Localization

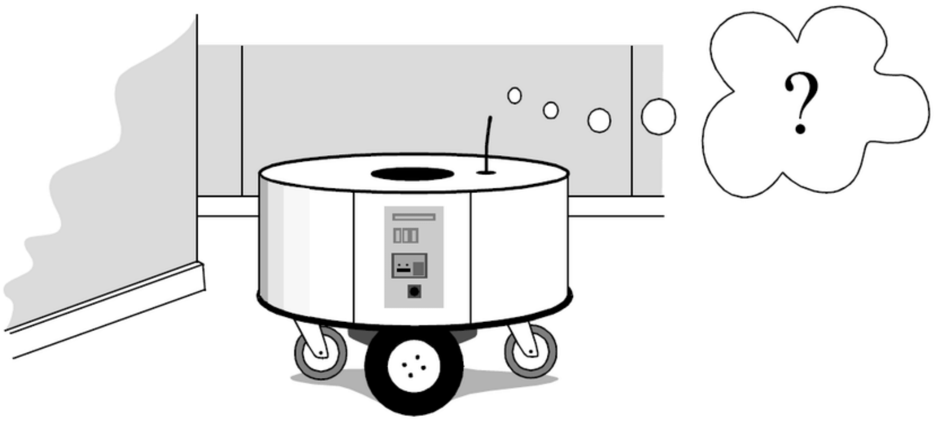


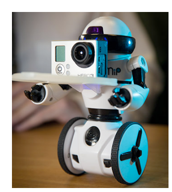
Image: Ibrahim, Omar. (2011). Extended Kalman Filter Simultaneous Localization and Mapping (Graduation Project)

1

A Note on Navigation

- Navigation is hard!
- Encompasses (at least) four components:

1. Perception: based on sensor data, what do I know about my environment?
2. Localization: Where am I in that environment?
3. Cognition: What should I do now?
4. Motion Control: How do I do that?



Navigation is a hard problem, but many tasks depend on it

www.youtube.com/watch?v=Is4JZqhAy-M
ucsdnews.ucsd.edu/feature/real_life_toy_story

2

2

Localization ≠ Mapping

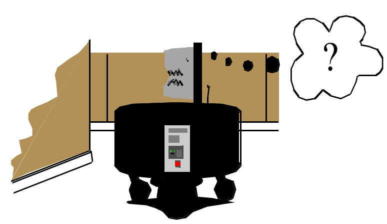
- **Mapping:** creating a map of an environment
- **Localization:** sensor and odometry data are used to figure out where robot is *in* a map
- Needs some kind of info about environment
  - Typically a map of some kind
    - Physical, semantic, topological
- Doesn't care about source of information
    - Could be given a map beforehand
    - Or constructing the map as you go
- Localization often includes mapping as a step

4

4

Localization + Map Building

- To localize or not to localize
  - When is hard-coding better?
- Belief representation
  - How do I represent the environment and my state?
- Map representation
  - What does a map contain?
- And more…
  - Probabilistic map-based localization
  - Autonomous map building



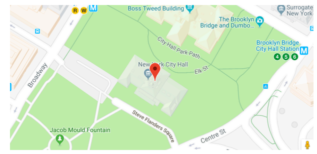
5

Image: Ibrahim, Omar. (2011). Extended Kalman Filter Simultaneous Localization and Mapping (Graduation Project)

5

Challenges of Localization

- Knowing absolute position (e.g. GPS) isn't enough

  Lat 40.7127, Long -74.0059 ≠ 

- Localization in human-scale
  - "Give or take 5 meters" – in a building? On the street?
- Many sources of uncertainty
  - Sensor noise, sensor aliasing, effector noise, odometric position estimation, …
- Aliasing (coming soon)

7

7

Sensor Noise

- Sensors give "noisy" (uncertain, imperfect) readings
- Source of sensor noise may be environmental
  - Surfaces, illumination, background noise…
  - Glass walls 😕

- Or the nature of the sensor
  - Interference between ultrasonic sensors
  - Cameras in high dynamic range lighting (like outside)
- Or may just be because sensors are imperfect

8

vivarailings.com/portfolio_page/aluminum-glass-railing-suny/

8

Challenge 1: Sensor Noise

- Sensor noise drastically reduces useful readings
- Solutions:
  - Improve sensors
  - Change assumptions
    - "So, if we have glass walls, we'll see readings like…"
  - Use multiple readings
  - Employ temporal and/or multi-sensor fusion

"Is my environment made of dots?"

