Introducing Uncertainty

(It is not the world that is imperfect, it is our knowledge of it)

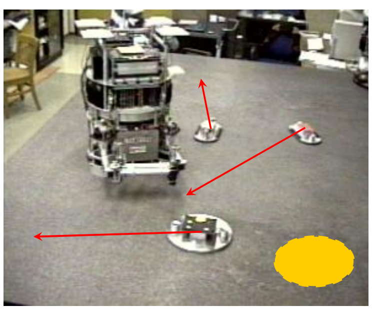
R&N: Chap. 3, Sect 3.6 + Chap. 13

1

- So far, we have assumed that:
  - World states are perfectly observable, → the current state is exactly known
  - Action representations are perfect, → states are exactly predicted
- We will now investigate how an agent can cope with imperfect information
- We will also study how limited resources (mainly time) affect reasoning
- Occasionally, we will consider cases where the world is dynamic

2

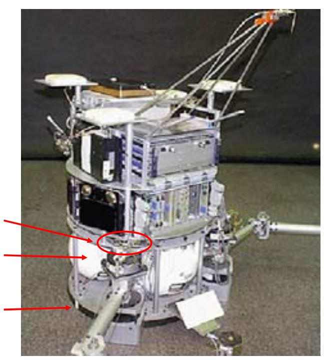
Introductory Example



A robot with imperfect sensing must reach a goal location among moving obstacles (dynamic world)

3

Robot created at Stanford's ARL Lab to study issues in robot control and planning in no-gravity space environment



air thrusters

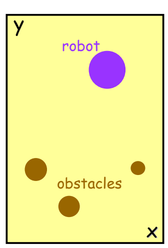
gas tank

air bearing

4

Model, Sensing, and Control

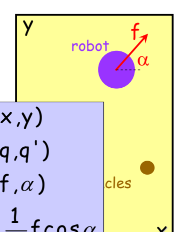
- The robot and the obstacles are represented as disks moving in the plane
- The position and velocity of each disc are measured by an overhead camera every 1/30 sec



5

Model, Sensing, and Control

- The robot and the obstacles are represented as disks moving in the plane
- The position and velocity of each disc are measured by an overhead camera within 1/30 sec
- The robot controls the magnitude f and the orientation α of the total pushing force exerted by the thrusters



$q=(x,y)$

$s=(q,q')$

$u=(f,\alpha)$

$x''=\frac{1}{m}f\cos\alpha$

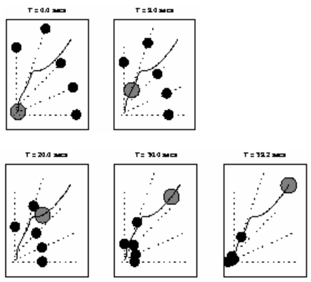
$y''=\frac{1}{m}f\sin\alpha$

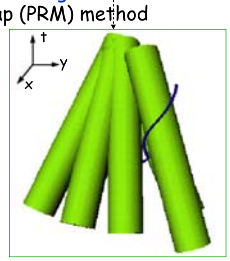
$f \le f_M$

6

Motion Planning

The robot plans its trajectories in configuration×time space using a probabilistic roadmap (PRM) method





Obstacle map to cylinders in configuration×time space

But executing this trajectory is likely to fail ...

1) The measured velocities of the obstacles are inaccurate
2) Tiny particles of dust on the table affect trajectories and contribute further to deviation
   → Obstacles are likely to deviate from their expected trajectories
3) Planning takes time, and during this time, obstacles keep moving
   → The computed robot trajectory is not properly synchronized with those of the obstacles

➔ The robot may hit an obstacle before reaching its goal

[Robot control is not perfect but "good" enough for the task]

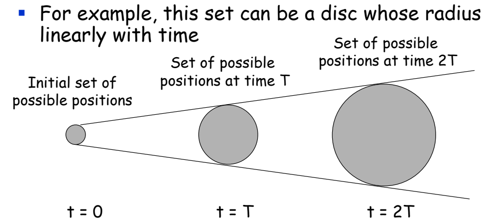
But executing this trajectory is likely to fail ...

