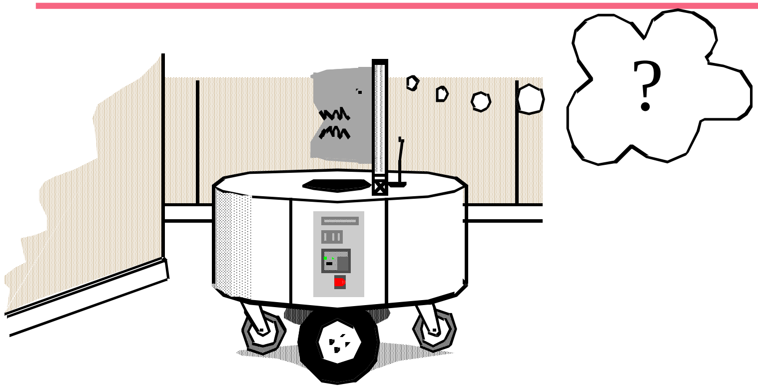
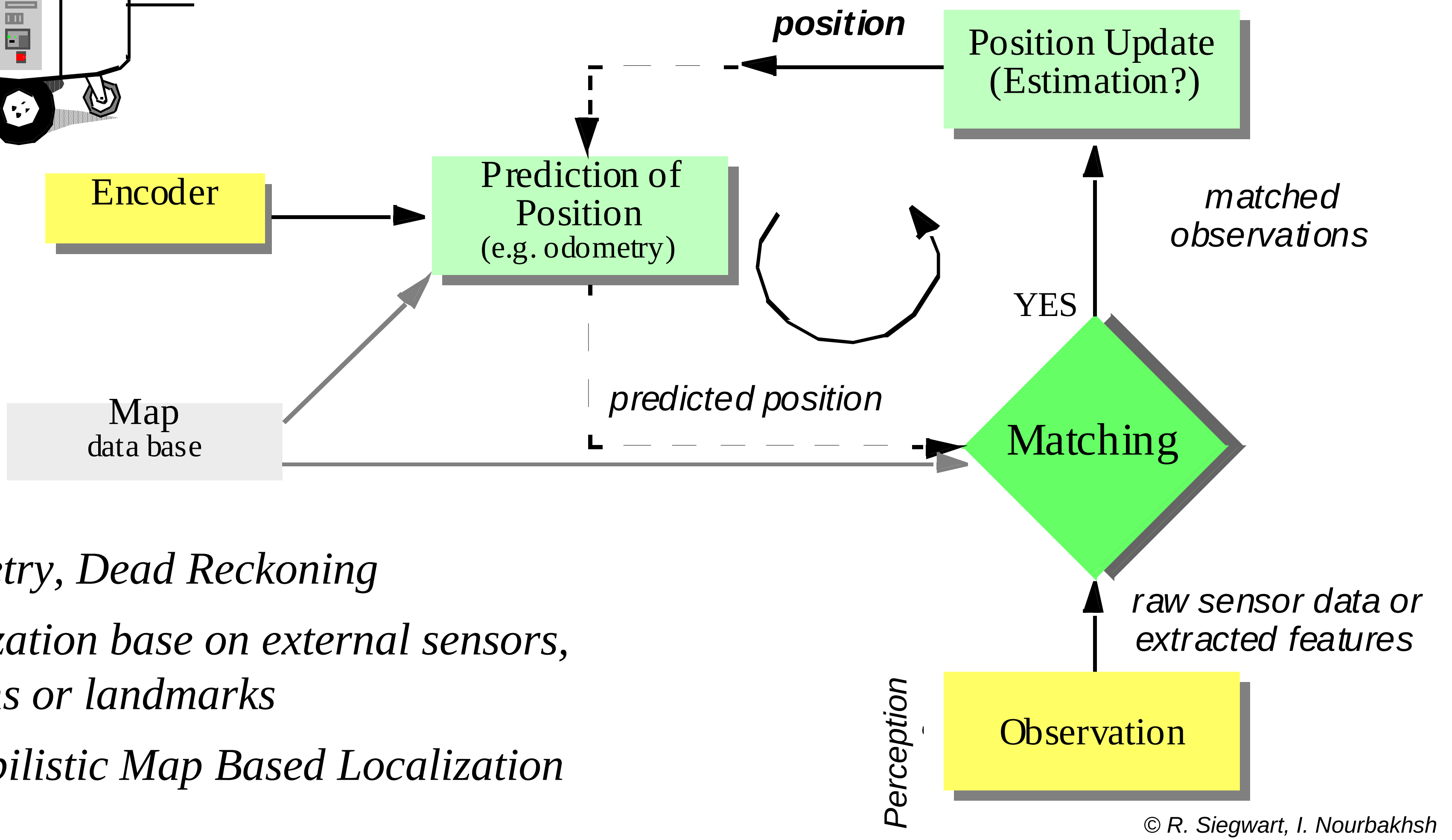
Localization, Where am I?