9

www.photoreview.com.au/tips/shooting/how-to-control-image-noise

9

Challenge 2: Sensor Aliasing

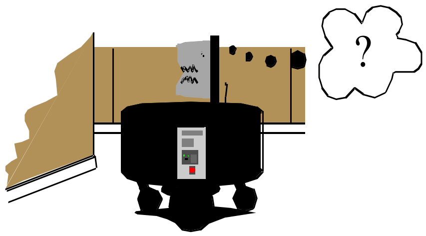
- **Different** positions give the **same** sensor readings
- Robots: non-uniqueness of sensors readings is normal
  - What does that mean?
- To people, unique places look unique
  - We're really good at picking up on differences
  - We have really good sensors
- To robots, different places often look the same
  - There is a many-to-one mapping from environmental state to robot's perceptual inputs

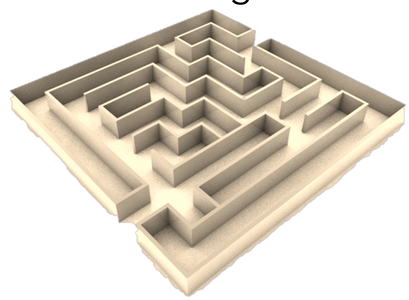
10

10

Sensor Aliasing (2)

- Different places give the same readings



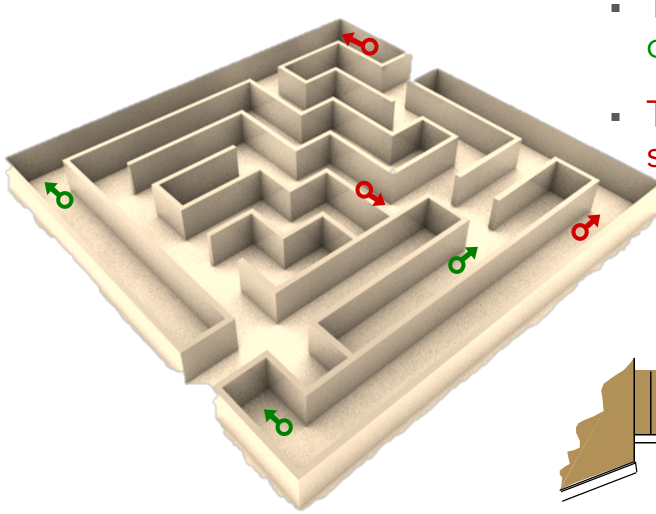


- Information from sensors often not enough to identify position from a single reading
  - Solution: localization usually based on a series of readings
  - Enough information is recovered by the robot over time

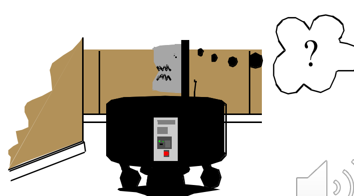
11

11

Sensor Aliasing (3)



- These look different.
- These look the same.
  - Wall in front, wall on the right, opening on left.



12

12

Odometry, Dead Reckoning

- Odometry: wheel sensors only
  - E.g., go 5 cm, turn 10 degrees – where are you?
- Dead reckoning: also use heading sensors
  - Add a compass or gyroscope
- Position update is partly based on proprioceptive sensors
  - Sensing movement: wheel encoders + heading sensors
  - Integrate that into model of environment to get the position
    - Pros: Straightforward, easy
    - Cons: Errors are integrated → accumulative

13

13

Effector (Actuator) Noise

- Causes:
  - Inexact actuation + noise in sensors: Probably not exactly 5 cm
  - Environment: Duct tape is slippery!
- This error is cumulative over time, but reduced with additional sensors (not eliminated)
- Errors exist on a spectrum:

Deterministic (systematic): "This servo always turns 2% too far" ⟷ Non-deterministic (random): "Sometimes this servo goes too far or not far enough"

14

14

Odometry: dealing with errors

- Deterministic errors can be eliminated with calibration
- Random errors can be described by error models
  - Will always lead to uncertain position estimate
- More major sources of error:
  - 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, not planar …)

15

15

Some movement errors

- **Range error:** Integrated path length (distance) of movement is wrong
  - How far has robot moved in heading direction?
- **Turn error:** similar to range error, but for turns
  - What's robot's θ from starting position?
  - Accumulated error over multiple turns
- **Drift error:** difference in wheels → error in angular orientation
  - Right wheel turns 90˚, left turns 89.8˚. What happens?

