Motion Planning under Uncertainty

METR 4202: Advanced Control & Robotics

Drs Surya Singh, Paul Pounds, and Hanna Kurniawati

Lecture #15

metr4202@itee.uq.edu.au
http://itee.uq.edu.au/~metr4202/

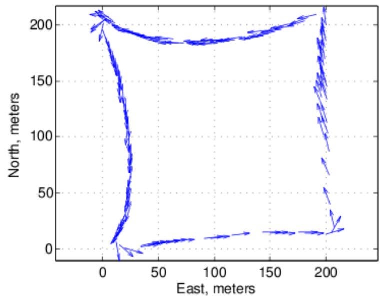
So far… wonderful world of "perfect" robotics

- Kinematics.
- Dynamics.
- Control.
- Obstacle avoidance.
- Motion planning.

Assume "perfect" robotic system

In real life, nothing is perfect…

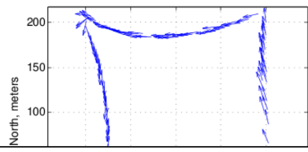
- Remember lab 1 ? ☺
- Outside disturbances cause trouble too



Censam sea trials 2009. Courtesy: T. Bandyopadhyay, L. Sarcione, & F. Hover.

In real life, nothing is perfect…

- Remember lab 1 ? ☺
- Outside disturbances cause trouble too



So…

- Previous lectures & the cool skype lecture useles ???

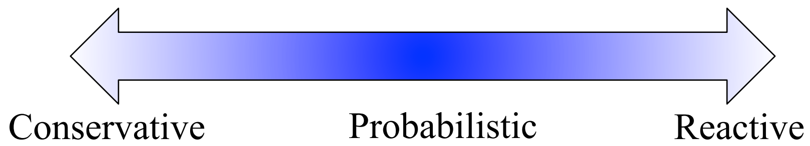
So…

- Previous lectures & the cool skype lecture useles ???
- No, depending on the task & environment, uncertainty (disturbances, errors, etc.) may be negligible.

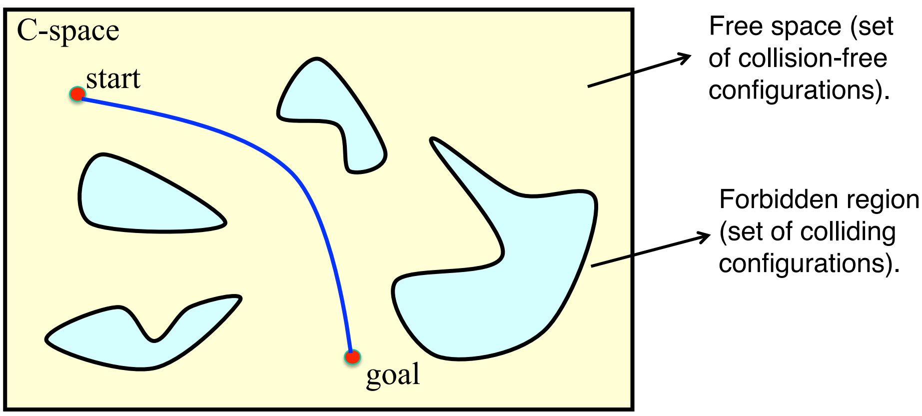
	Tasks only need low accuracy	**Tasks require high accuracy / critical tasks**
Inaccurate robotic system		
Accurate robotic system		

Today: wonderful world of ~~“perfect”~~ robotics

- Motion planning under uncertainty:
  - ❑ What is it.
  - ❑ How difficult to solve it.
  - ❑ Several approaches:

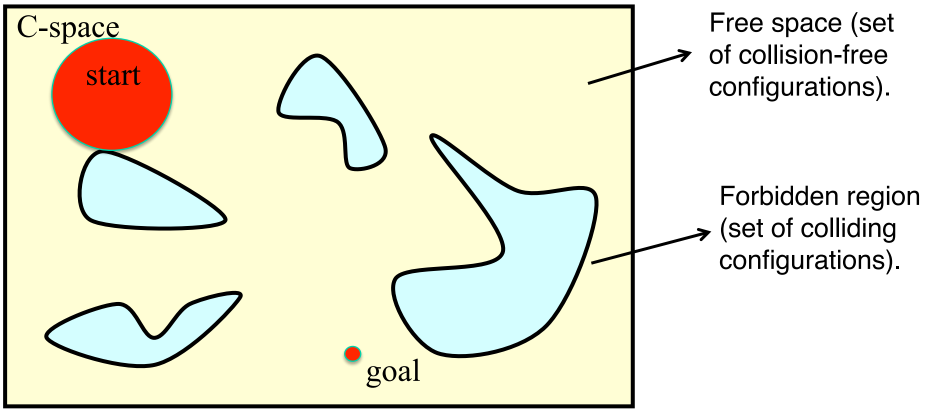


Recall: Motion planning (week-5) & skype lecture



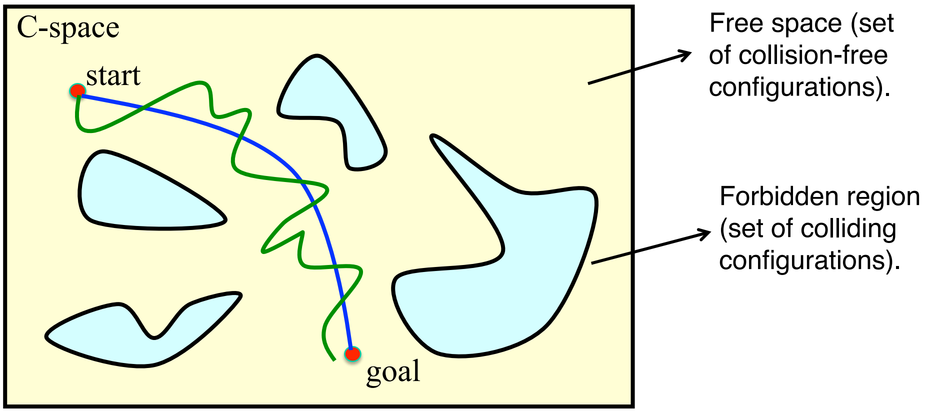
Deterministic motion planning: Find a valid path between two configurations, to accomplish a given task.

Motion planning under uncertainty



Initial is not known exactly.

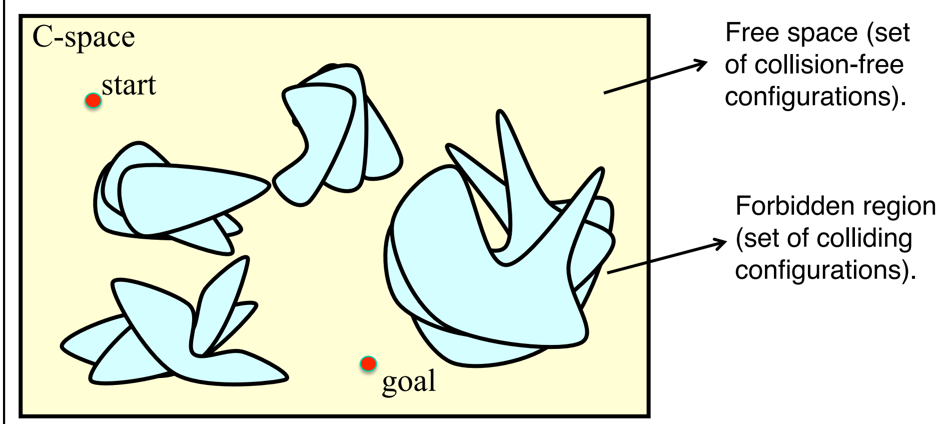
Motion planning under uncertainty



Robot's motion is erroneous, either due to system errors & noise or due to outside disturbances.

Relax the goal position.

Motion planning under uncertainty



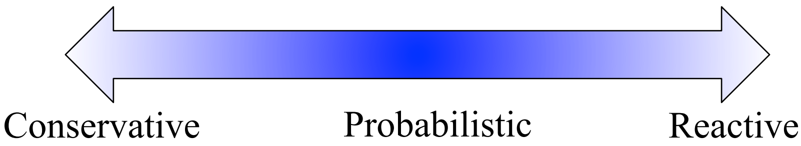
Environment is not known exactly.

Deterministic motion planning vs motion planning under uncertainty

- Deterministic motion planning
  - Find a valid path between two configurations in order to accomplish a task, given:
  - No control error.
  - No sensing.
  - Know the operating environment perfectly.
- Motion planning under uncertainty (today)
  - Find a motion strategy to accomplish a task, where there's a combination of:
  - Control error.
  - Sensing error.
  - Partially / unknown operating environment.

Today: wonderful world of ~~“perfect”~~ robotics

- In particular: Motion planning under uncertainty.
  - ✓ What is it.
  - ❑ How difficult to solve it.
  - ❑ Several approaches:

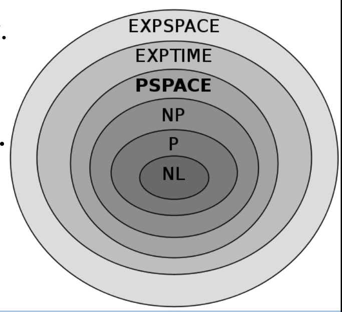


Problem hardness

- Finding a motion strategy for:
  - A point robot operating in 3D environment, where obstacles are planar walls.
  - To move from a known initial configuration to a point in a given goal region.
  - Control error: Bounded velocity error.
  - Sensing error: Bounded localization error.