- *Odometry, Dead Reckoning*
- *Localization base on external sensors, beacons or landmarks*
- *Probabilistic Map Based Localization*

Challenges of Localization

- Knowing the absolute position (e.g. GPS) is not sufficient
- Localization in human-scale in relation with environment
- Planning in the *Cognition* step requires more than only position as input
- Perception and motion plays an important role
  - *Sensor noise*
  - *Sensor aliasing*
  - *Effector noise*
  - *Odometric position estimation*

Sensor Noise

- Sensor noise is mainly influenced by environment
  e.g. surface, illumination …

- or by the measurement principle itself
  e.g. interference between ultrasonic sensors

- Sensor noise drastically reduces the useful information of sensor readings. The solution is:
  - *to take multiple reading into account*
  - *employ temporal and/or multi-sensor fusion*

Sensor Aliasing

- In robots, non-uniqueness of sensors readings is the norm
- Even with multiple sensors, there is a many-to-one mapping from environmental states to robot's perceptual inputs
- Therefore the amount of information perceived by the sensors is generally insufficient to identify the robot's position from a single reading
  - *Robot's localization is usually based on a series of readings*
  - *Sufficient information is recovered by the robot over time*

Effector Noise: Odometry, Dead Reckoning

- Odometry and dead reckoning:
  Position update is based on proprioceptive sensors
  - *Odometry: wheel sensors only*
  - *Dead reckoning: also heading sensors*
- The movement of the robot, sensed with wheel encoders and/or heading sensors is integrated to the position.
  - *Pros: Straight forward, easy*
  - *Cons: Errors are integrated -> unbounded*
- Using additional heading sensors (e.g. gyroscope) might help to reduce the accumulated errors, but the main problems remain the same.

Odometry: Error sources

deterministic (systematic) $\longleftrightarrow$ non-deterministic (non-systematic)

- *deterministic errors can be eliminated by proper calibration of the system.*
- *non-deterministic errors have to be described by error models and will always lead to uncertain position estimate.*

- Major Error Sources:
  - *Limited resolution during integration (time increments, measurement resolution …)*
  - *Misalignment of the wheels (deterministic)*
  - *Unequal wheel diameter (deterministic)*
  - *Variation in the contact point of the wheel*
  - *Unequal floor contact (slipping, non-planar …)*
  - *…*

Odometry: Classification of Integration Errors

- Range error: integrated path length (distance) of the robots movement
  - *sum of the wheel movements*
- Turn error: similar to range error, but for turns
  - *difference of the wheel motions*
- Drift error: difference in the error of the wheels leads to an error in the robot's angular orientation