16

16

Error Severity

- Over long periods of time, turn and drift errors **far** outweigh range errors.
  - As θ grows linearly, change in location grows nonlinearly
- Why?
- Imagine moving forward a distance d on a straight line along axis x.
- As $\Delta\theta$ (angular error) grows, error in y will have a component of $d \sin \Delta\theta$.

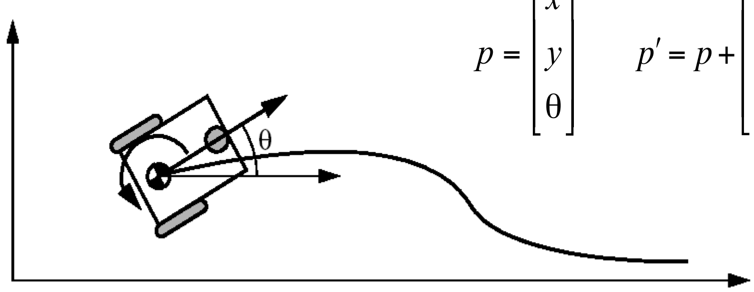
17

17

Odometry & Diff. Drive *

- Discrete sampling rate Δt
- Δs_r, Δs_l: right wheel, left wheel distance travelled
- Changes in pose: Δx, Δy, $\Delta\theta$
- Distance between wheels (wheel base): b

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



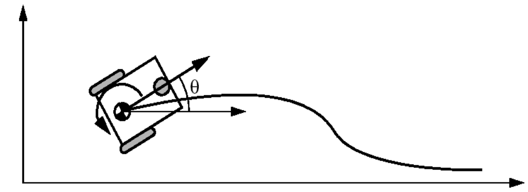
18

* 2 wheels, shared axis, each can be rotated forward or back

18

Odometry & Diff. Drive

- Derivations
  - Δt = Sampling rate
  - Δs_r, Δs_l = right/left wheel travel
  - b = Wheel base (distance between wheels)



These are *given* here. These can be derived, but try to make mechanical/ intuitive sense of them.

$$p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} \quad \left\{ \begin{aligned} \Delta s &= (\Delta s_r + \Delta s_l)\ /2 \\ \Delta x &= \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta y &= \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta &= (\Delta s_r - \Delta s_l)\ /b \end{aligned} \right.$$

19

19

Odometry & Diff. Drive

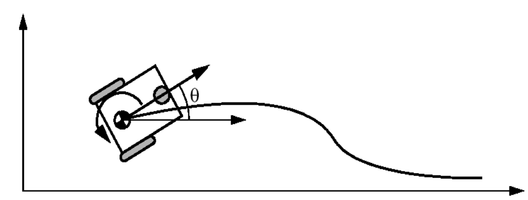
- Kinematics

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

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

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

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



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

This is the matrix for finding p'. Don't worry about it too

20

20

Odometry & Diff. Drive

- **How to model error:** Represent uncertainty of location over time
  - Using a covariance matrix of position estimate – how do these parameters vary with respect to each other?
- We can assumpe:
  - Left and right wheel errors are independent
  - $d \propto \Delta s_r, \Delta s_l$: Variance of wheel errors are proportional to distance traveled
  - An initial matrix Σ_p is known
- So we can get a covariance matrix that describes how error varies as a function of terms.
  - The derivation of this matrix is in the text – for now, make sure you have the general idea.

