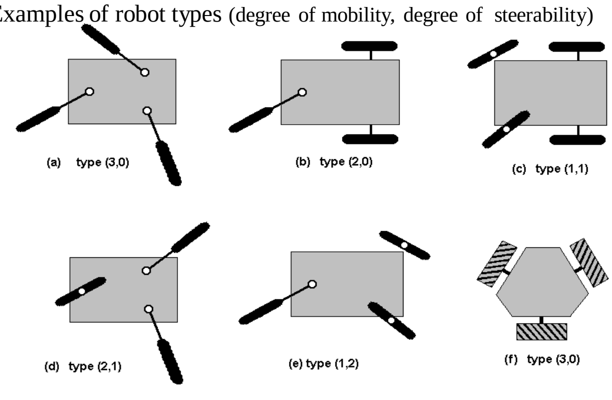
Degree of Maneuverability

- The overall degrees of freedom that a robot can manipulate:

$$\delta_M = \delta_m + \delta_s$$

Degree of Mobility	3	2	2	1	1
Degree of Steerability	0	0	1	1	2

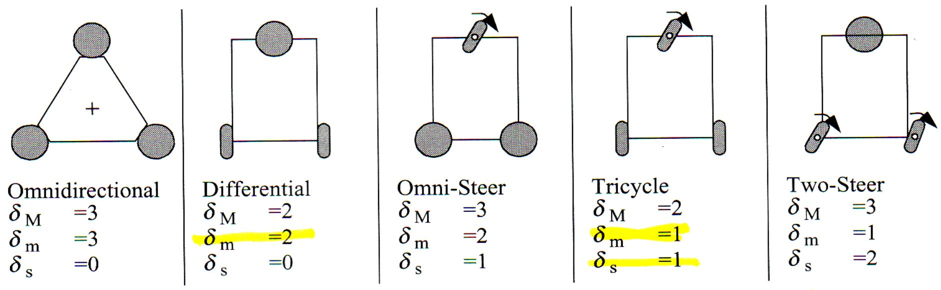
- Examples of robot types (degree of mobility, degree of steerability)



16

Degree of Maneuverability

$$\delta_M = \delta_m + \delta_s$$



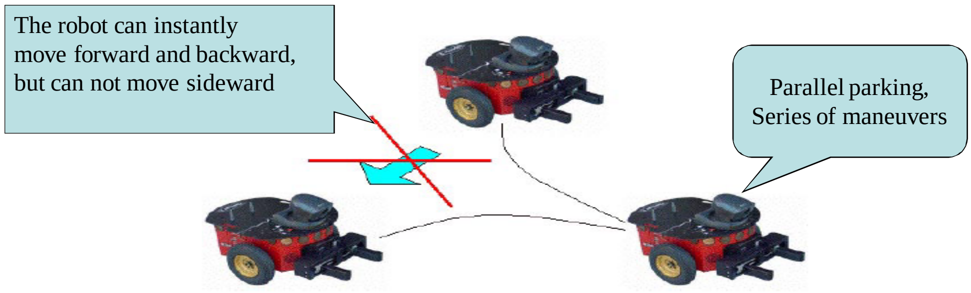
17

Non-holonomic constraint

A non-holonomic constraint is a constraint on the feasible **velocities** of a body

So what does that mean?

Your robot can move in some directions (forward and backward), but not others (sideward).



18

19