Celebrating EECS

MASTERWORKS *2008*

where: R&D area: level 4, Stata center (building 32)
when: May 6th, 4:30 - 6:00pm

Featuring over 35 posters across all areas of Electical Engineering & Computer Science.

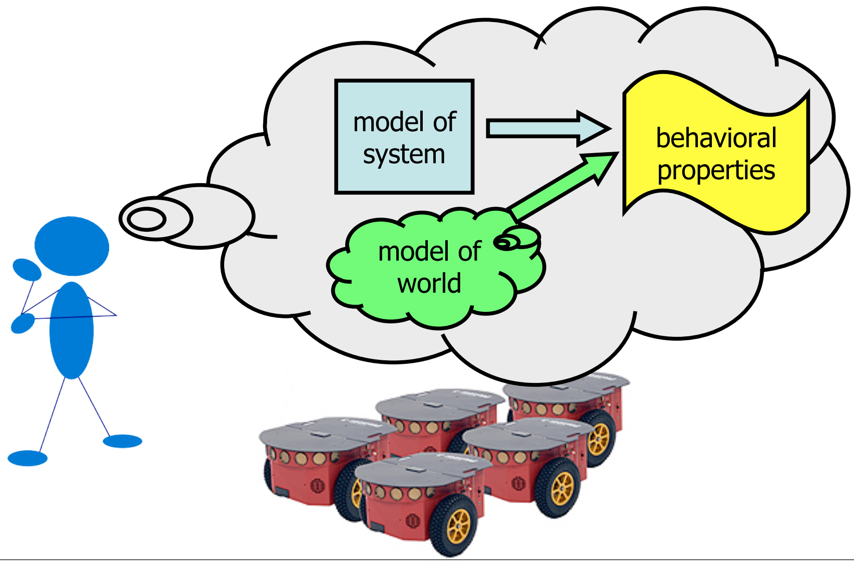
Run over to:

- *Vote for the best poster*
- *Check out a flying robot, and other exciting inventions*
- *Get free ice cream*

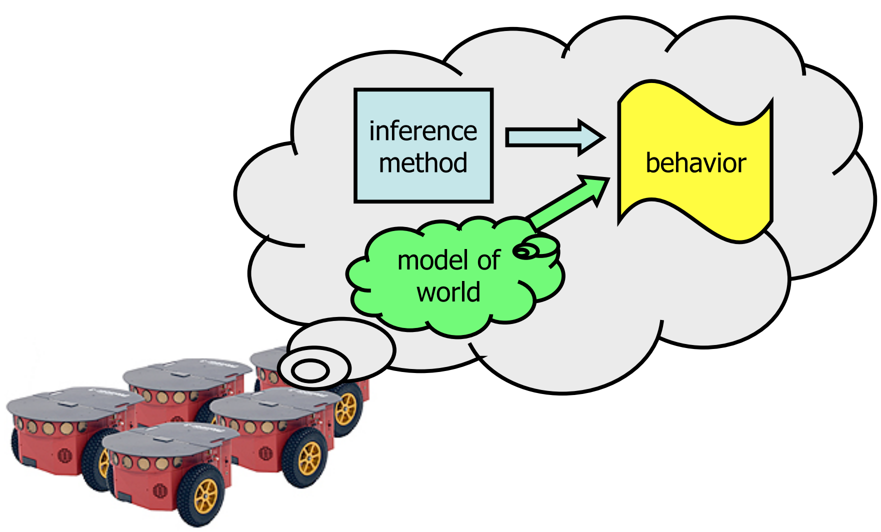
End Game

- Nano-quiz make-up night Wed May 7 from 5:30PM – 10PM in 32-123. Closed book, notes.
- Email lpk@mit.edu by 5PM on Tuesday (today) if you have an excused extra NQ to take
- Software lab write-up due this week Thu/Fri
- Design lab check-off required; no write-up
- Please do survey on the tutor
- Please do HKN review
- Practice final available on web; review sessions TBA
- Ex Camera Final Wed 21 May 1:30PM – 7:30PM
  Pick up and turn in exams in 34-501
- **We will be watching for collaboration**