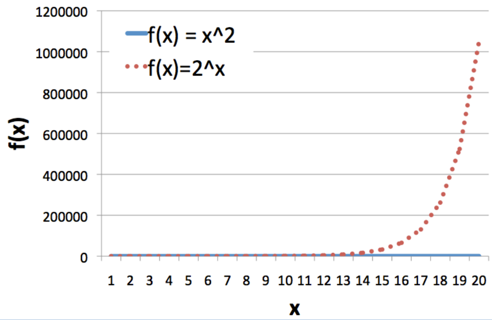
  is PSPACE-hard [Natarajan'86].

  is NEXPTIME-hard [Cany & Reif'87].



A little bit on computational complexity 1/3

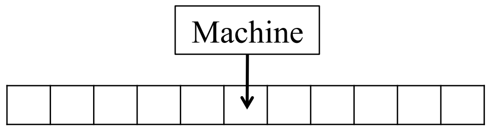
- Algorithms are **not** made to be used only once & are **not** made to be used for only one particular problem.
- How long does it take for the algorithm to find the solution when the input size increases ?
  - In particular, is it polynomial or exponential ?



A little bit on computational complexity 2/3

Today's computer

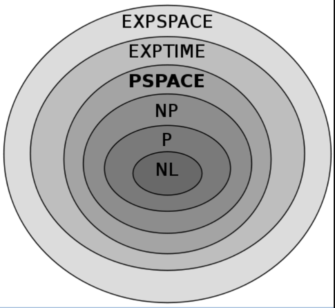
(Deterministic) Turing machine



Non-Deterministic Turing machine
It can generate multiple possible program executions at once.
Same capability as Turing machine, but can get things done faster.

A little bit on computational complexity 3/3

- P: Can be solved in polynomial time in Turing machine.
- NP: Can be solved in polynomial time on a non-deterministic Turing machine.
  → Verifiable in polynomial time in today's computer.
- PSPACE: Can be solved using polynomial space in Turing machine.



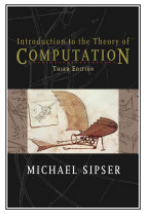
A little bit on computational complexity 3/3

- P: Can be solved in polynomial time in Turing machine.
- NP: Can be solved in polynomial time on a non-deterministic Turing machine.
  → Verifiable in polynomial time in today's computer.
- PSPACE: Can be solved using polynomial space in Turing machine.
- NEXPTIME: Can be solved in exponential time on a non-deterministic Turing machine.

A little bit on computational complexity 3/3

- P: Can be solved in polynomial time in Turing machine.
- NP: Can be solved in polynomial time on a non-deterministic Turing machine.

  → Verifiable in polynomial time in today's computer.
- PSPACE: Can be solved using polynomial space in Turing machine.
- NEXPTIME: Can be solved in exponential time on a non-deterministic Turing machine.



Want to know more ?
Introduction to the Theory of Computation
by Michael Sipser.

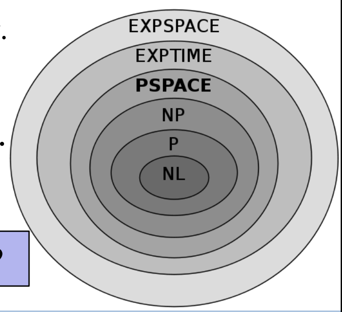
Problem hardness

- Finding a motion strategy for:
  - A point robot operating in 3D environment, where obstacles are planar walls.
  - To move from a known initial configuration to a point in a given goal region.
  - Control error: Bounded velocity error.
  - Sensing error: Bounded localization error.

  is PSPACE-hard [Natarajan'86].

  is NEXPTIME-hard [Cany & Reif'87].

  input size: number of planar walls.

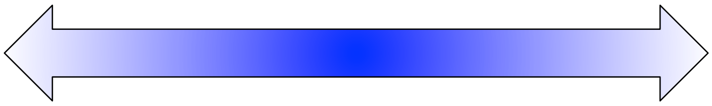


Ok, it's hard… So, what should we do ?