1) The measured velocities of the obstacles are inaccurate
2) Tiny particles of dust on the table affect trajectories and contribute further to deviation
   → Obstacles are likely to deviate from their expected trajectories
3) Planning takes time, and during this time, obstacles are moving
   → The computed robot trajectory is not properly synchronized with those of the obstacles

Planning must take both uncertainty in world state and time constraints into account

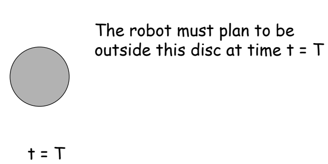
Dealing with Uncertainty

- The robot can handle uncertainty in an obstacle position by representing the set of all positions of the obstacle that the robot think possible at each time (belief state)
- For example, this set can be a disc whose radius grows linearly with time



Dealing with Uncertainty

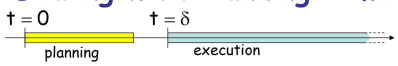
- The robot can handle uncertainty in an obstacle position by representing the set of all positions of the obstacle that the robot think possible at each time (belief state)
- For example, this set can be a disc whose radius grows linearly with time



Dealing with Uncertainty

- The robot can handle uncertainty in an obstacle position by representing the set of all positions of the obstacle that the robot think possible at each time (belief state)
- For example, this set can be a disc whose radius grows linearly with time
- The forbidden regions in configuration×time space are cones, instead of cylinders
- The trajectory planning method remains essentially unchanged

Dealing with Planning Time



- Let t=0 the time when planning starts. A time limit δ is given to the planner
- The planner computes the states that will be possible at $t = \delta$ and use them as the possible initial states
- It returns a trajectory at some $t < \delta$, whose execution will start at $t = \delta$
- Since the PRM planner isn't absolutely guaranteed to find a solution within δ, it computes two trajectories using the same roadmap: one to the goal, the other to any position where the robot will be safe for at least an additional δ. Since there are usually many such positions, the second trajectory is at least one order of magnitude faster to compute

13

Are we done?

- Not quite !
- The uncertainty model may itself be incorrect, e.g.:
  - There may be more dust on the table than anticipated
  - Some obstacles have the ability to change trajectories
- But if we are too careful, we will end up with forbidden regions so big that no solution trajectory will exist any more
- So, it might be better to take some "risk"

→ The robot must monitor the execution of the planned trajectory and be prepared to re-plan a new trajectory

14

Are we done?

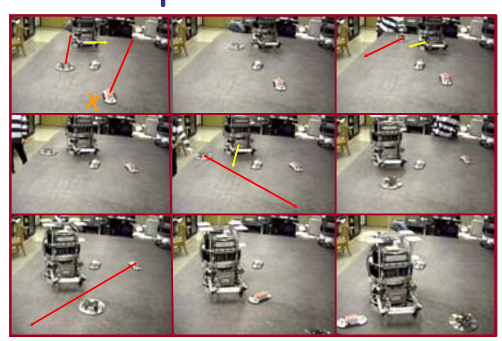
If an obstacle has an unexpected position, the planner is called back to compute a new trajectory.

Execution monitoring consists of using the camera (at 30Hz) to verify that all obstacles are at positions allowed by the robot's uncertainty model

→ The robot must monitor the execution of the planned trajectory and be prepared to re-plan a new trajectory

15

Experimental Run



Total duration : 40 sec

16

Experimental Run

17

Is this guaranteed to work?

Of course not :

- Thrusters might get clogged
- The robot may run out of air or battery
- The granite table may suddenly break into pieces
- Etc ...

[Unbounded uncertainty]

18

Sources of Uncertainty

The Real World and its Representation

3x3 matrix filled with 1, 2, .., 8, and 'empty'

Agent's conceptualization
(→ representation language)

Real world

8-puzzle

The Real World and its Representation

Logic sentences using propositions like Block(A), On(A,B), Handempty, ... and connectives

Agent's conceptualization
(→ representation language)

Real world

Blocks world

The Real World and its Representation

Geometric models and equations of motion

Agent's conceptualization
(→ representation language)

Real world

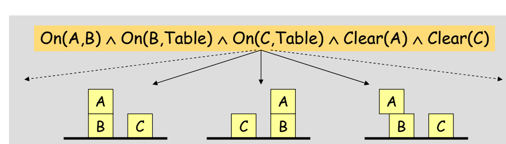
Air-bearing robot navigating among moving obstacles

Who provides the representation language?