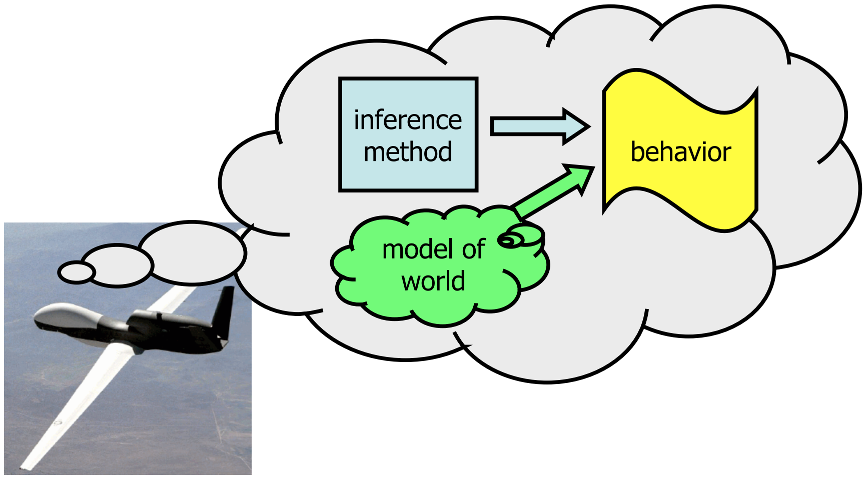
Model-based Analysis



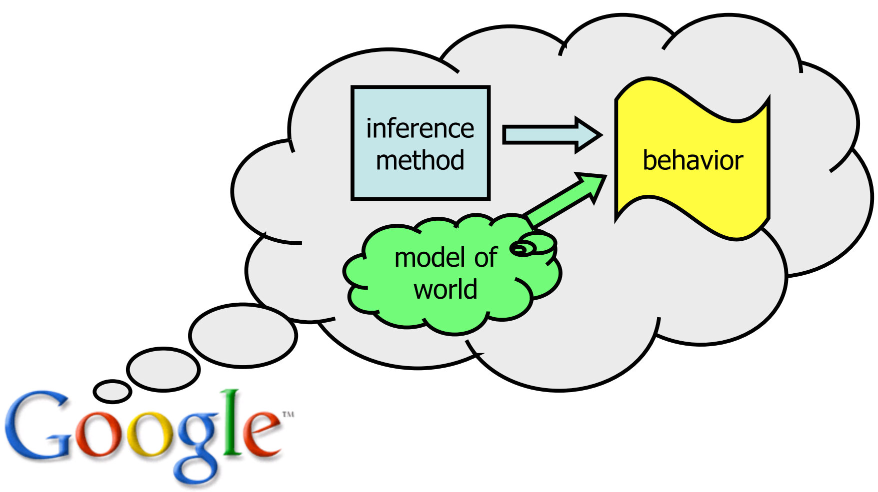
Model-based Implementation



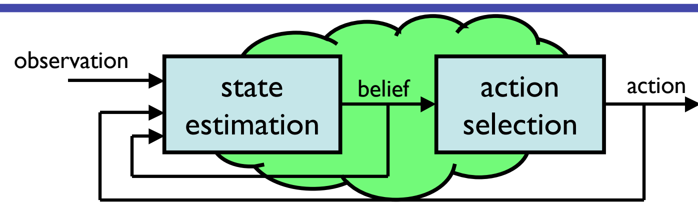
Model-based Implementation



Model-based Implementation



Model-Based Inference Problems



State estimation: What can I conclude about the current state of the world given my history of actions and observations?

Action selection: What action should I take given my current beliefs about the state of the world?

Markov Model

- States
- Actions
- Observations
- P(S(0))
- P(S(t) | S(t−1), A(t−1))
- P(O(t) | S(t))

A model is Markov if: the state contains all information relevant to predicting next state and observation

State Estimation

Given a sequence of actions and observations, what state is the system in?

Answer is a "belief state": probability distribution over states

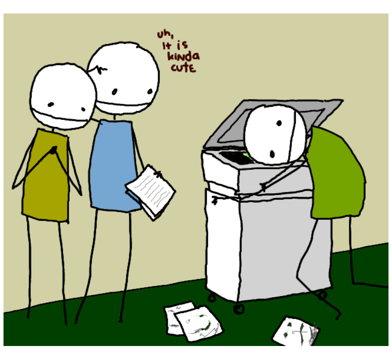
Copy Machine Problem

Initial state distribution

	s = good	bad
$P(S_0 = s)$	0.9	0.1

State transition model

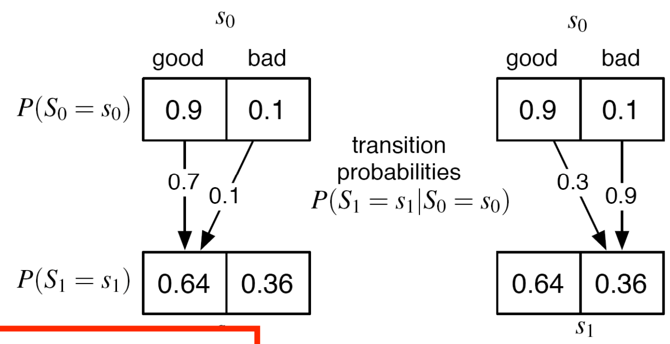
	s = good	bad
$P(S_{t+1} = s \mid S_t = good)$	0.7	0.3
$P(S_{t+1} = s \mid S_t = bad)$	0.1	0.9