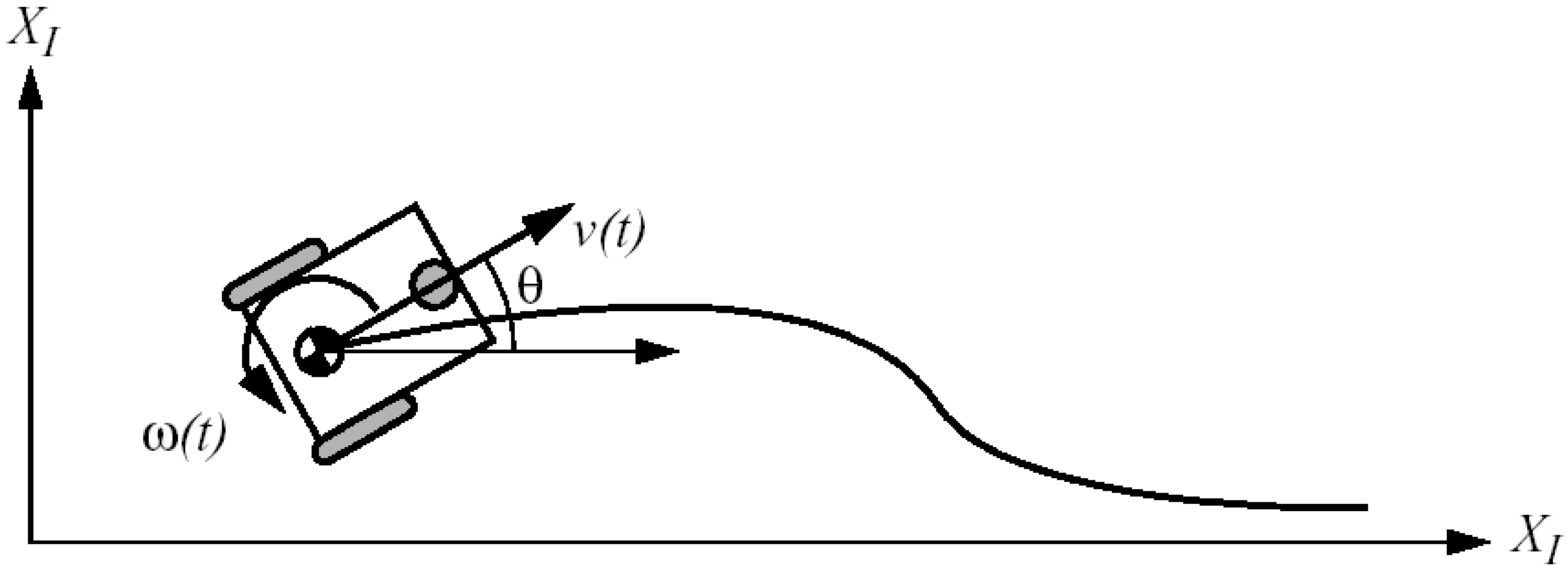
Over long periods of time, turn and drift errors

far outweigh range errors!

  - *Consider moving forward on a straight line along the* **x** *axis. The error in the* **y***-position introduced by a move of* **d** *meters will have a component of* $\mathbf{d}\sin\Delta\theta$*, which can be quite large as the angular error* $\Delta\theta$ *grows.*

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

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

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

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

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

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

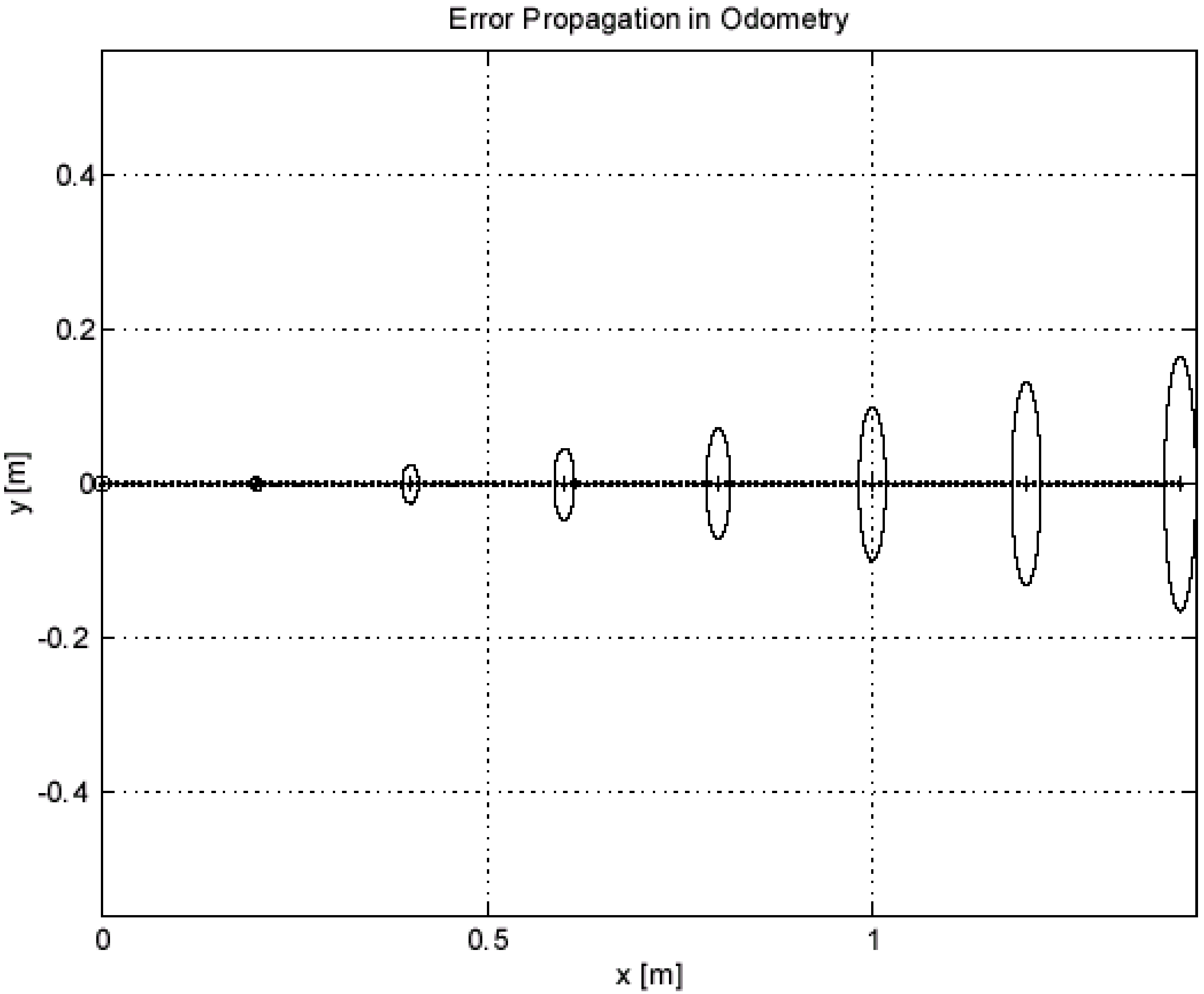
Odometry: The Differential Drive Robot (3)

