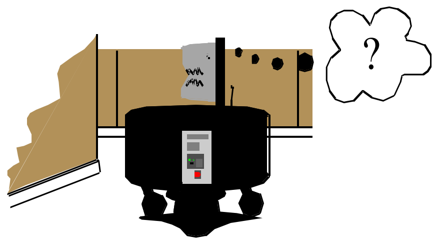
Localization
where am I?



A Note on Navigation

2

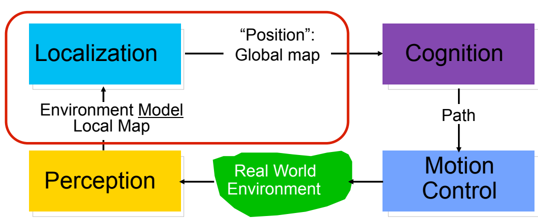
- Navigation is hard!
- Encompasses (at least) four components:
1. **Perception:** based on sensor data, what do I know about my environment?
2. **Localization:** Where am I in that environment?
3. **Cognition:** What should I do now?
4. **Motion Control:** How do I do that?

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

Navigation Cycle

3



Localization ≠ Mapping

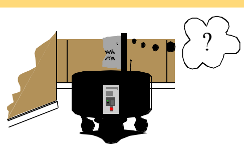
5

- Sensor and odometry data used to identify where in an environment you are
- Need some kind of info about environment
- Typically a map of some kind
  - Physical, semantic, topological
- Localization doesn't care about source of information
  - Could be given a map beforehand
  - Or constructing the map as you go
- Localization often subsumes mapping

Localization + Map Building

6

- Challenges: noise and aliasing
  - Odometric position estimation
- 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 kind of info does a map contain?
- And more…
  - Probabilistic map-based localization
  - Autonomous map building



Challenges of Localization

8

- Knowing absolute position (e.g. GPS) is not sufficient
  - Lat 40.7127, Long -74.0059 ≠

- Localization in human-scale
  - "Give or take 5 meters" – in a building? On the street?
- May need >1 position to plan task
- Many sources of uncertainty
  - Sensor noise, sensor aliasing, effector noise, odometric position estimation

Sensor Noise

9

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

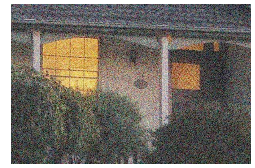
vivarailings.com/portfolio_page/aluminum-glass-railing-suny/

- Or by 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

Challenge 1: Sensor Noise

10

- Sensor noise drastically reduces useful sensor readings



"Where am I?"

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

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

Challenge 2: Sensor Aliasing

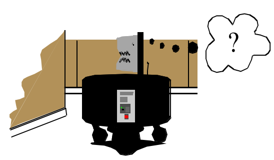
11

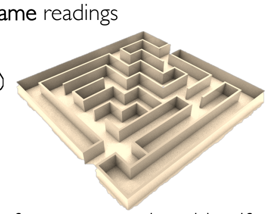
- Different positions give the same sensor readings
- Robots: non-uniqueness of sensors readings is normal
  - What does that mean?
- To people, unique places look unique
  - We're really good at picking up on differences
  - We have really good sensors
- To robots, distinct places often look the same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

Sensor Aliasing (2)

12

- **Different** places give the **same** readings

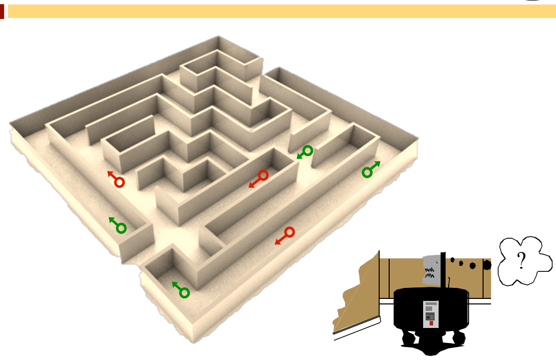




- Information from sensors often not enough to identify position from a single reading
  - Robot's localization is usually based on a series of readings
  - Sufficient information is recovered by the robot over time

Sensor Aliasing (3)

