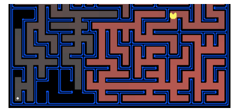
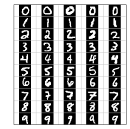
CS 188: Artificial Intelligence

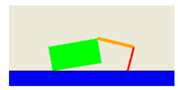
Advanced Applications:
Robotics

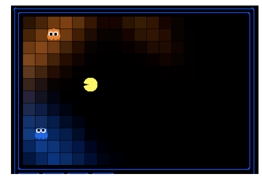
Pieter Abbeel – UC Berkeley
A few slides from Sebastian Thrun, Dan Klein

So Far Mostly Foundational Methods









Advanced Applications

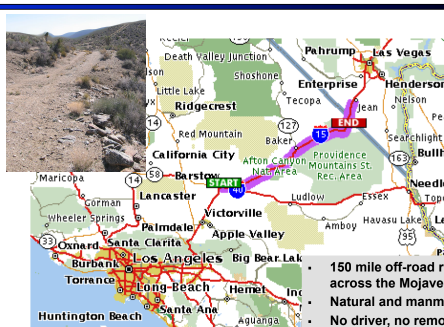
[DEMO: Race, Short]

Autonomous Vehicles

Autonomous vehicle slides adapted from Sebastian Thrun

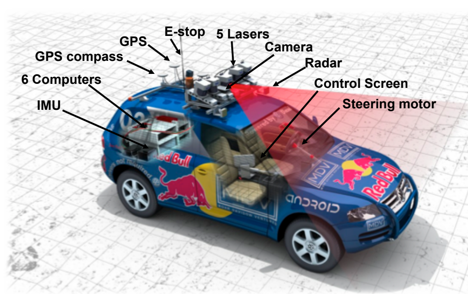
[DEMO: GC Bad, Good]

Grand Challenge: Barstow, CA, to Primm, NV

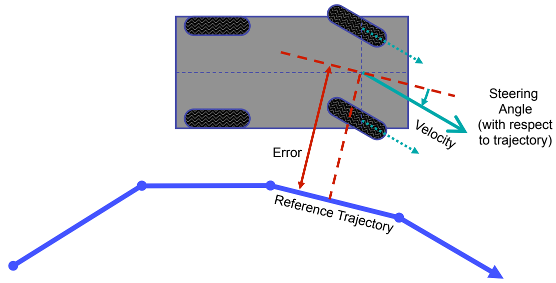


- 150 mile off-road robot race across the Mojave desert
- Natural and manmade hazards
- No driver, no remote control
- No dynamic passing

An Autonomous Car

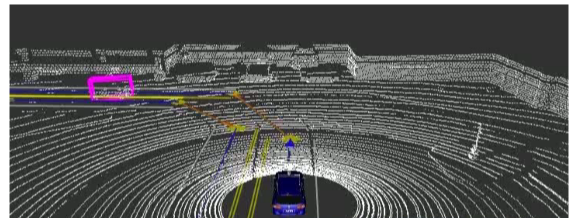


Actions: Steering Control

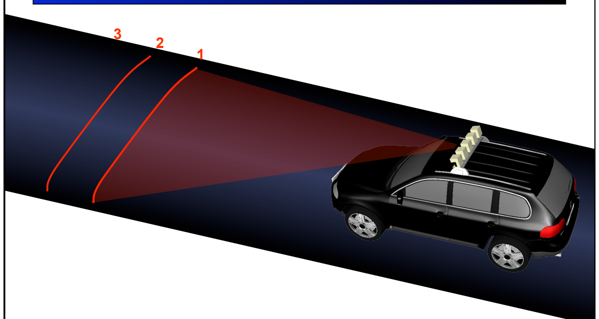


[DEMO: LIDAR]