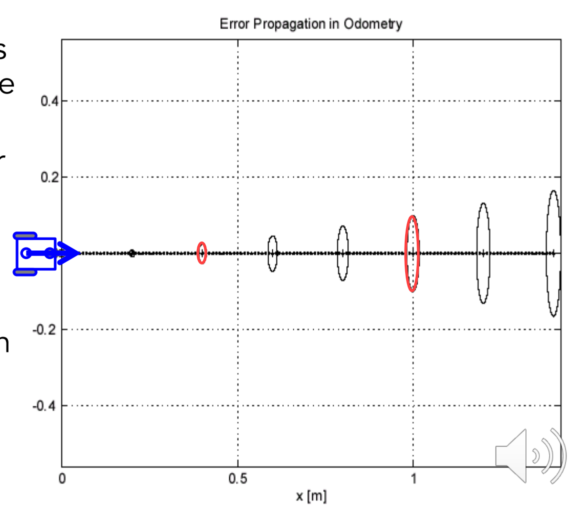
21

21

Odometry:

Growth of Pose Uncertainty for Straight Line Movement

- Intuitively: how do errors grow as the robot moves around?
  - Think of the circles (error) as where the robot *might* be
  - Errors grow slower in x (direction of travel) than in y
  - So, robot is more likely to be off to the side than ahead or behind of where you expect