13



Odometry, Dead Reckoning

14

1. Odometry: wheel sensors only
   - E.g., you tell your robot to go 5 cm and turn 10 degrees
2. Dead reckoning: also heading sensors
   - If your robot had mini-GPS

- Position update is based on **proprioceptive** sensors
  - Sense movement with wheel encoders + heading sensors
  - Integrate that into model of environment to get the position
    - Pros: Straightforward, easy
    - Cons: Errors are integrated → unbound

Effector (Actuator) Noise

15

- 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 0.2% too far"	⟷	Non-deterministic (random): "Sometimes this servo goes too far or not far enough"

Odometry: Error sources

16

- Deterministic errors can be eliminated with calibration
- Random errors can be described by error models
  - 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, not planar …)

Classification of Errors

17

- **Range error:** Integrated path length (distance) of movement
  - How far have I gone?
  - Sum of wheel movements: 5cm + 5cm + 5cm = …16cm?
- **Turn error:** similar to range error, but for turns
  - What's my θ from starting position?
  - Accumulated error over multiple turns
- **Drift error:** difference in the error of the wheels → error in angular orientation
  - One wheel turns 90°, other turns 89.8°. What happens?

Error Severity

18

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

Odometry & Dead Reckoning

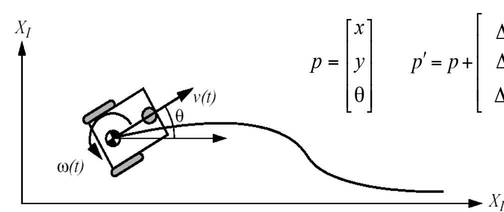
19

- Position update is based on proprioceptive sensors
  - Odometry: uses …
  - Dead reckoning: uses …

Odometry & Diff. Drive

20

- 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

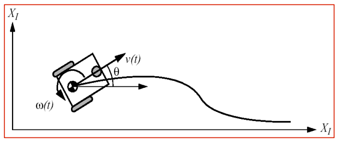


$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}$$

Odometry & Diff. Drive

21

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

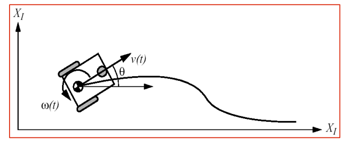


$$p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} \quad \begin{cases} \Delta s = (\Delta s_r + \Delta s_l)/2 \\ \Delta x = \Delta s\cos(\theta + \theta/2) \\ \Delta y = \Delta s\sin(\theta + \theta/2) \\ \Delta\theta = (\Delta s_r - \Delta s_l)/b \end{cases}$$

Odometry & Diff. Drive

22

- 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 & Diff. Drive

23

- Modeling Error: Represent uncertainty of location over time in a covariance matrix of position estimate
- Assumptions:
  - Left/right wheel errors are independent
  - $d \propto \Delta s_r, \Delta s_l$: Variance of wheel errors proportional to distance traveled
  - Initial matrix is Σ_p known
- So we can get a covariance matrix that describes how error varies as a function of terms

Odometry & Diff. Drive

24

Derivation in text. Make sure you have the general idea.

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

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

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

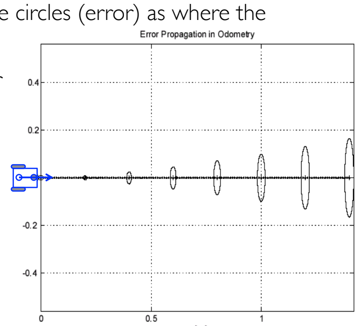
$$F_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\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

Odometry:

Growth of Pose Uncertainty for Straight Line Movement

25

- You can think of the circles (error) as where the robot *might* be
- Errors grow slower in x (direction of travel) than in y
- More likely to be off to the side than ahead or behind of where you intended

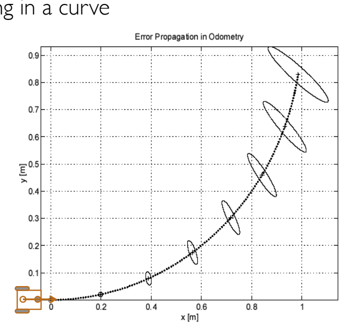


Odometry:

Growth of Pose uncertainty for Movement on a Circle

26

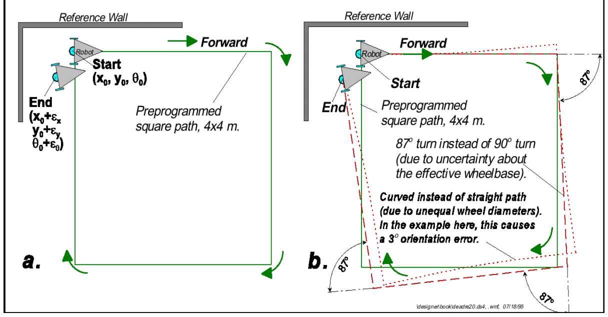
- Now imagine moving in a curve
- Errors don't stay perpendicular to direction of movement



Calibration of Errors

27

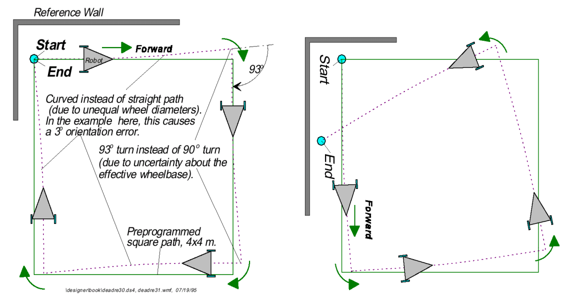
- The unidirectional square path experiment



Calibration of Errors

28

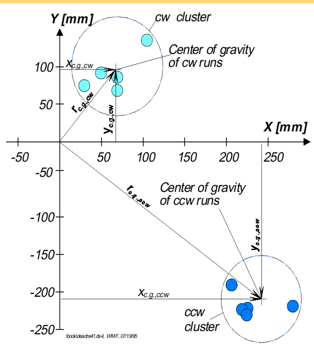
- Bi-directional square path experiment



Calibration of Errors

29

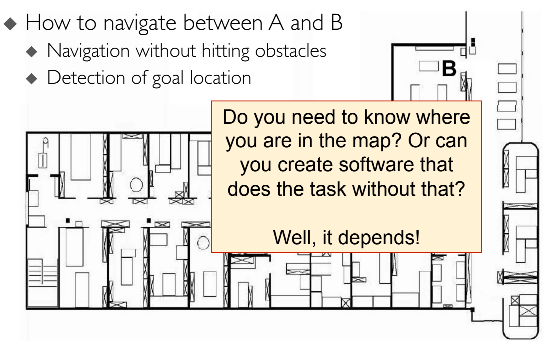
- Look at actual path traversed: what errors occur?
  - That is, where in the x, y space do we see Δ from expected location?
- Deterministic and non-deterministic errors



To Localize, Or Not To…?

30

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



Do you need to know where you are in the map? Or can you create software that does the task without that?

Well, it depends!

Localization Summary (1)

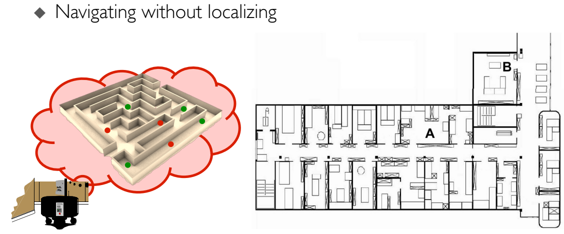
31

- 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

Localization Summary (2)

32

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



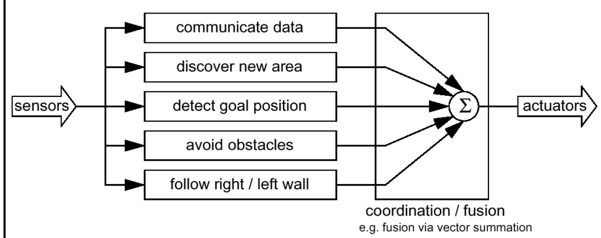
Behavior Based Navigation

33

- When you see X, do Y.
  - 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