Today: wonderful world of ~~"perfect"~~ robotics

- In particular: Motion planning under uncertainty.
  - ✓ What is it.
  - ✓ How difficult to solve it.
  - ❑ Several approaches:

    Methods: Algorithms vs heuristics

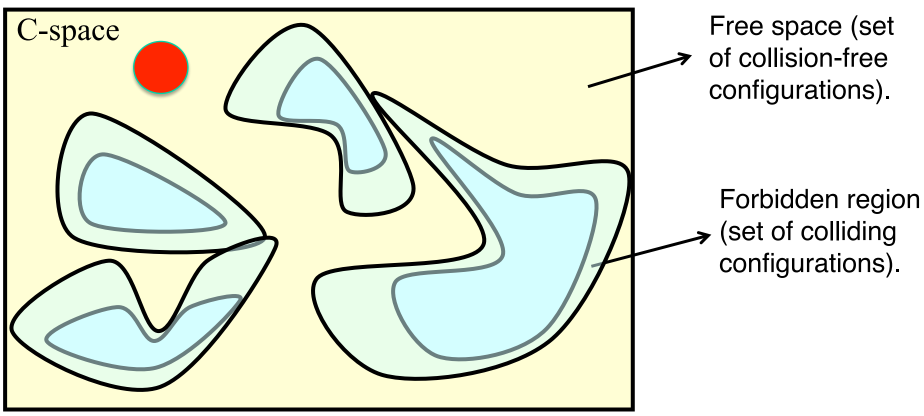


Conservative
Overestimate risk.

Probabilistic
Quantify uncertainty, to tradeoff risk w. achieving goal.

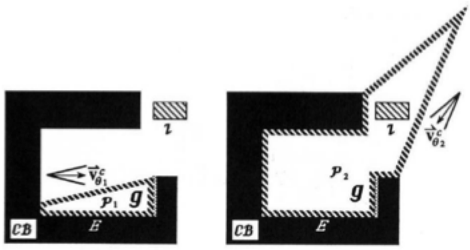
Reactive
Underestimate difficulty of achieving goal.

Simplest algorithm: Enlarging obstacles



Pre-image backchaining algorithm

- Uncertainty in motion & sensing.
  - Ability to recognize if it's in a particular region.
- Motion command: (control input, termination condition).
- Pre-image: Region of C-space, where a motion command is guarantee to reach a given goal region, recognizably.

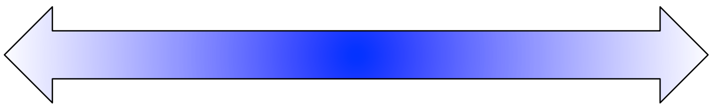


Ch. 10

Imagine…

Today: wonderful world of ~~“perfect”~~ robotics

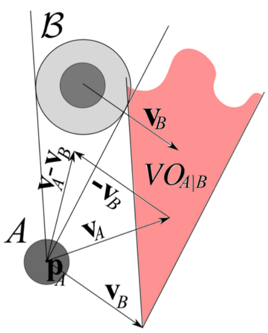
- In particular: Motion planning under uncertainty.
  - ✓ What it is.
  - ✓ How difficult to solve it.
  - ❑ Several approaches:



✓Conservative Probabilistic Reactive

Reactive greedy heuristic, based on Velocity obstacle

- Velocity obstacle



$$VO_{A|B} = \left\{ v \mid \exists t > 0 \; p_A + t(v - v_B) \in PosCol \right\}$$

p_A : center of robot A.

$PosCol$: cone region, with tip p_A, covering $B \oplus A$.

- Avoid velocity in VO, choose the one closest to goal.

In open environment…

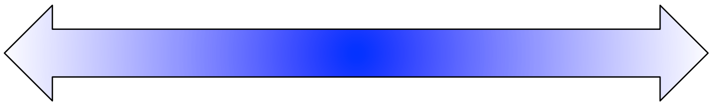
T. Bandyopadhyay & F. Hover, 2009.

Seems good, but in cluttered environment…

- From 3000 simulation runs, #success: 0.

Today: wonderful world of ~~“perfect”~~ robotics

- In particular: Motion planning under uncertainty.
  - ✓ What it is.
  - ✓ How difficult to solve it.
  - ❑ Several approaches:



✓Conservative

Probabilistic

POMDP
- Probability review.
- POMDP.
- POMDP solvers.

✓Reactive

How to quantify uncertainty ? Probability to the rescue…

Bertsekas & Tsitsiklis, Introduction to Probability.

- FATHER(F): Nurse, what is the probability that the drug will work?
- NURSE (N): I hope it works, we’ll know tomorrow.
- F: Yes, but what is the probability that it will?
- N: Each case is different, we have to wait.
- F: But let’s see, out of a hundred patients that are treated under similar conditions, how many times would you expect it to work?
- N (somewhat annoyed): I told you, every person is different, for some it works, for some it doesn’t.
- F (insisting): Then tell me, if you had to bet whether it will work or not, which side of the bet would you take?
- N (cheering up for a moment): I’d bet it will work.
- F (somewhat relieved): OK, now, would you be willing to lose two dollars if it doesn’t work, and gain one dollar if it does?
- N (exasperated): What a sick thought! You are wasting my time!