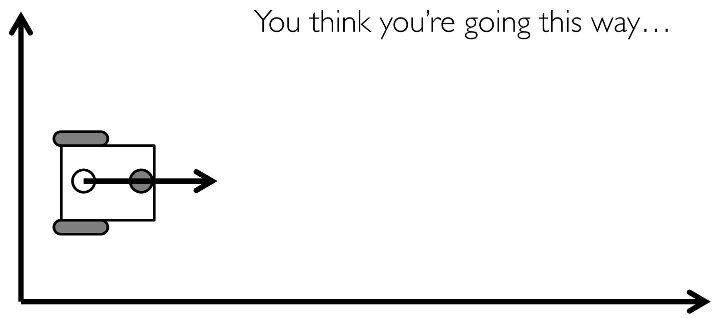


23

23

Error Severity

- Errors grow slower in x (direction of travel) than y

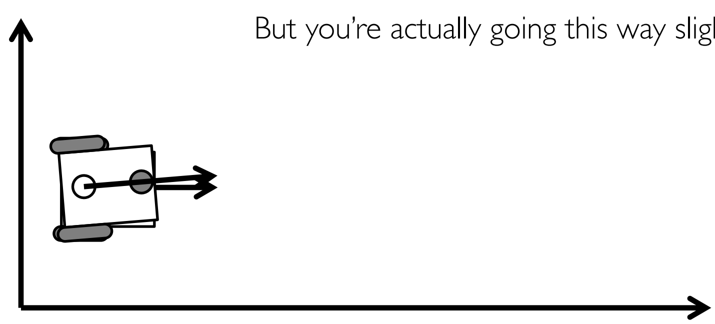


24

24

Error Severity

- Errors grow slower in x (direction of travel) than y



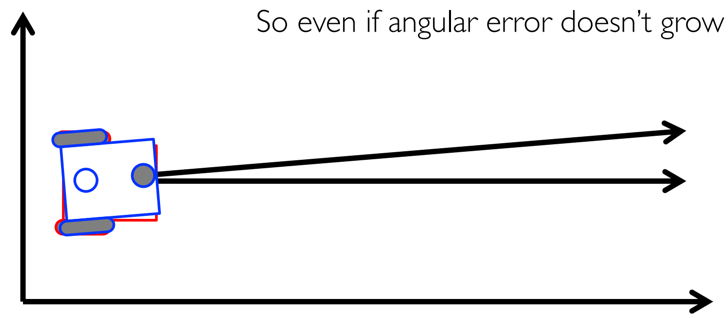
25

25

Error Severity

- Errors grow slower in x (direction of travel) than y

So even if angular error doesn't grow…



Expected vs. actual

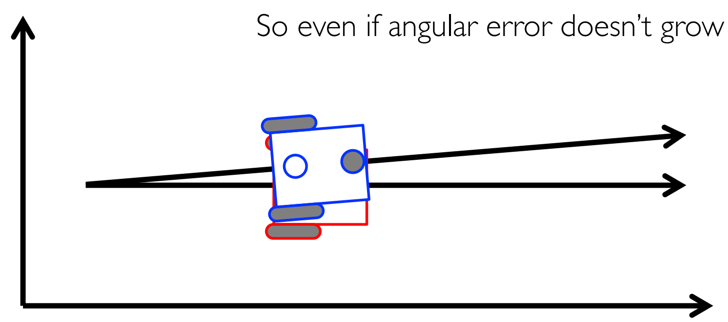
26

26

Error Severity

- Errors grow slower in x (direction of travel) than y

So even if angular error doesn't grow…



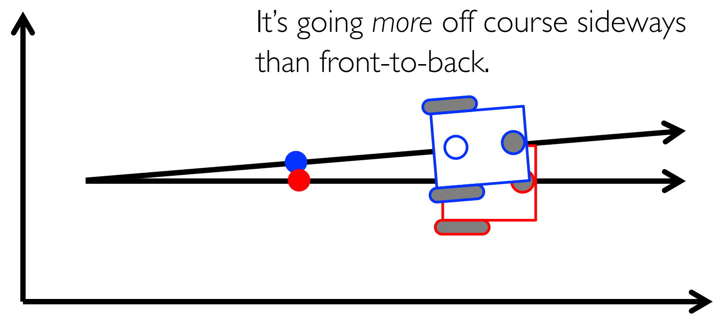
Expected vs. actual

27

27

Error Severity

- Errors grow slower in x (direction of travel) than y



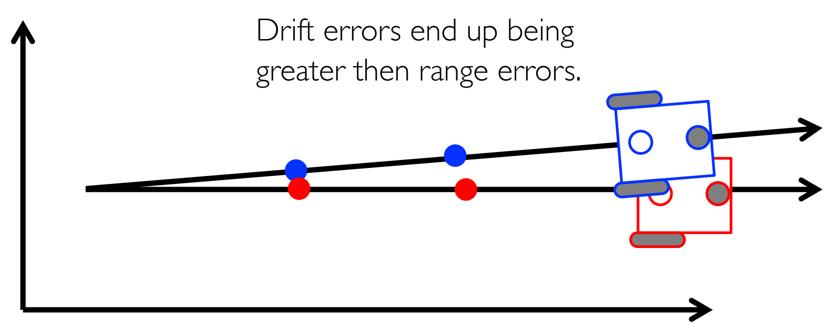
Expected vs. actual

28

28

