Odometry

Odometry & Dead Reckoning

Odometry

- Use rotation sensors to update position estimate

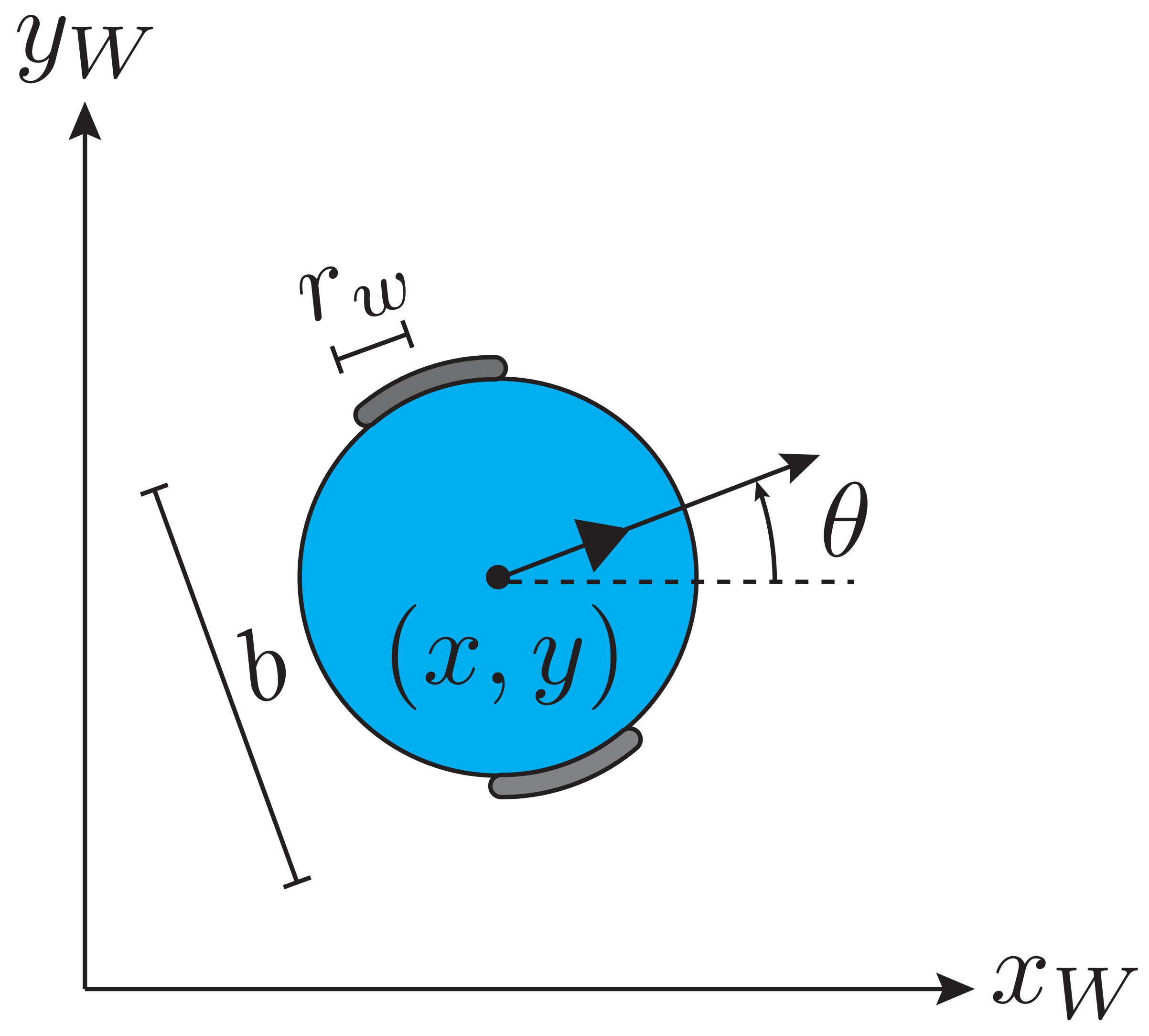
Deduced “Dead” Reckoning

- Use velocity sensors & heading sensor to update position estimate

- Errors accumulate and grow without bound
- Often used as an initial estimate of change in position to be refined with another sensor (i.e. a LIDAR, GPS etc.) - SLAM