Observation model

	o = perfect	smudge	black
$P(O_t = o \mid S_t = good)$	0.8	0.1	0.1
$P(O_t = o \mid S_t = bad)$	0.1	0.7	0.2

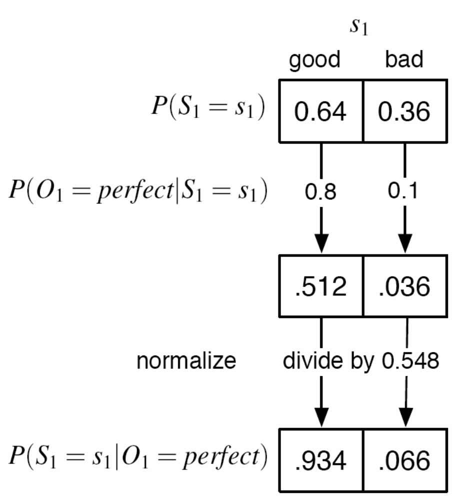
Time Passes



State transition model

	s: good	bad
$P(S_{t+1} = s \| S_t = good)$	0.7	0.3
$P(S_{t+1} = s \| S_t = bad)$	0.1	0.9

One Perfect Copy



Observation model

	o: perfect	smudge	black
$P(O_t = o \| S_t = good)$	0.8	0.1	0.1
$P(O_t = o \| S_t = bad)$	0.1	0.7	0.2

Another Day in Paradise

	good	bad
$P(S_1 = s_1 \mid O_1 = perfect)$	.934	.066
$P(S_2 = s_2 \mid S_1 = s_1)$	0.7	0.1
$P(S_2 = s_2 \mid O_1 = perfect)$	.661	.339

A Blot on our Escutcheon

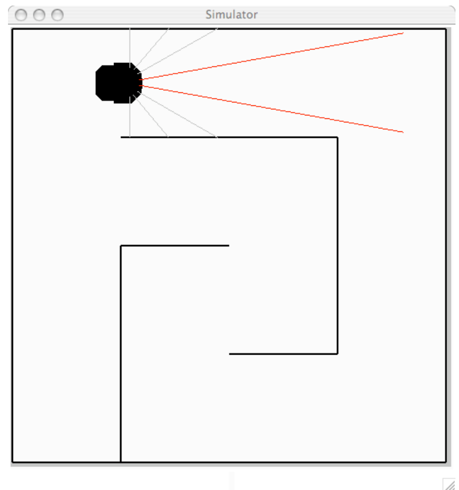
Observation model

o	perfect	smudge	black
$P(O_t = o \mid S_t = good)$	0.8	0.1	0.1
$P(O_t = o \mid S_t = bad)$	0.1	0.7	0.2

$P(S_2 = s_2 \mid O_1 = perfect)$	.661	.339
$P(O_2 = smudged \mid S_2 = s_2)$	0.1	0.7
	.066	.238
divide by .304		
$P(S_2 = s_2 \mid O_1 = perfect, O_2 = smudged)$	.218	.782