Probability review 1/4: Probabilistic Modeling

- View:
  - Experiments with random outcome.
  - Quantifiable properties of the outcome.
- Three components:
  - Sample space: Set of all possible outcomes.
  - Events: Subsets of sample space.
  - Probability: Quantify how likely an event occurs.

Probability review 2/4: Probability

- Probability: A function that maps events to real numbers satisfying these axioms:
  1. Non-negativity: $P(E) \geq 0$, where E is an event.
  2. Normalization: $P(S) = 1$, where S is the sample space.
  3. Additivity of finite / countably infinite events.

$$P\left(\bigcup_{i=1}^{\infty/n} E_i\right) = \sum_{i=1}^{\infty/n} P\left(E_i\right),$$

where E_i are disjoint / mutually exclusive, i : natural number.

Probability review 3/4: Random Variables

- Interest is on numerical values associated w. samples, e.g.:
  - Sample 50 students enrolled in METR4202, what's the major of most of the students.
  - Roll a fair dice, get \$5 if the outcome is even, & loose \$5 if the outcome is odd.
- Random variable X is a function $X : S \to Num$.
  - Num: countable set (e.g., integer) → discrete random variable.
  - Num: uncountable set (e.g., real) → continuous random variable.

Probability review 4/4: Characterizing Random Variables

- Cumulative distribution function (cdf)

$$F_X(x) = P(X \le x) = P\left(\left\{s \middle| X(s) \le x, s \in S\right\}\right)$$

- Discrete: Probability mass function (pmf)

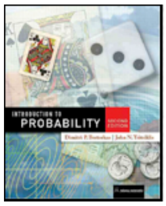
$$f_X[x] = P(X = x)$$

- Continuous: Probability density function/probability distribution function (pdf)

$$f_X(x) = \frac{dF_X(x)}{dx} \quad ; \quad P(a \le X \le b) = \int_a^b f_X(x)dx$$

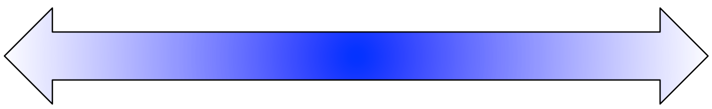
For more on probability…

- STAT2202 (compulsary for Mechatronic students).
- ENGG7302:
  - http://itee.uq.edu.au/~engg7302/material/stocProc/lecSP01.pdf
  - http://itee.uq.edu.au/~engg7302/material/stocProc/lecSP02.pdf
- Introduction to Probability by Bertsekas & Tsitsiklis.



Today: wonderful world of ~~“perfect”~~ robotics

- In particular: Motion planning under uncertainty.
  - ✓ What it is.
  - ✓ How difficult to solve it.
  - ❑ Several approaches:



✓Conservative

Probabilistic

POMDP

✓Probability review.

- POMDP.
- POMDP solvers.

✓Reactive

POMDP:
Partially Observable Markov Decision Processes

- Main components:
  - State space (S).
  - Action space (A).
  - Observation space (Ω).

POMDP: Modeling motion error

- Main components:
  - State space (S).
  - Action space (A).
  - Observation space (Ω).
  - Transition function

    $T(s, a, s') =$

    $P(S_{t+1} = s' \mid S_t = s, A_t = a)$

POMDP: Modeling sensing error

- Main components:
  - State space (S).
  - Action space (A).
  - Observation space (Ω).
  - Transition function: T(s, a, s').
  - Observation function
    Z(s, a, o) =
    $P(\Omega_t = o \mid S_t = s, A_t = a)$

POMDP: Modeling objective function

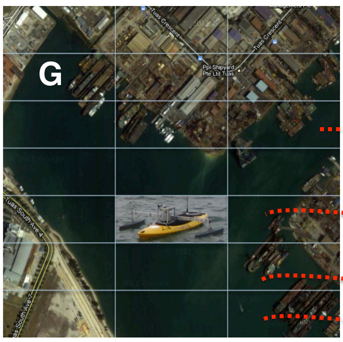
- Main components:
  - State space (S).
  - Action space (A).
  - Observation space (Ω).
  - Transition function T(s, a, s').
  - Observation function Z(s, a, o).
  - Reward function
    R(s, a,): Reward received when the robot performs action a from state s.

POMDP: key points

- Main components:
  - State space (S).
  - Action space (A).
  - Observation space (Ω).
  - Transition function T(s, a, s').
  - Observation function Z(s, a, o).
  - Reward function R(s, a).