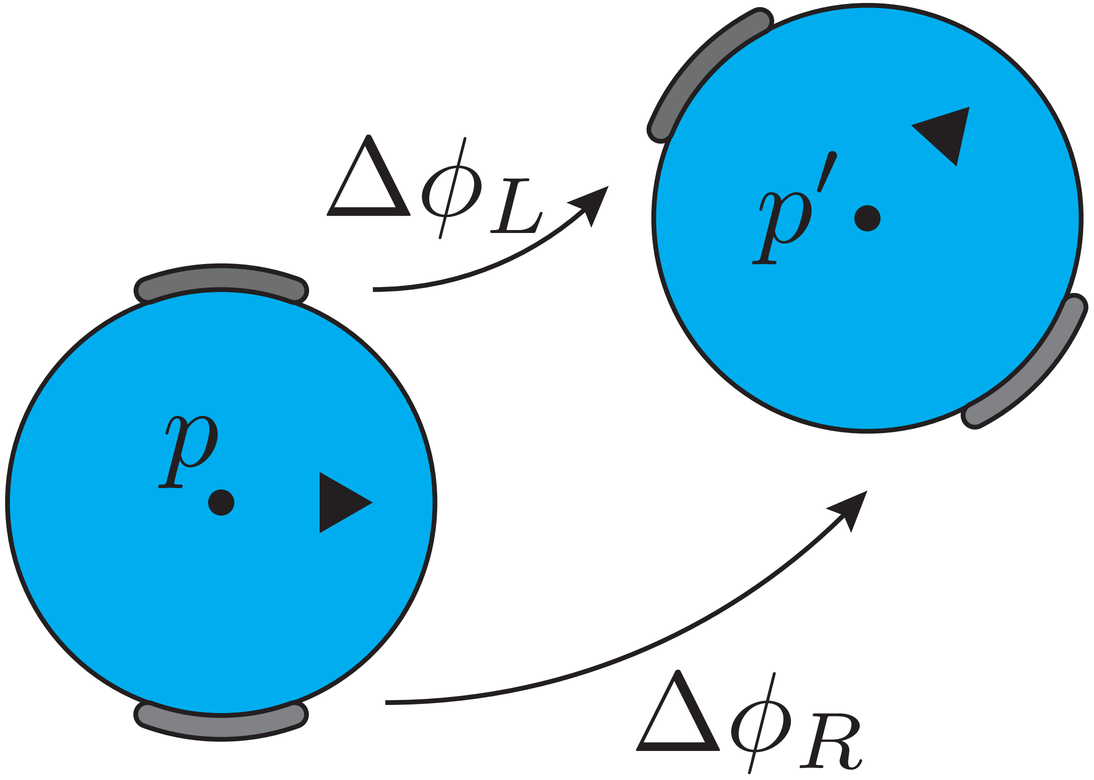
Robot Configuration

- Consider a differential drive robot in some global frame
- Configuration is given by
  $$(x, y, \theta) \in SE(2)$$
- Robot has wheel radius r_w
- Robot has wheelbase b



Odometry

- If the robot starts at point p in the global frame and each wheel rotates a certain angle, what is the new location p' ?
- To compute we first calculate the motion in the robots local reference frame and then map that motion onto the global frame



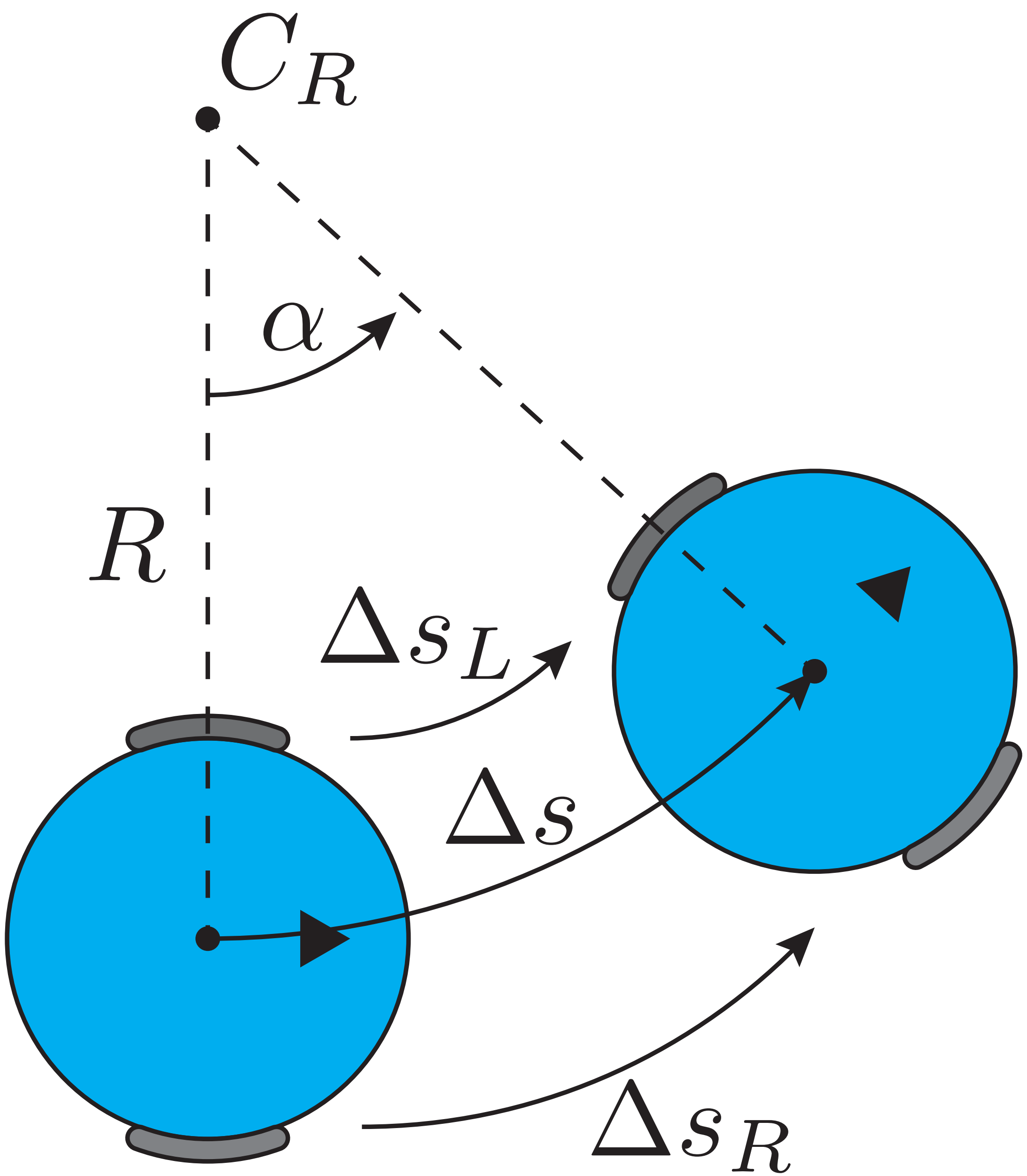
Motion Model

- Motion of robot is about a center of rotation.
- Angle swept by each wheel must be the same.
- For straight motion: $R = \infty$
- Turning in place: $R = 0$

$$\Delta s_L = (R - \frac{b}{2})\alpha$$

$$\Delta s_R = (R + \frac{b}{2})\alpha$$

$$\Delta s = R\alpha$$



Motion Model

The angle swept by the robots path:

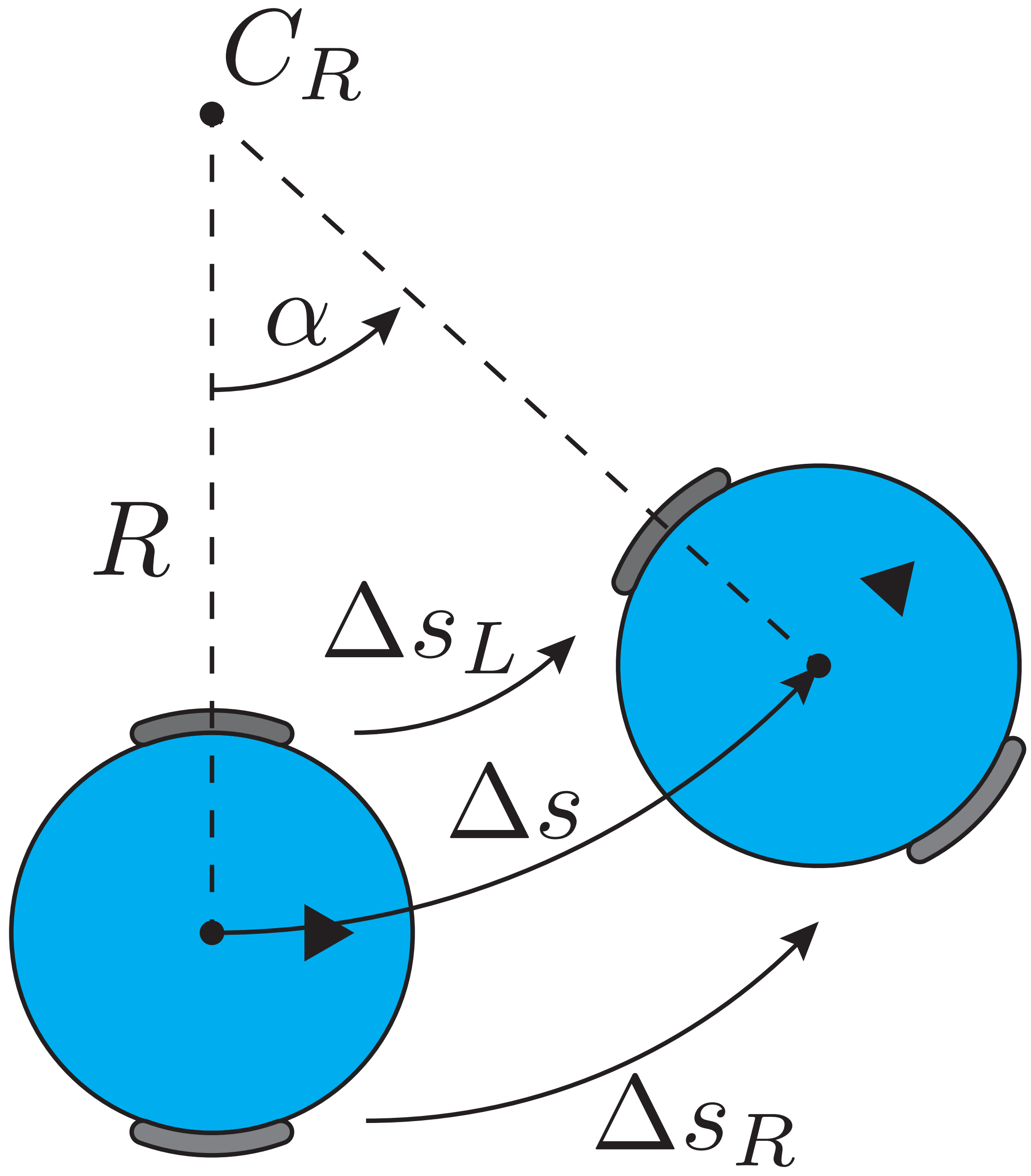
$$\Delta s_L + b\alpha/2 = R\alpha$$

$$\Delta s_R - b\alpha/2 = R\alpha$$

$$\Delta s_L + b\alpha/2 = \Delta s_R - b\alpha/2$$

$$b\alpha = \Delta s_R - \Delta s_L$$

$$\alpha = \frac{\Delta s_R - \Delta s_L}{b}$$



Motion Model

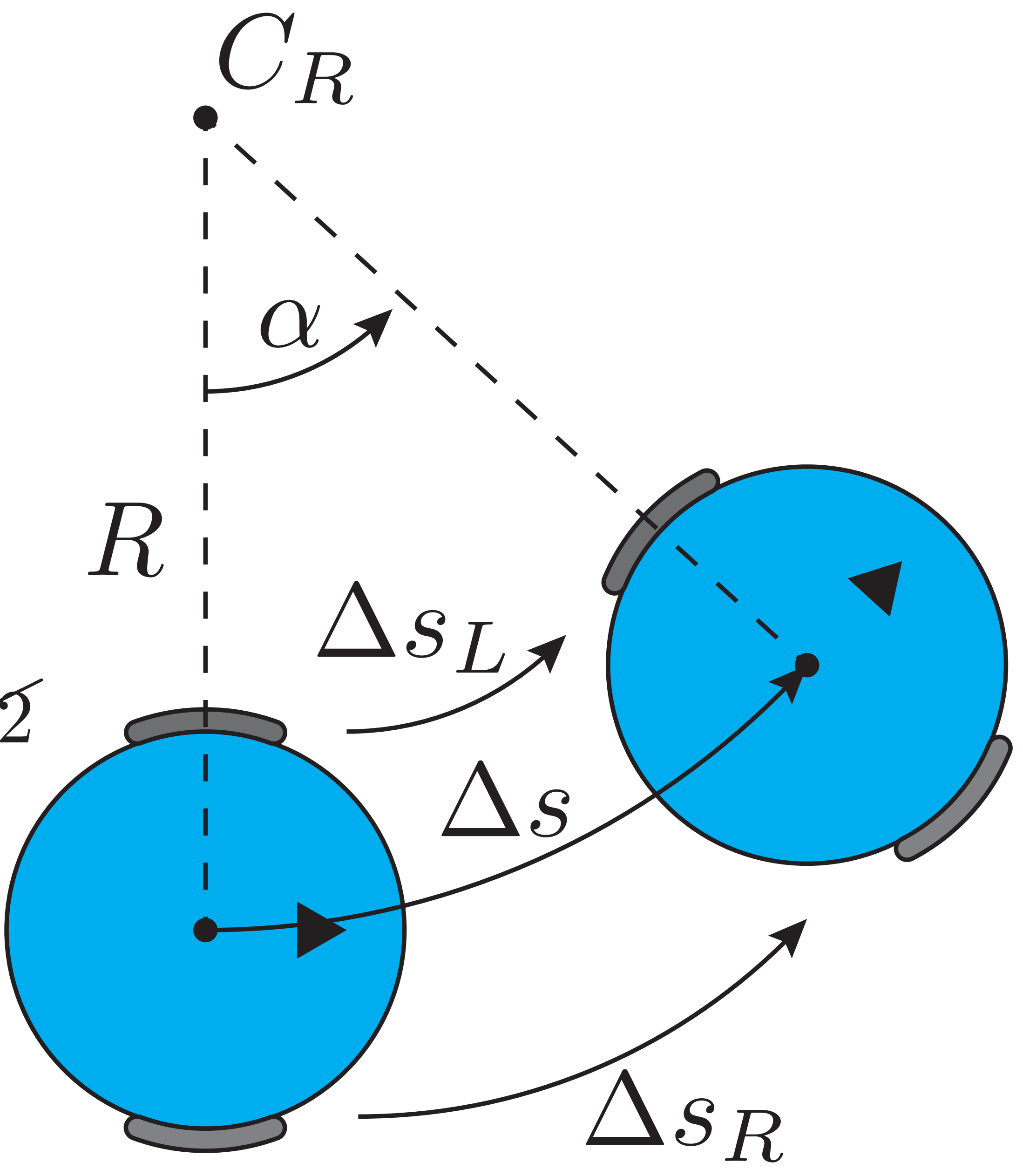
Distance travelled along path:

$$\Delta s_L = \Delta s - b\alpha/2$$

$$\Delta s_R = \Delta s + b\alpha/2$$

$$2\Delta s = \Delta s_L + \cancel{b\alpha/2} + \Delta s_R - \cancel{b\alpha/2}$$

$$\Delta s = \frac{\Delta s_L + \Delta s_R}{2}$$



Differential Drive Kinematics

- Note that:

$$\Delta\theta = \alpha$$

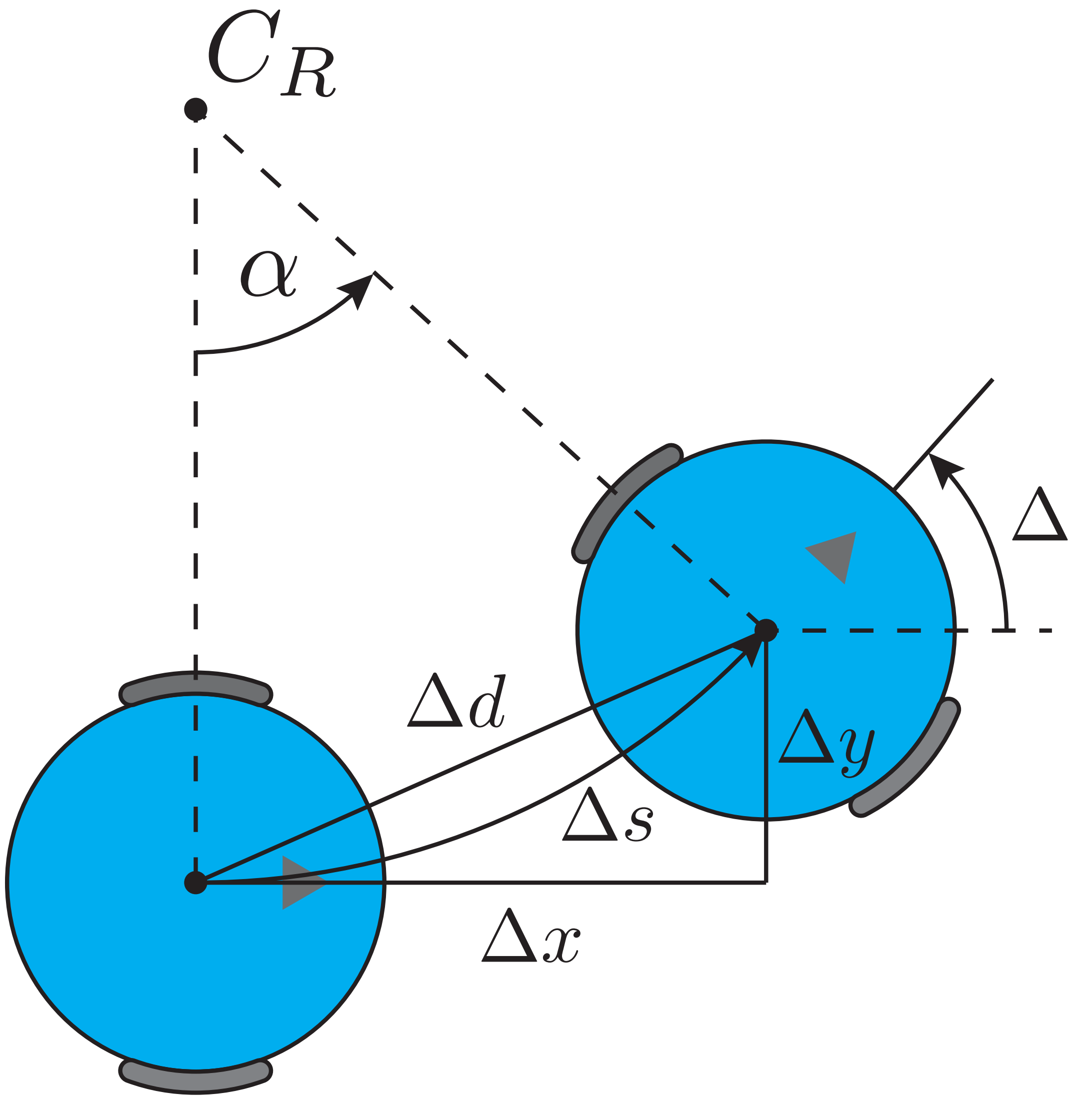
- For small changes:

$$\Delta d \approx \Delta s$$

- We can calculate how our position in the world changed:

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

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



Summary of Odometry Equations

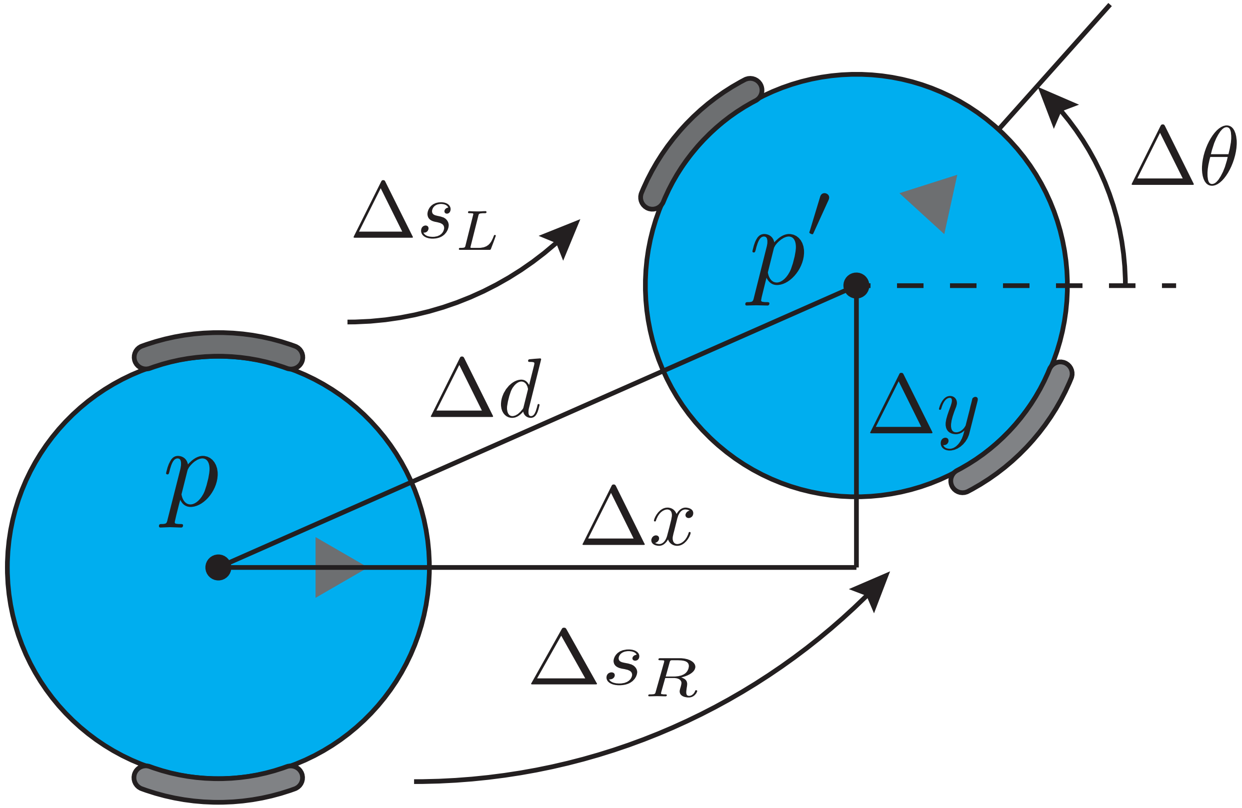
$$\Delta\theta = \frac{\Delta s_R - \Delta s_L}{b}$$