- The agent's designer
- As of today, no practical techniques exist allowing an agent to autonomously abstract features of the real world into useful concepts and develop its own representation language using these concepts
- Inductive learning techniques are steps in this direction, but much more is needed
- The issues discussed in the following slides arise whether the representation language is provided by the agent's designer or developed over time by the agent

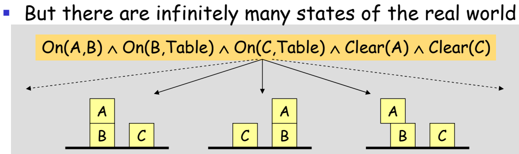
First Source of Uncertainty:
The Representation Language

- There are many more states of the real world than can be expressed in the representation language
- So, any state represented in the language may correspond to many different states of the real world, which the agent can't represent distinguishably



First Source of Uncertainty: The Representation Language

- 6 propositions On(x,y), where x, y = A, B, C and $x \neq y$
- 3 propositions On(x,Table), where x = A, B, C
- 3 propositions Clear(x), where x = A, B, C
- At most 2^{12} states can be distinguished in the language [in fact much fewer, because of state constraints such as On(x,y) → ¬On(y,x)]
- But there are infinitely many states of the real world



→ An action representation may be incorrect ...

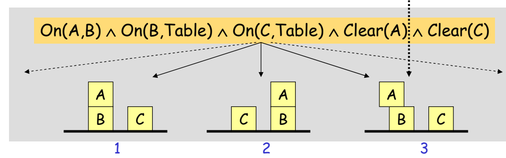
Stack(C,A)

P = Holding(C) ∧ Block(C) ∧ Block(A) ∧ Clear(A)

D = Clear(A), Holding(C)

A = On(C,A), Clear(C), Handempty

is likely not to have the described effects in case 3 because the precondition is "incomplete"



... or may describe several alternative effects

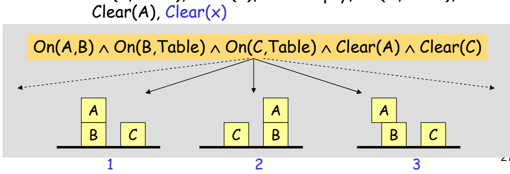
Stack(C,A)

P = Holding(C) ∧ Block(C) ∧ Block(A) ∧ Clear(A)
[If On(A,x) ∧ ($x \neq$ Table)]

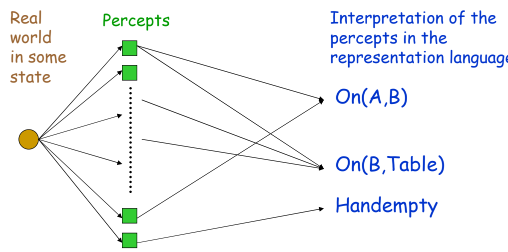
E_1 { D = Clear(A), Holding(C)
A = On(C,A), Clear(C), Handempty

OR

E_2 { D = Holding(C), On(A,x)
A = On(C,Table), Clear(C), Handempty, On(A,Table), Clear(A), Clear(x)



Observation of the Real World

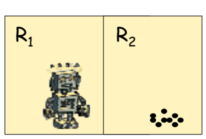


Percepts can be user's inputs, sensory data (e.g., image pixels), information received from other agents, ...

Second source of Uncertainty: Imperfect Observation of the World

Observation of the world can be:

- Partial, e.g., a vision sensor can't see through obstacles (lack of percepts)

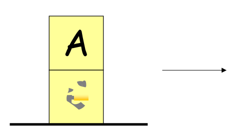


The robot may not know whether there is dust in room R2

Second source of Uncertainty: Imperfect Observation of the World

Observation of the world can be:

- Partial, e.g., a vision sensor can't see through obstacles
- Ambiguous, e.g., percepts have multiple possible interpretations



On(A,B) ∨ On(A,C)

Second source of Uncertainty: Imperfect Observation of the World

Observation of the world can be:

- Partial, e.g., a vision sensor can't see through obstacles
- Ambiguous, e.g., percepts have multiple possible interpretations
- Incorrect