Not known

POMDP: key points



0.1	...	0.2	0.07	0.03

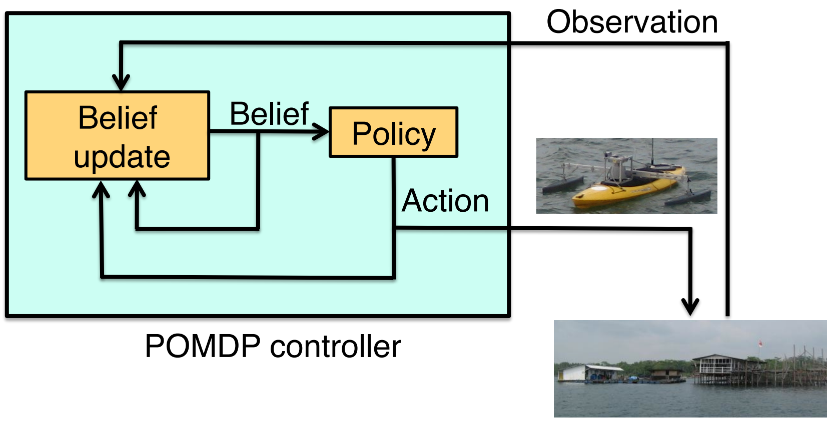
0.3	...	0.1	0.01	0.2

⋮

Belief

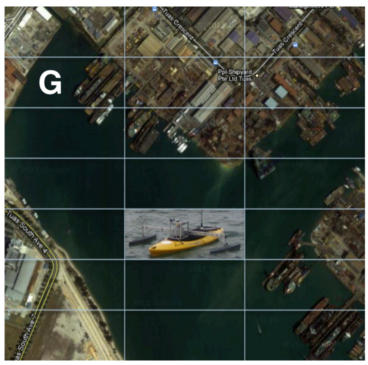
- Belief: Distribution over states.
- Belief space: The set of all possible beliefs.
- Policy: Mapping from beliefs to actions.
- Goal: Find optimal policy.

POMDP policy: Usage



Belief update

- Recall POMDP definition:
  - State space (S).
  - Action space (A).
  - Observation space (Ω).
  - Transition function: T(s, a, s').
  - Observation function: Z(s, a, o).
  - Reward function: R(s, a).
- If the robot is currently at belief b_t, what is its belief after performing action a_t of A, and perceiving observation o_t of Ω ?

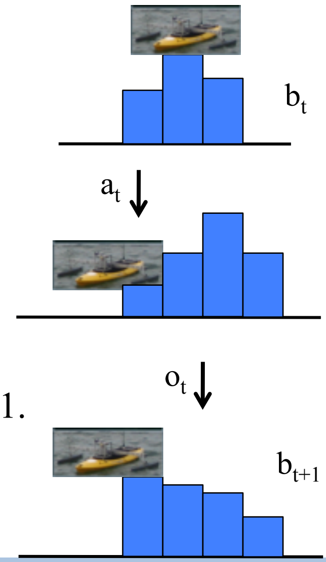


Belief update

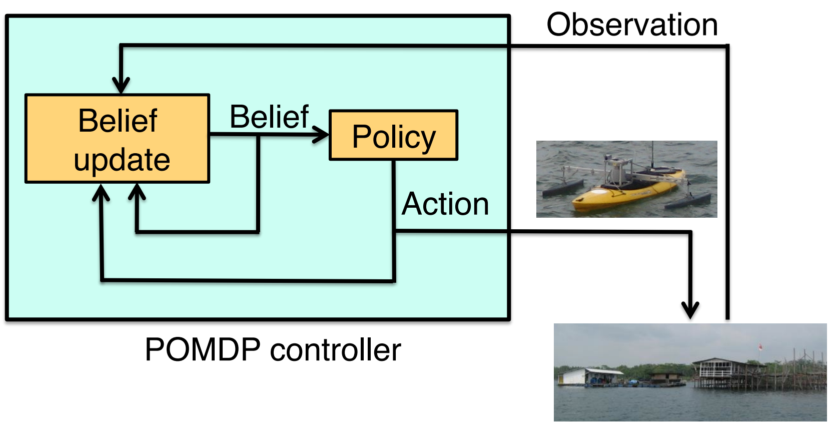
- Current belief b_t can be updated to a new belief $b_{t+1} = T(b_t, a_t, o_t)$, after the robot performs a_t and perceives o_t using:

$$b_{t+1}(s') = \frac{Z(s', a_t, o_t) \sum_{s \in S} T(s, a_t, s') b_t(s)}{P(o_t | a_t, b_t)}$$

$P(o_t | a_t, b_t)$ can be treated as a normalizing constant, s.t. b_{t+1} sum to 1.