- Error model (details are beyond the scope of our class; just know that we can build an error model…)

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

Odometry: Growth of Pose uncertainty for Straight Line Movement

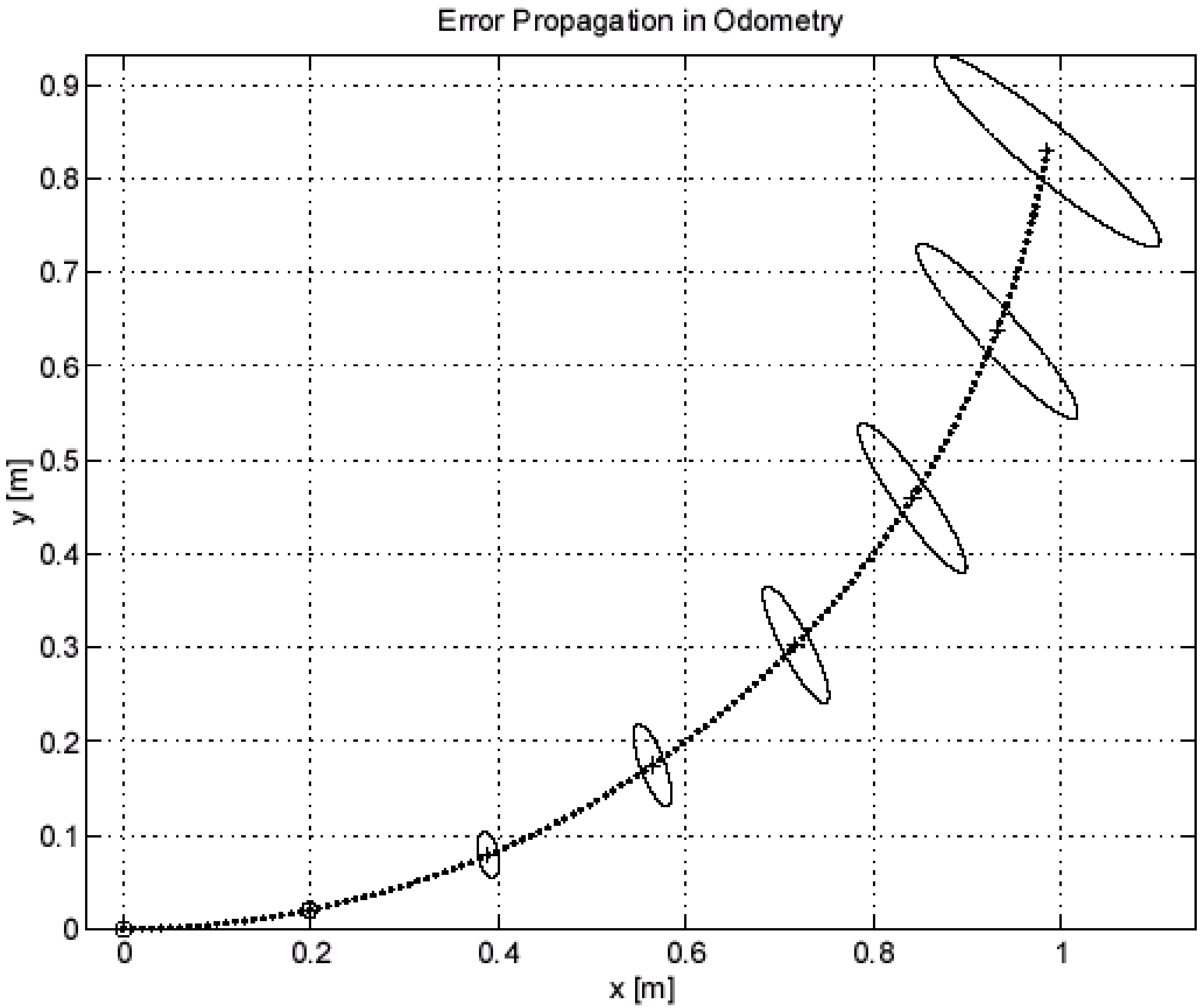
- Note: Errors perpendicular to the direction of movement are growing much faster!