$$\Delta d = \frac{\Delta s_R + \Delta s_L}{2}$$

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

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

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

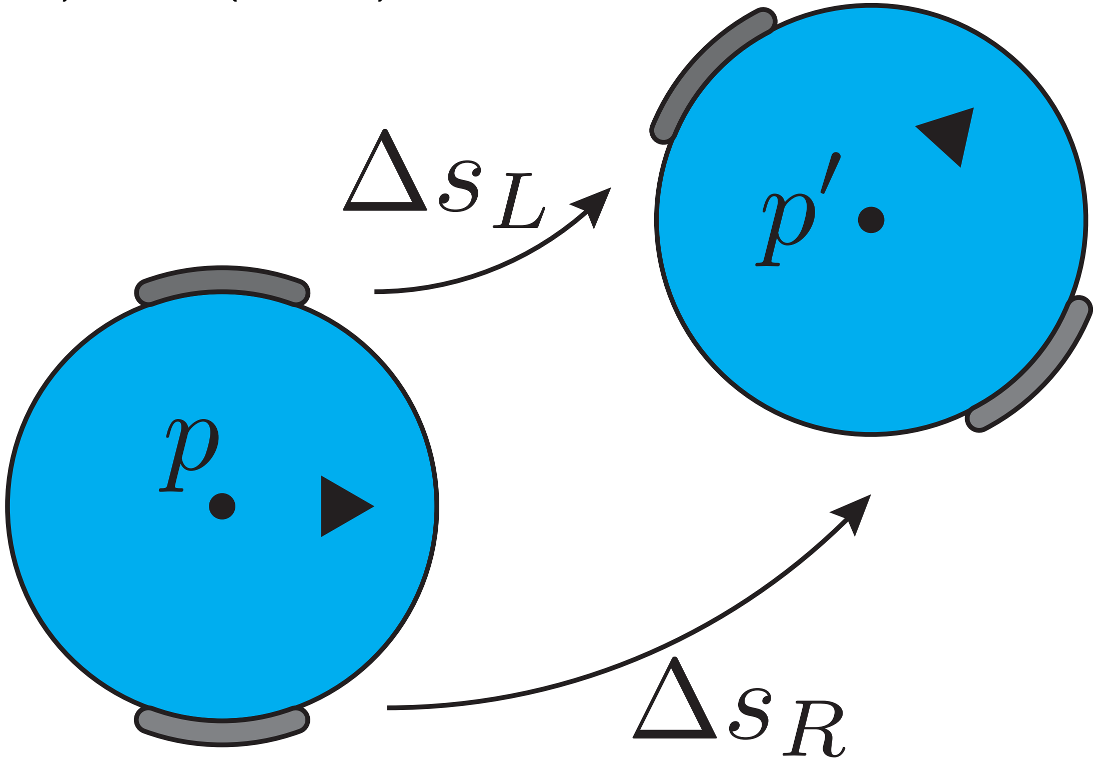


Odometry

$$\Delta\phi = 2\pi \times \Delta \begin{pmatrix}\text{encoder} \\ \text{reading}\end{pmatrix} \div \begin{pmatrix}\text{encoder} \\ \text{resolution}\end{pmatrix} \div \begin{pmatrix}\text{gear} \\ \text{ratio}\end{pmatrix}$$

- To perform odometry, we have a sensor on each motor shaft
- We want to know the distance each wheel traveled