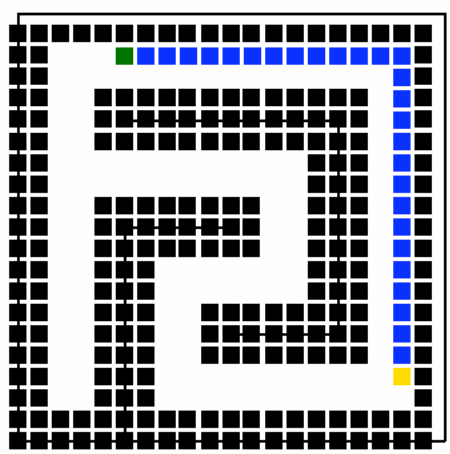
Robot Localization

States: robot poses
Actions: robot motions
Observations: sonar readings

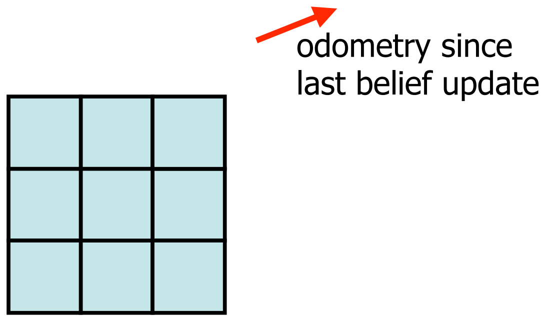


Discrete Grid

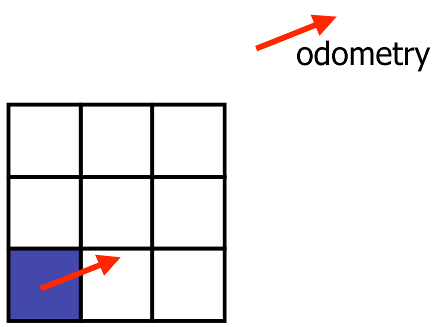
- Discretize x, y
- Discretize angle into same granularity
- Markov?
- Big!!
- But there's structure



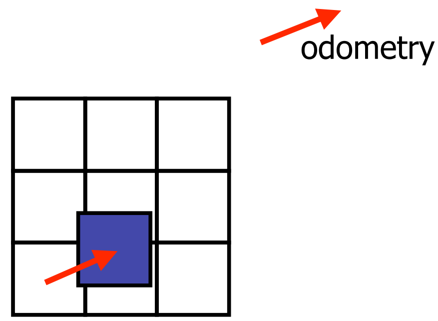
Transitions



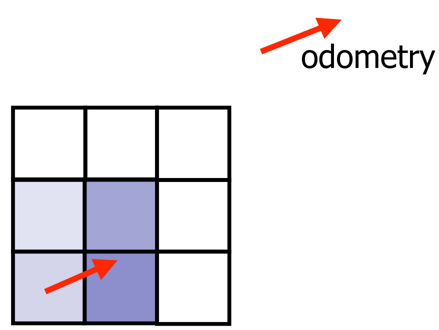
Transitions



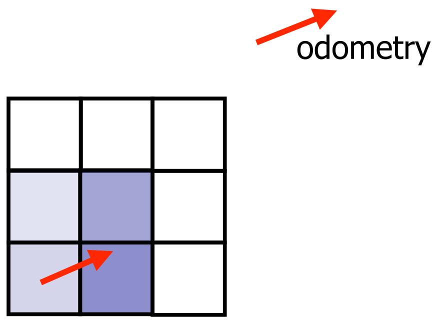
Transitions



Transitions



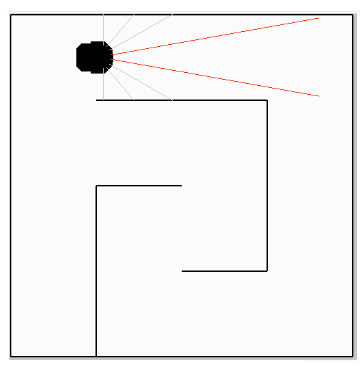
Transitions



Discretization introduces error;
should add odometry error as well

Observations

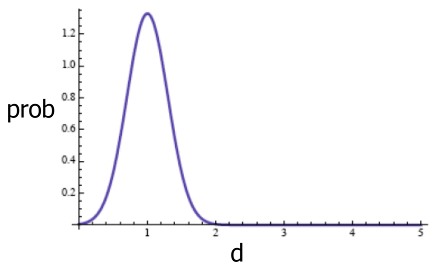
- 8 sonar readings
- model readings as independent given state



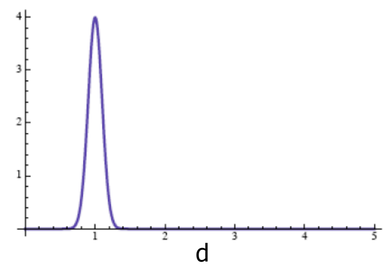
$$\begin{aligned}\Pr(\langle o_0 \ldots o_7 \rangle | x, y, \theta) &= \Pr(\langle o_0 \ldots o_7 \rangle | \langle g_0 \ldots g_7 \rangle) \\ &= \prod_{i=0}^{7} \Pr(o_i | \underbrace{g_i}_{\text{nominal reading}})\end{aligned}$$