Error Severity

- Errors grow slower in x (direction of travel) than y



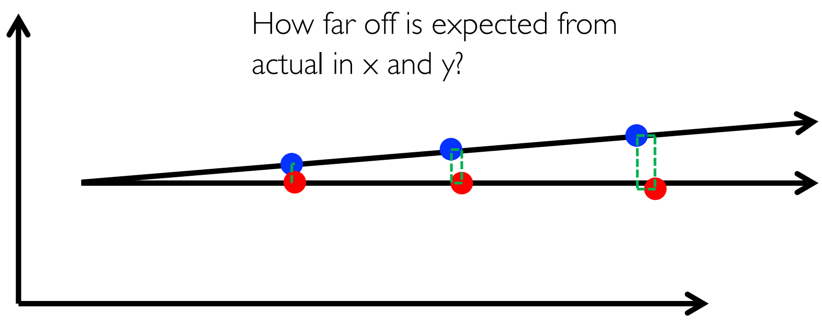
Expected vs. actual

29

29

Error Severity

- Errors grow slower in x (direction of travel) than y



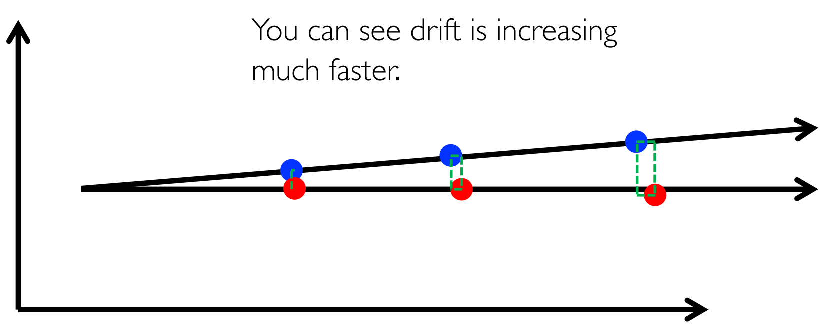
Expected vs. actual

30

30

Error Severity

- Errors grow slower in x (direction of travel) than y



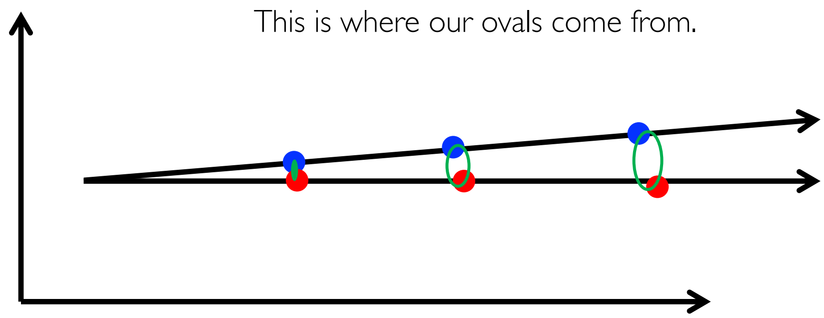
Expected vs. actual

31

31

Error Severity

- As $\Delta\theta$ (angular error) grows, error in y will have a component of $d \sin \Delta\theta$.



- You can do the same thing with a curved line.

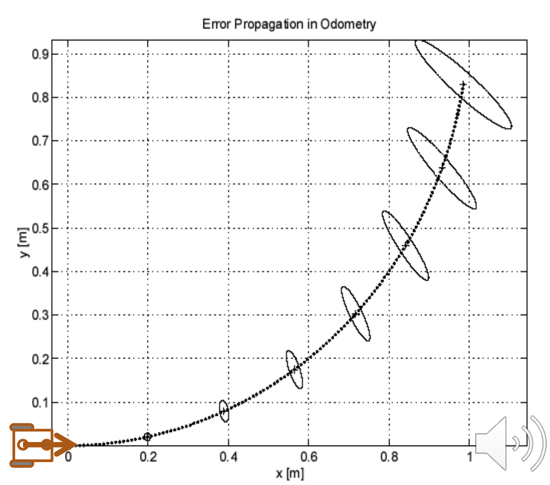
32

32

Odometry:

Growth of Pose uncertainty for Movement on a Circle

- Now imagine moving in a curve
- Errors don't stay perpendicular to direction of movement