Third Source of Uncertainty: Ignorance, Laziness, Efficiency

- An action may have a long list of preconditions, e.g.:
  Drive-Car:
  P = Have(Keys) $\wedge$ $\neg$Empty(Gas-Tank) $\wedge$ Battery-Ok $\wedge$ Ignition-Ok $\wedge$ $\neg$Flat-Tires $\wedge$ $\neg$Stolen(Car) ...
- The agent's designer may ignore some preconditions ... or by laziness or for efficiency, may not want to include all of them in the action representation
- The result is a representation that is either incorrect – executing the action may not have the described effects – or that describes several alternative effects

Representation of Uncertainty

- Many models of uncertainty
- We will consider two important models:
  - Non-deterministic model:
    Uncertainty is represented by a set of possible values, e.g., a set of possible worlds, a set of possible effects, ...
    → The next two lectures
  - Probabilistic model:
    Uncertainty is represented by a probabilistic distribution over a set of possible values
    → The following two lectures

Example: Belief State

- In the presence of non-deterministic sensory uncertainty, an agent belief state represents all the states of the world that it thinks are possible at a given time or at a given stage of reasoning

- In the probabilistic model of uncertainty, a probability is associated with each state to measure its likelihood to be the actual state

What do probabilities mean?

- Probabilities have a natural frequency interpretation
- The agent believes that if it was able to return many times to a situation where it has the same belief state, then the actual states in this situation would occur at a relative frequency defined by the probabilistic distribution

This state would occur 20% of the times

Example

- Consider a world where a dentist agent D meets a new patient P
- D is interested in only one thing: whether P has a cavity, which D models using the proposition Cavity
- Before making any observation, D's belief state is:

Cavity	$\neg$ Cavity
p	1-p

- This means that D believes that a fraction p of patients have cavities

Where do probabilities come from?

- Frequencies observed in the past, e.g., by the agent, its designer, or others
- Symmetries, e.g.:
  - If I roll a dice, each of the 6 outcomes has probability 1/6
- Subjectivism, e.g.:
  - If I drive on Highway 280 at 120mph, I will get a speeding ticket with probability 0.6
  - Principle of indifference: If there is no knowledge to consider one possibility more probable than another, give them the same probability

37

Non-Deterministic vs. Probabilistic

- If the world is adversarial and the agent uses probabilistic methods, it is likely to fail consistently
- If the world in non-adversarial and failure must be absolutely avoided, then non-deterministic techniques are likely to be more efficient computationally
- In other cases, probabilistic methods may be a better option, especially if there are several “goal” states providing different rewards and life does not end when one is reached

38

Uncertainty and Errors

- The uncertainty model may itself be incorrect
- Representing uncertainty can reduce the risk of errors, but does not eliminate it entirely !!
- Execution monitoring is required to detect errors and (hopefully) fix them [closed-loop execution]
  - What to monitor?
  - How to fix errors?

39

What to monitor?

- Action monitoring:
  - Check preconditions before executing an action and effects after
  - Not very efficient (e.g., a precondition may have been false for a while)
- Plan monitoring:
  - Check the preconditions of the entire remaining plans
  - → Triangle tables

40

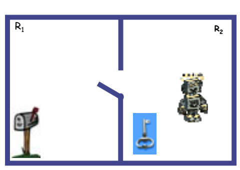
Key-in-Box Problem

Grasp-Key-in-R_2
- P = In(Robot,R_2) $\wedge$ In(Key,R_2)
- D = $\emptyset$
- A = Holding(Key)

Lock-Door
- P = Holding(Key)
- D = Unlocked(Door)
- A = Locked(Door)

Move-Key-from-R_2-into-R_1
- P = In(Robot,R_2) $\wedge$ Holding(Key) $\wedge$ Unlocked(Door)
- D = In(Robot,R_2), In(Key,R_2)
- A = In(Robot,R_1), In(Key,R_1)

Put-Key-Into-Box
- P = In(Robot,R_1) $\wedge$ Holding(Key)
- D = Holding(Key), In(Key,R_1)
- A = In(Key,Box)



41

Triangle Table

Plan:
Grasp-Key-in-R_2, Move-Key-from-R_2-into-R_1, Lock-Door, Put-Key-Into-Box
to achieve Locked(Door) $\wedge$ In(Key, Box)