$$\Delta s_w = r_w \Delta\phi_w$$



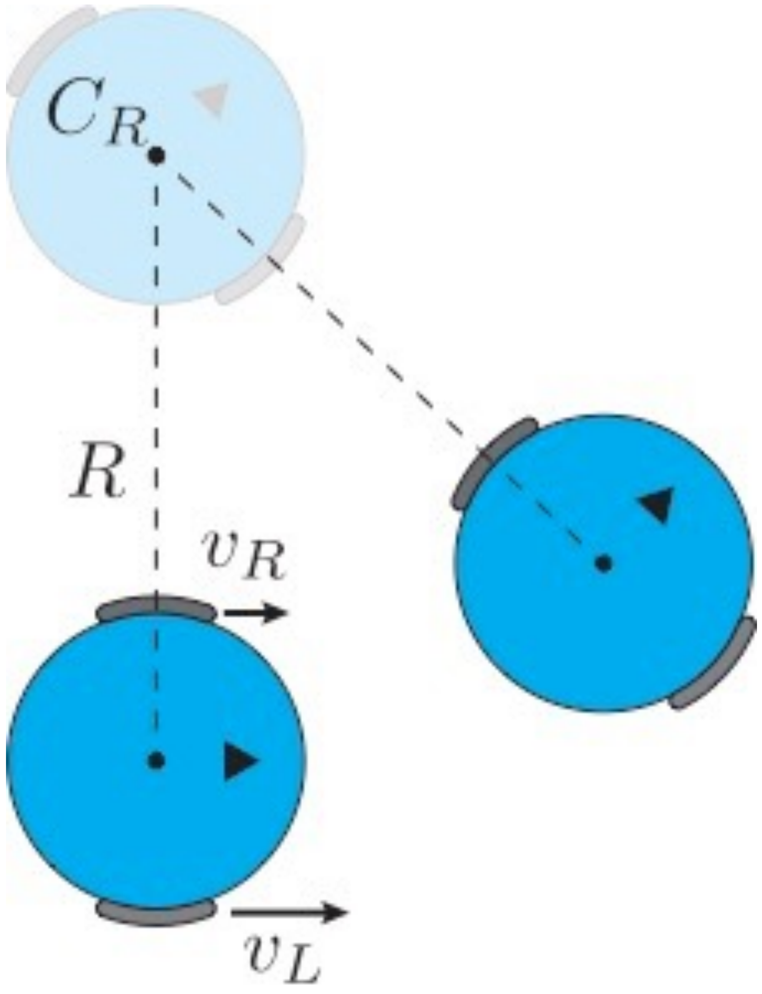
Wheel velocity model

$$v_R = \omega(R + b/2)$$
$$v_L = \omega(R - b/2)$$

$$R = \frac{b}{2}\frac{v_R + v_L}{v_R - v_L}$$

$$\omega = \frac{v_R - v_L}{b}$$

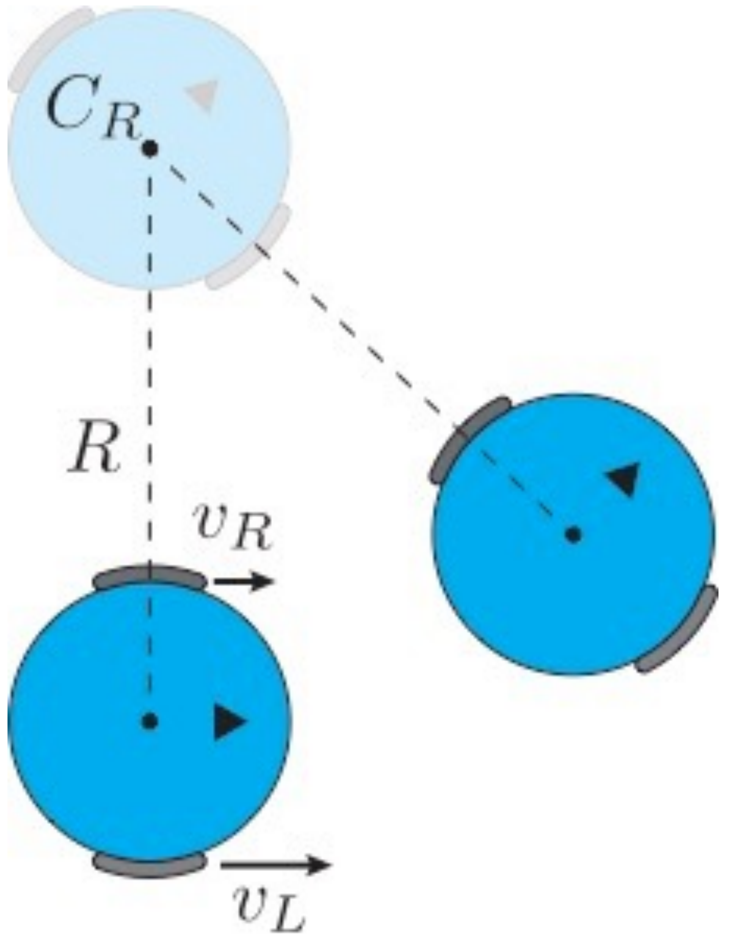
$$C_R = \begin{bmatrix} x - R\sin\theta & y + R\cos\theta \end{bmatrix}$$



Wheel velocity model

$$C_R = \begin{bmatrix} x - R\sin\theta & y + R\cos\theta \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - C_{R,x} \\ y - C_{R,y} \\ \theta \end{bmatrix} + \begin{bmatrix} C_{R,x} \\ C_{R,y} \\ \omega\Delta t \end{bmatrix}$$



- Translate along old Y axis, rotate by $\omega\Delta t$, translate back along new Y axis