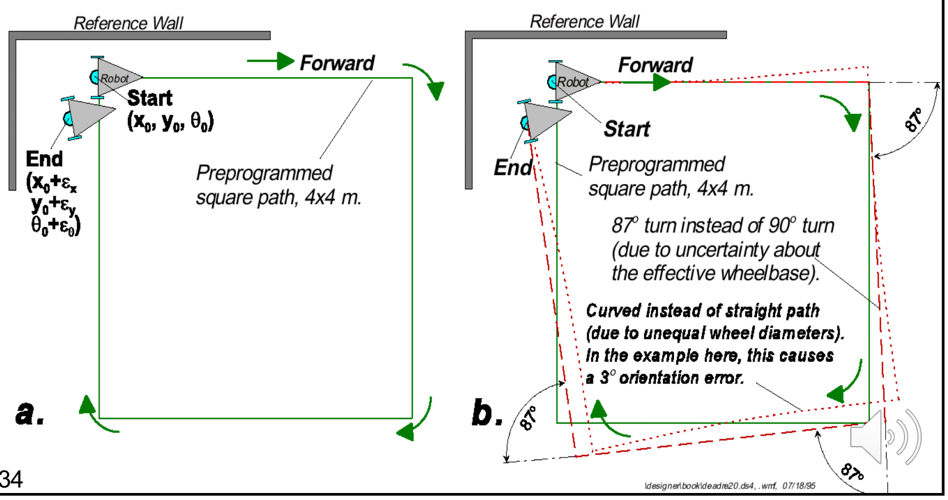


33

33

Calibration of Errors

- The unidirectional square path experiment

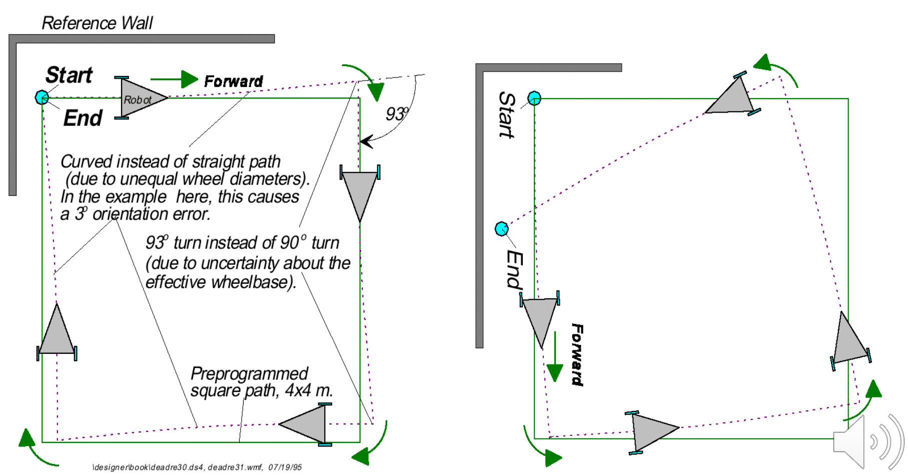


34

34

Calibration of Errors

- Bi-directional square path experiment

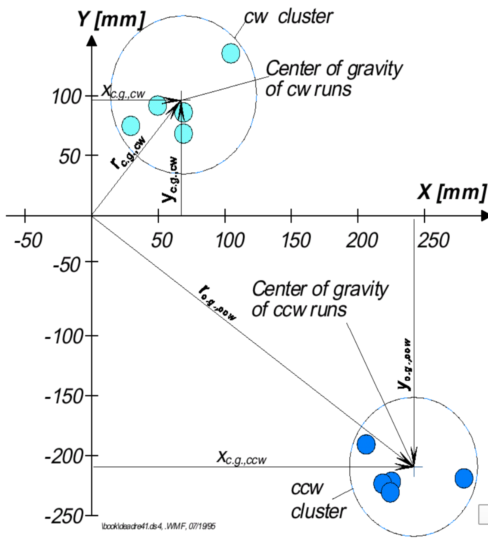


35

35

Calibration of Errors

- Look at actual path traversed: what errors occur?
  - That is, where in the x, y space do we not see the expected location?
- Deterministic and non-deterministic errors

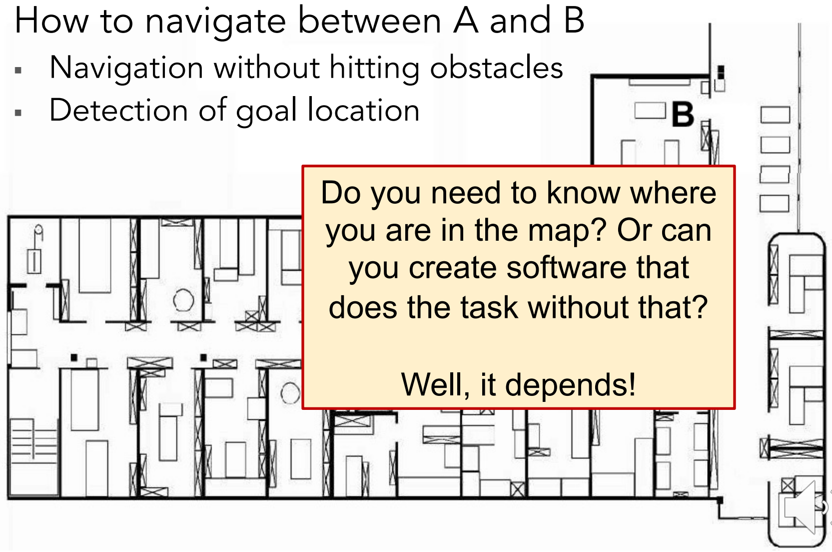


36

36

To Localize, Or Not To…?