(ellipses represent uncertainty in position)

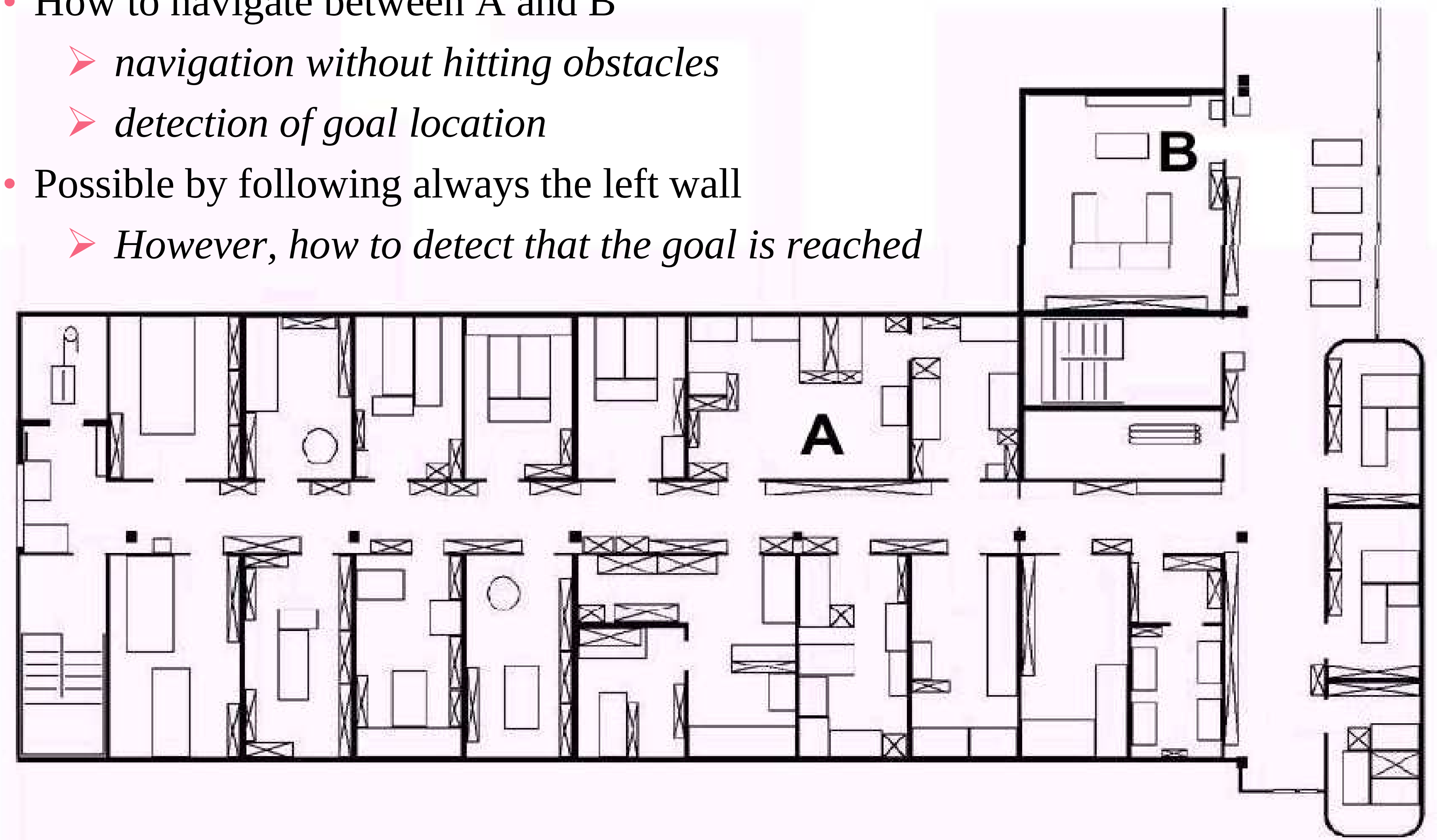
Odometry: Growth of Pose uncertainty for Movement on a Circle