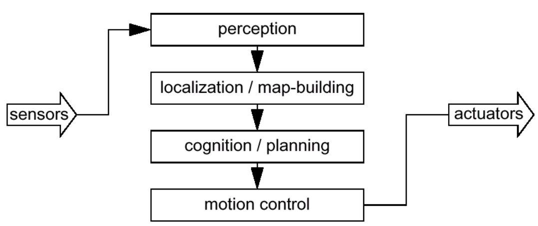
Behavior Based Navigation

34

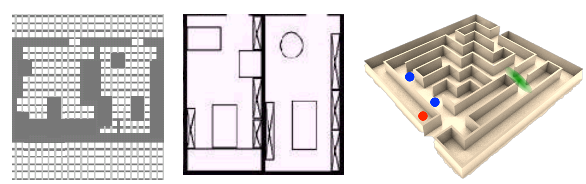


Model Based Navigation

35



Belief Representation



Location (Belief) Representation

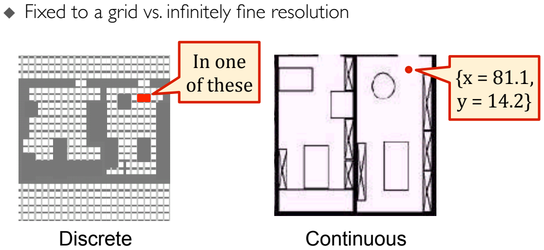
37

- How do we characterize belief?
- Discrete vs. continuous
  - Fixed to a grid, or anywhere?
- Single vs. multiple hypotheses
  - At any given time, how many possible locations are being considered?
- Probabilistic vs. bounded vs. point
  - The first two are multiple-hypothesis

Characterizing Belief R. (1)

38

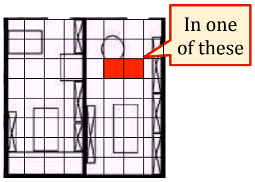
- Discrete vs. continuous
  - Fixed to a grid vs. infinitely fine resolution



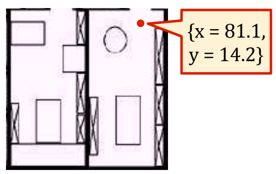
Characterizing Belief R. (1.1)

39

- Discrete vs. continuous
  - Belief can be discretized on a continuous map



Discrete

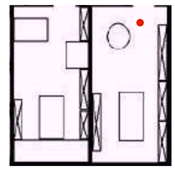


Continuous

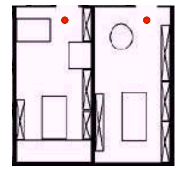
Characterizing Belief R. (2)

40

- Single hypothesis vs. multiple hypothesis



Single

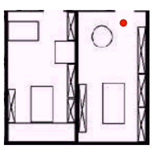


Multiple

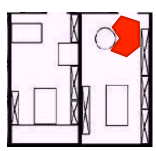
Characterizing Belief R. (3)

41

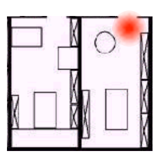
- Probabilistic vs. bounded vs. point



Point



Bounded Polygon



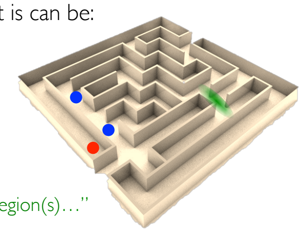
Probabilistic

- You are here
- Somewhere in here (undifferentiated)
- Spread of likelihood

Belief Rep. Characteristics

42

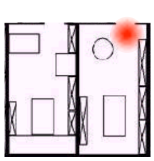
- Beliefs about where robot is can be:
  - Single hypothesis:
    "Best guess, I am here"
  - Multiple hypotheses:
    "Here or here?"
- In practice we always use probabilities.
  "Somewhere in this/these region(s)…"
- Can be continuous or discretized.



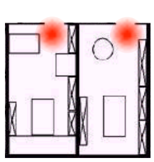
Probability & Combinations

43

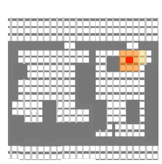
- Single or multiple
- Discrete or continuous
- These are orthogonal choices



Single Hypothesis



Multiple Hypothesis



Discrete