- How to navigate between A and B
  - Navigation without hitting obstacles
  - Detection of goal location

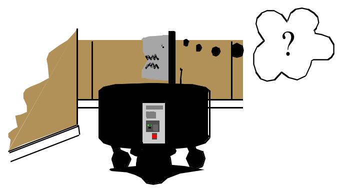


37

37

Localization Summary (1)

- What is localization?
  - Figuring out location wrt. a model of the world
- Purely proprioceptive approaches:
  - Odometry: belief about motion only
    - Wheel encoders, mostly
  - Dead reckoning: belief about motion + heading sensors

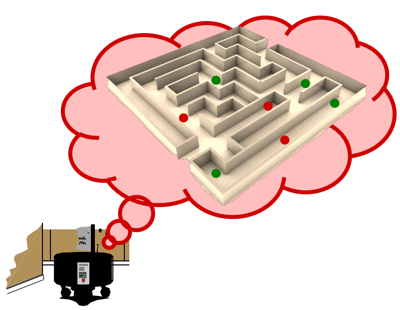


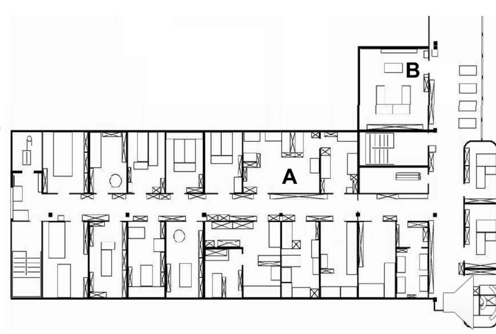
38

38

Localization Summary (2)

- What is sensor aliasing?
  - Different locations giving the same sensor readings
- What is behavior-based navigation?
  - Navigating without localizing





39

39

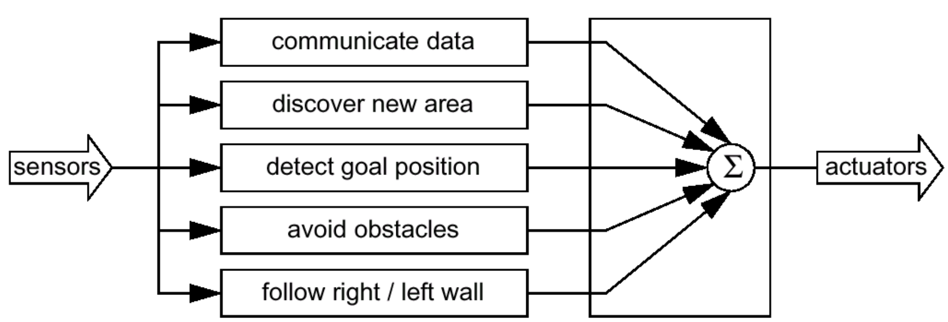
Behavior based navigation

- An alternative to localization
- When you see <*input*>, do <*action*>.
  - Given *these* inputs, behave *this* way.
- When is this a good choice?
  - ✓ Fast to implement
  - ✓ Robust against error accumulation
  - ✓ Effective in unchanging environment
  - x Does not scale to new environments
  - x Behaviors must be designed and debugged
  - x Sensor changes change behavior

40

40

Behavior Based Navigation

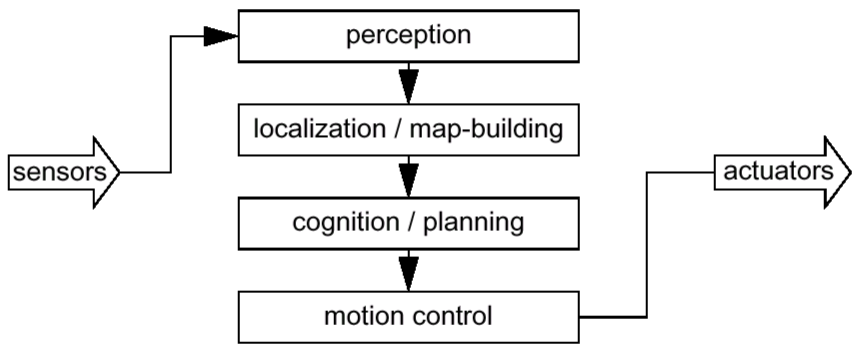


coordination / fusion
e.g. fusion via vector summation

41

41

Model Based Navigation



42

42