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

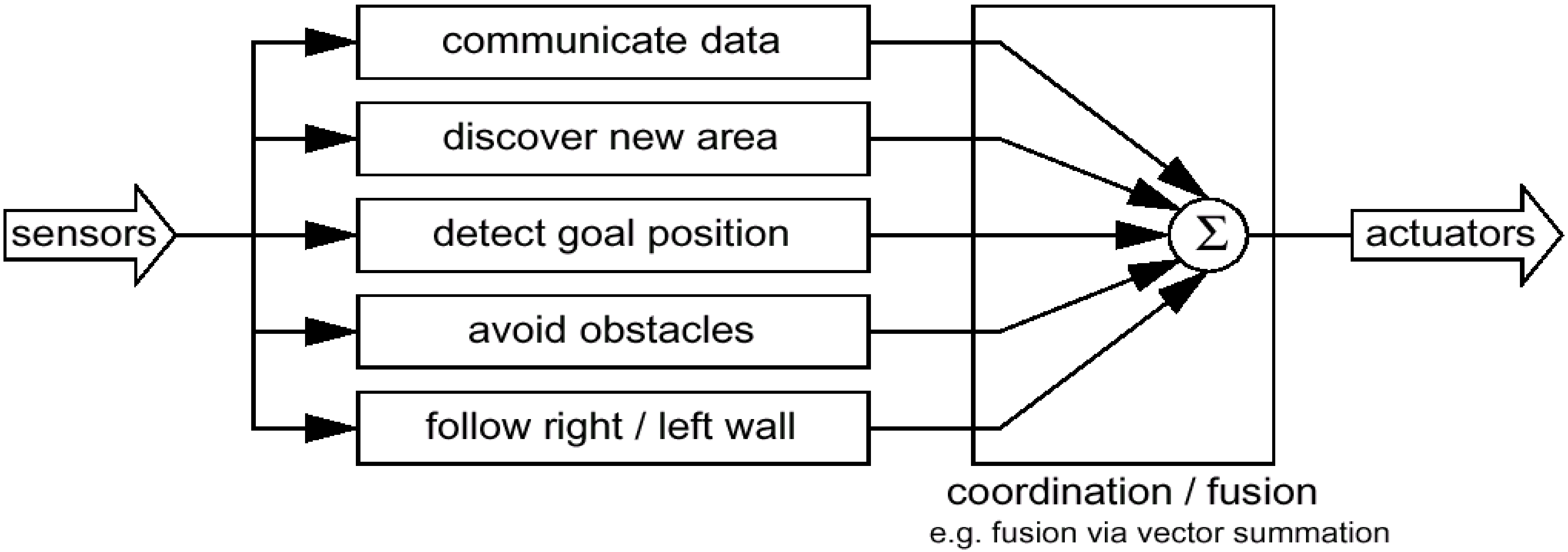


To localize or not?

- How to navigate between A and B
  - *navigation without hitting obstacles*
  - *detection of goal location*
- Possible by following always the left wall
  - *However, how to detect that the goal is reached*



Behavior Based Navigation



Model Based Navigation