Gaussian Distribution

$$g(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(d-m)^2/2\sigma^2}$$



mean = 1, sigma 0.3

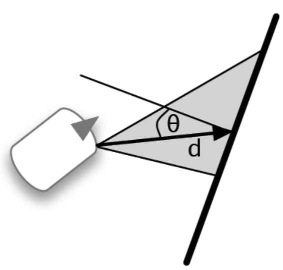


mean = 1, sigma 0.1

Sonar Modeling

"Nominal readings" assume

- scattering (good reading for any angle)
- no beam width
- robot at center of grid (x, y, th) cell

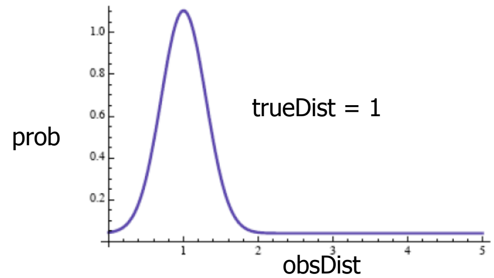


Our robots

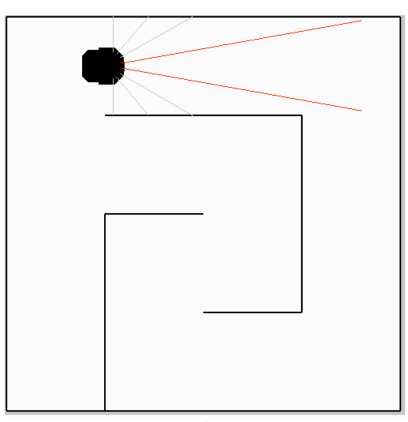
- have different max readings (5.0, 2.465)
- usually generate a max reading after 1.5 meters

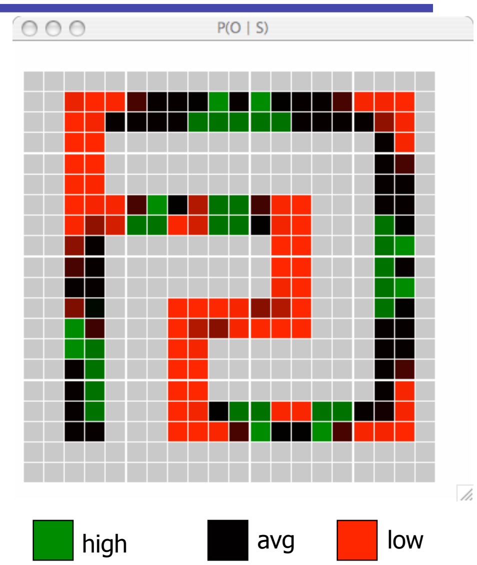
Sonar Modeling

```
def obsProb(obsDist, trueDist):
    if trueDist < maxGoodReading:
        expected = trueDist
    else:
        expected = maxReading
    predProb = gaussian(obsDist, expected, sigma)
    uniformError = (1/maxReading)
    prob = 0.9*predProb + 0.1*uniformError
    return prob
```



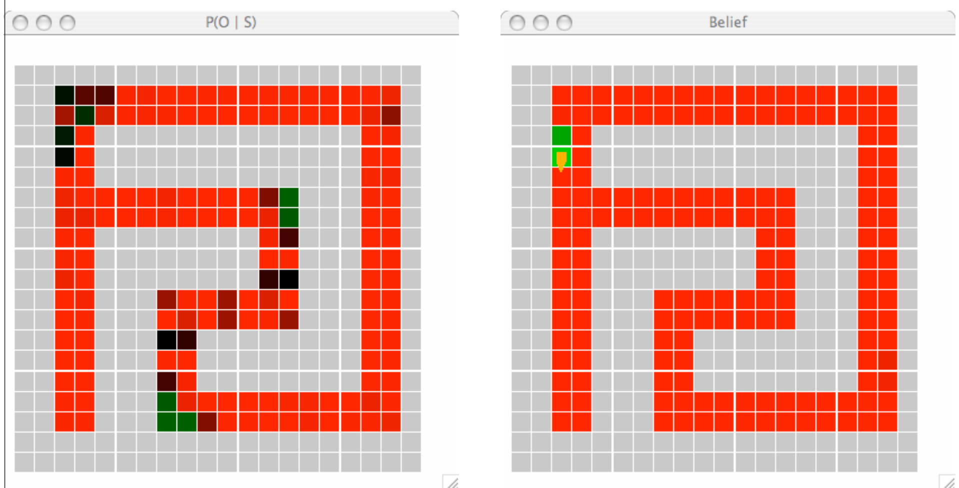
P(O | S)





State Estimation

Belief incorporates history of observations and actions



Planning

- Assume most likely state
- Use 4 headings in state representation (decreases spinning)
- Update belief after each macro action
- Replan