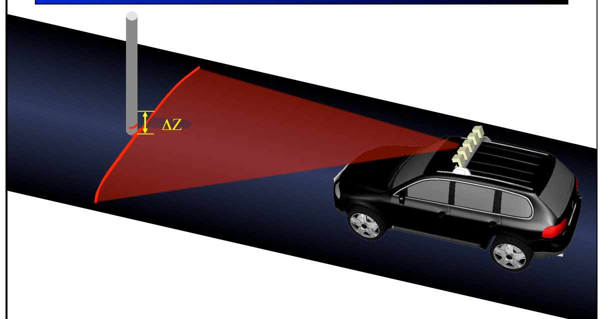
Sensors: Laser Readings



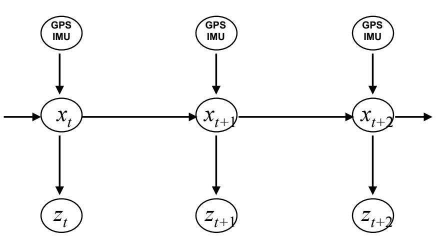
Readings: No Obstacles



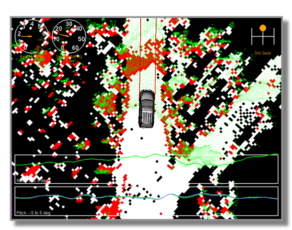
Readings: Obstacles



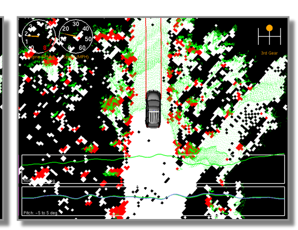
Probabilistic Error Model



HMMs for Detection



Raw Measurements: 12.6% false positives



HMM Inference: 0.02% false positives

Motivating Example

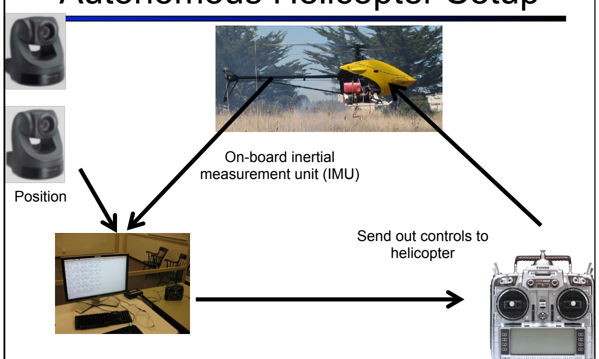
- How do we execute a task like this?

[demo: autorotate / tictoc]

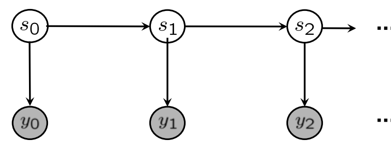
Autonomous Helicopter Flight

- Key challenges:
  - Track helicopter position and orientation during flight
  - Decide on control inputs to send to helicopter

Autonomous Helicopter Setup



HMM for Tracking the Helicopter



- State: $s = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$
- Measurements:
  - 3-D coordinates from vision, 3-axis magnetometer, 3-axis gyro, 3-axis accelerometer
- Transitions (dynamics): [time elapse update]
  - $s_{t+1} = f(s_t, a_t) + w_t$

  [f encodes helicopter dynamics]

  [w is a probabilistic noise model]

Helicopter MDP

- State: $s = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$
- Actions (control inputs):
  - a_{lon} : Main rotor longitudinal cyclic pitch control (affects pitch rate)
  - a_{lat} : Main rotor latitudinal cyclic pitch control (affects roll rate)
  - a_{coll} : Main rotor collective pitch (affects main rotor thrust)
  - a_{rud} : Tail rotor collective pitch (affects tail rotor thrust)
- Transitions (dynamics):
  - $s_{t+1} = f(s_t, a_t) + w_t$

  [f encodes helicopter dynamics]

  [w is a probabilistic noise model]

- Can we solve the MDP yet?

Problem: What's the Reward?

- Rewards for hovering: [demo: hover]

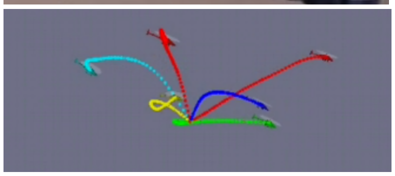
$$\begin{aligned} R(s) &= -(\alpha_x (x - x^*)^2 + \alpha_y (y - y^*)^2 + \alpha_z (z - z^*)^2 \\ &\quad + \alpha_{\dot{x}} (\dot{x} - \dot{x}^*)^2 + \alpha_{\dot{y}} (\dot{y} - \dot{y}^*)^2 + \alpha_{\dot{z}} (\dot{z} - \dot{z}^*)^2) \end{aligned}$$

- Rewards for "Tic-Toc"?
  - Problem: what's the target trajectory?
  - Just write it down by hand?

[demo: bad]

[demo: unaligned]

Helicopter Apprenticeship?



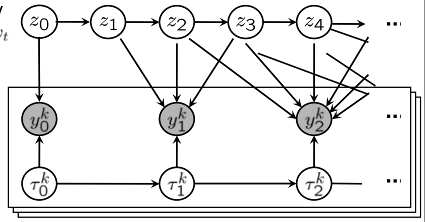
30

Probabilistic Alignment using a Bayes' Net

Intended trajectory
$z_{t+1} = f(z_t) + \omega_t$

Expert demonstrations
$y_j = z_{\tau_j} + \nu_j$

Time indices

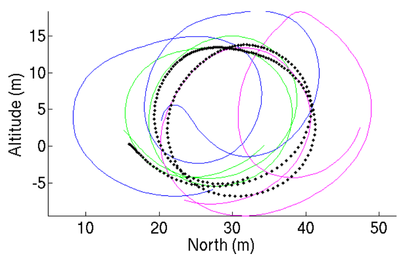


- Intended trajectory satisfies dynamics.
- Expert trajectory is a noisy observation of one of the hidden states.
  - But we don't know exactly which one.

[Coates, Abbeel & Ng, 2008]

[demo: alignment]

Alignment of Samples



- Result: inferred sequence is much cleaner!

32

[demo: airshow]

Final Behavior

33

Quadruped

- Low-level control problem: moving a foot into a new location → search with successor function ~ moving the motors
- High-level control problem: where should we place the feet?
  - Reward function R(x) = w . f(s) [25 features]

[Kolter, Abbeel & Ng, 2008]

Apprenticeship Learning

- Goal: learn reward function from expert demonstration
- Assume $R(s) = w \cdot f(s)$
- Get expert demonstrations $\mathbf{s} = (s_0, s_1, \ldots s_n)$
- Guess initial policy π_0
- Repeat:
  - Find w which make the expert better than $\{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$

    $$w_i \leftarrow \text{distinguish}\left(\pi^*, \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}\right)$$
  - Solve MDP for new weights w:

    $$\pi_i \leftarrow \text{solve}\left(MDP(w_i)\right)$$

35


Without learning


With learned reward function