Sources of Error

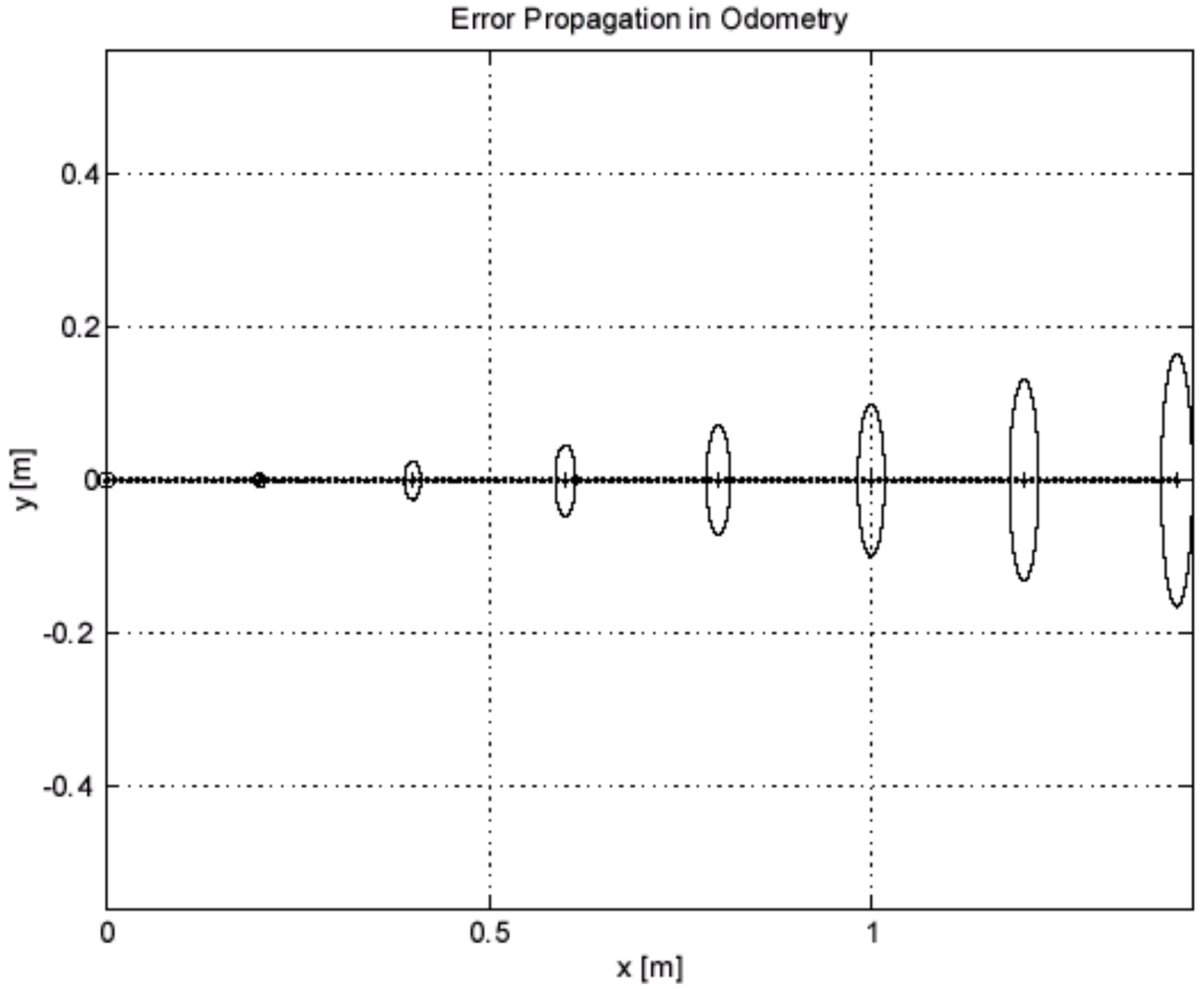
Systematic

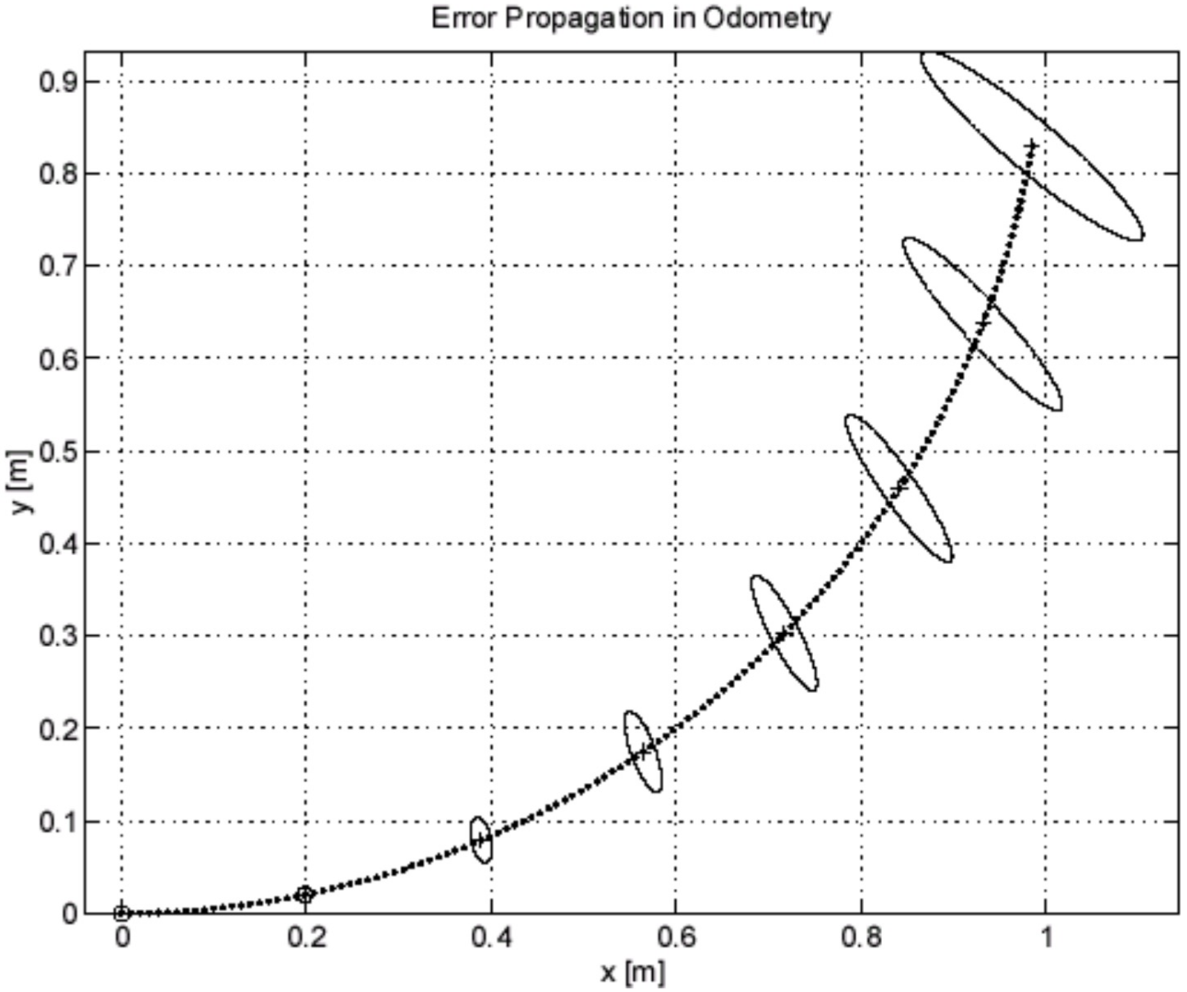
- Limited resolution of wheel sensors
- Unequal wheel diameter
- Wheel diameters not nominal
- Motor axes misaligned
- Uncertainty about the wheelbase

Non-systematic

- Travel over uneven floor
- Travel over an unexpected object on the floor
- Wheel slipping
  - Slippery floor
  - Over-acceleration
  - Fast turning (skidding & drifting)
  - Hitting Obstacles

Error Propagation





Dealing with systematic errors

- Can find average wheel diameter and actual baseline through a calibration called UMBmark
- Calibration finds the two major sources of systematic error

$$E_{\mathrm{d}} = D_{\mathrm{R}}/D_{\mathrm{L}} \qquad E_{\mathrm{b}} = b_{\mathrm{actual}}/b_{\mathrm{nominal}}$$

IEEE Transactions on Robotics and Automation, Vol 12, No 6, December 1996, pp. 869-880.

Measurement and Correction of Systematic Odometry Errors in Mobile Robots

by

Johann Borenstein and Liqiang Feng
The University of Michigan

UMBark

- Idea: Drive around square in both directions
- Incorrect baseline causes errors in turning, but not straight-line motion
- Unequal wheel diameter causes error in straight line motion but not turning