POMDP policy: Usage



Policy representations

- Set of tuples of (belief, action).
- Policy graph, policy tree.
- Function (set of α-vectors).

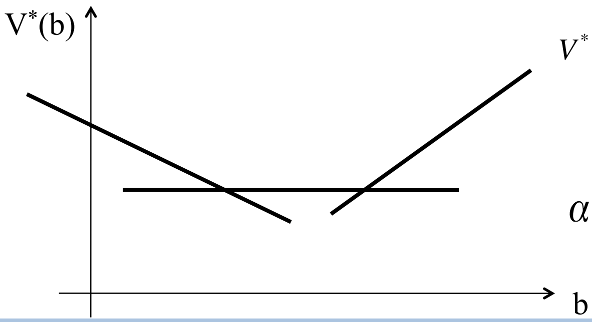
Optimal policy

- Each policy induces a value for each belief.
- The value $V_\pi(b)$ of belief b when following policy π is the expected total reward received if the robot starts from b and follows policy π.
- Optimal policy π^*: For all beliefs b, $V_{\pi^*}(b) \geq V_\pi(b)$ for any policy π.
- The value function V_{π^*} is called the optimal value function.

Properties of optimal value function

Smallwood & Sondik'73

- Optimal value function:
  - Finite horizon: Piecewise linear convex.
  - Infinite horizon: Can be approximated arbitrarily closely with a piecewise linear convex function.
  - Convex: $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$



$$V^*(b) = \max_{\alpha \in \Gamma} (\alpha \cdot b)$$
$$= \max_{\alpha \in \Gamma} \left(\sum_i \alpha(s_i) \cdot b(s_i) \right)$$

α : gradient of a line.

Computing optimal policy

- Dynamic programming to construct optimal value function: Value iteration algorithm.
- Starts from computing the optimal value for 1 step.
- Subsequently compute the optimal value for step-2, step-3, …, step-n for all beliefs, using:

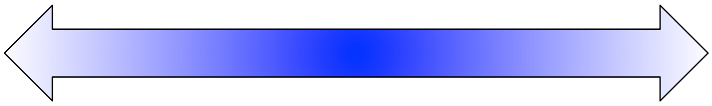
Bellman update

$$V_i^*(b) = HV_{i-1}(b)$$
$$= \max_a R(b,a) + \gamma \sum_o P(o \mid b,a) V_{i-1}^*(T(b,a,o))$$

Problem: Finding the optimal policy is PSPACE-hard.

Today: wonderful world of ~~“perfect”~~ robotics

- In particular: Motion planning under uncertainty.
  - ✓ What it is.
  - ✓ How difficult to solve it.
  - ❑ Several approaches:



✓Conservative

Probabilistic

POMDP

✓Probability review.

✓POMDP.

- POMDP solvers.

✓Reactive

QMDP heuristic

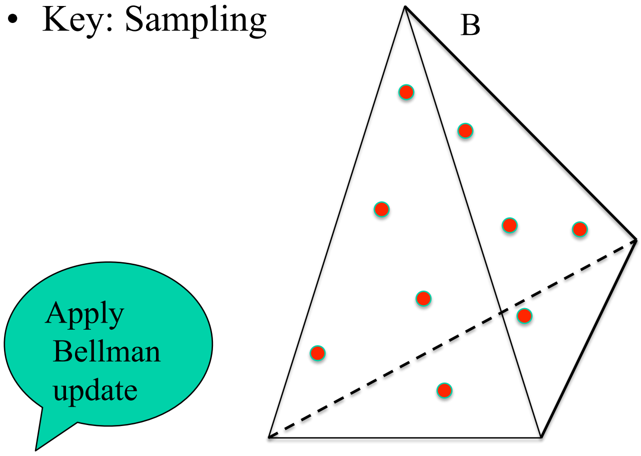
- Use a special case of POMDP –called MDP (Markov Decision Processes)–
  - Only motion uncertainty, no localization uncertainty.
- Assume that after one step, state uncertainty is gone.

$$Q_{MDP}\left(s,a\right) = R(s,a) + \gamma \sum_{s'} T\left(s,a,s'\right) V(s')$$

$$Q\left(b,a\right) = \sum_{s} b(s) Q_{MDP}(s,a)$$

Point-based approach

- Trade optimality with approximate optimality for speed.
- Key: Sampling



Plan only w.r.t. the representative set of sampled beliefs.

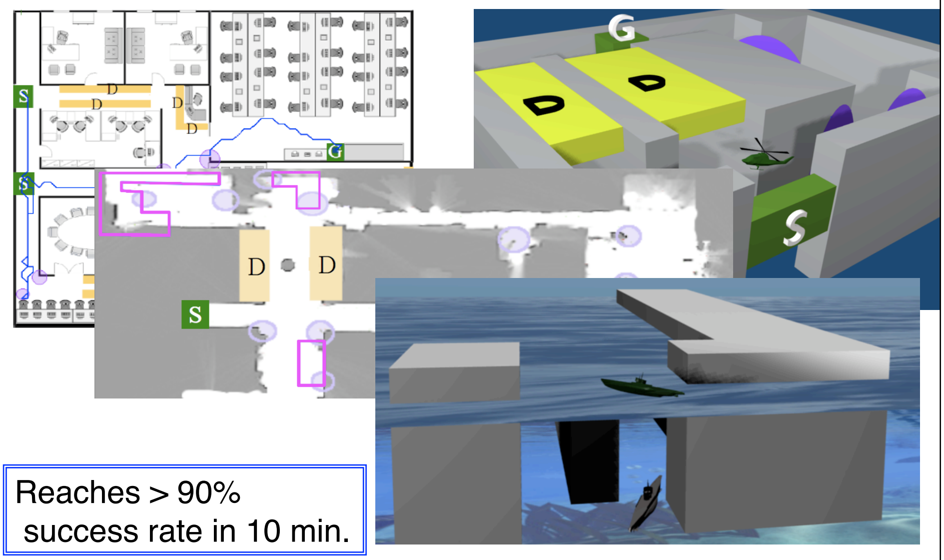
Point-based approach

- Interleave
  - Sample beliefs.
    - PBVI (Pineau, et.al. '03), HSVI 1 (Smith & Simmons '03), Perseus (Spaan & Vlassis'05), HSVI2 (Smith & Simmons '05), SARSOP (Kurniawati, et.al. '08).
  - Computing Bellman updates.
- Guarantee to converge to the optimal policy as # sampled beliefs goes to inf.

Thanks to point-based approach

12 states → 870 states

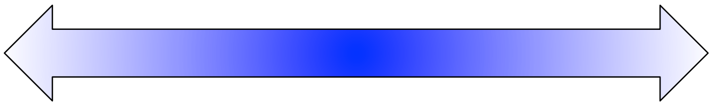
Thanks to point-based approach



Reaches > 90% success rate in 10 min.

Today: wonderful world of ~~“perfect”~~ robotics

- Motion planning under uncertainty:
  - ✓ What is it.
  - ✓ How difficult to solve it.
  - ✓ Several approaches:



✓Conservative
Overestimate risk.

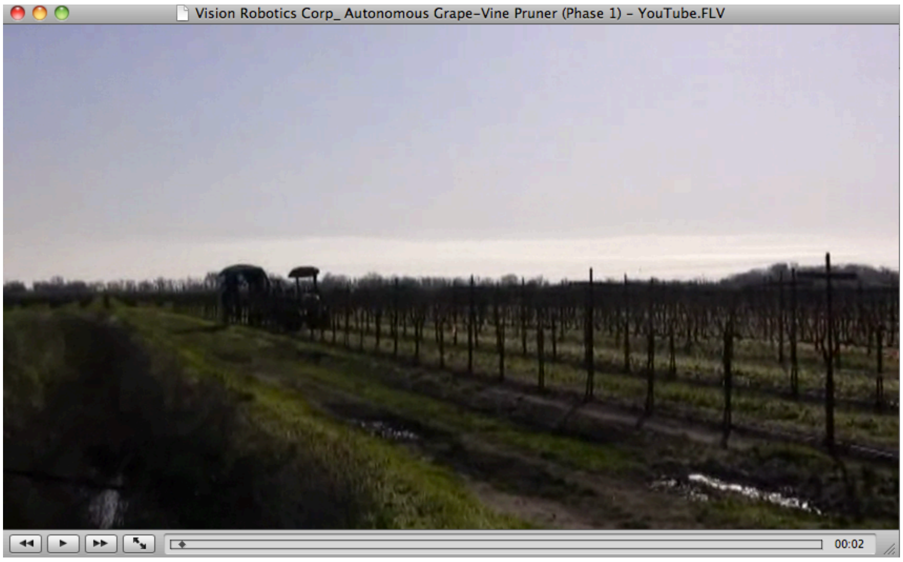
✓Probabilistic
Quantify uncertainty, to tradeoff risk w. achieving goal.

✓Reactive
Underestimate difficulty of achieving goal.

Myths… I hope you **don’t** buy after this class ☺

- No need to work on algorithms… Just build better faster computers…
- The maths in algorithm design/analysis:
  - For show-off…
  - To compensate lack of English ability…

Video share


Vision Robotics Corp_ Autonomous Grape-Vine Pruner (Phase 1) - YouTube.FLV
00:02