In(Robot,R_2) In(Key,R_2)	Grasp-Key-in-R_2			
In(Robot,R_2) Unlocked(Door)	Holding(Key)	Move-Key-from-R_2-into-R_1		
	Holding(Key)		Lock-Door	
	Holding(Key)	In(Robot,R_1)		Put-Key-Into-Box
			Locked(Door)	In(Key,Box)

42

Triangle Table

Propositions from the initial state that are necessary to the applicability of actions in the plan and the achievement of the goal

In(Robot,R_2) In(Key,R_2)	Grasp-Key-in-R_2				
In(Robot,R_2) Unlocked(Door)	Holding(Key)	Move-Key-from-R_2-into-R_1			
	Holding(Key)		Lock-Door		
	Holding(Key)	In(Robot,R_1)		Put-Key-Into-Box	
			Locked(Door)	In(Key,Box)	

43

Triangle Table

Propositions achieved by Grasp-Key-in-R_2 that are necessary to the applicability of subsequent actions and the achievement of the goal

In(Robot,R_2) In(Key,R_2)	Grasp-Key-in-R_2				
In(Robot,R_2) Unlocked(Door)	Holding(Key)	Move-Key-from-R_2-into-R_1			
	Holding(Key)		Lock-Door		
	Holding(Key)	In(Robot,R_1)		Put-Key-Into-Box	
			Locked(Door)	In(Key,Box)	

44

Triangle Table

Propositions contained in initial state or achieved by actions of the plan that are necessary to the achievement of the goal

In(Robot,R_2) In(Key,R_2)	Grasp-Key-in-R_2				
In(Robot,R_2) Unlocked(Door)	Holding(Key)	Move-Key-from-R_2-into-R_1			
	Holding(Key)		Lock-Door		
	Holding(Key)	In(Robot,R_1)		Put-Key-Into-Box	
			Locked(Door)	In(Key,Box)	

45

Triangle Table

3rd kernel of the plan:
If all the propositions in it are true, the agent can proceed to execute the 3rd action of the plan (Lock-Door)

In(Robot,R_2) In(Key,R_2)	Grasp-Key-in-R_2				
In(Robot,R_2) Unlocked(Door)	Holding(Key)	Move-Key-from-R_2-into-R_1			
	Holding(Key)		Lock-Door		
	Holding(Key)	In(Robot,R_1)		Put-Key-Into-Box	
			Locked(Door)	In(Key,Box)	

46

Triangle Table

5th kernel of the plan:
If all the propositions in it are true, the goal is achieved

In(Robot,R_2) In(Key,R_2)	Grasp-Key-in-R_2				
In(Robot,R_2) Unlocked(Door)	Holding(Key)	Move-Key-from-R_2-into-R_1			
	Holding(Key)		Lock-Door		
	Holding(Key)	In(Robot,R_1)		Put-Key-Into-Box	
			Locked(Door)	In(Key,Box)	

47

Execution Monitoring with Triangle Tables

Repeat:

1. Observe the world and identify the largest k such that all the propositions in the kth kernel are true
2. If k = 0 then re-plan
3. Else execute the kth action of the plan

→ Actions that fail are repeated
→ Actions that are not needed are skipped

48

But ...

- Repeating an action that failed assumes that it may succeed next time. But what if the agent picked the wrong key in R_2?
- Either the agent has more knowledge or sensors than it used so far, and it's time to use them
- Or it doesn't have any of these, and it has no choice – fail or call another agent
  [I do the same when my car does not start and I can't figure out why]

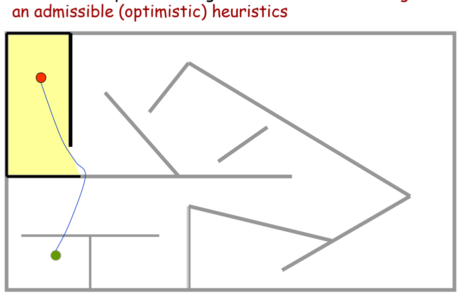
49

On-Line Search

- Sometimes uncertainty is so large that actions need to be executed for the agent to know their effects
- Example: A robot must reach a goal position. It has no prior map of the obstacles, but its vision system can detect all the obstacles visible from a the robot's current position

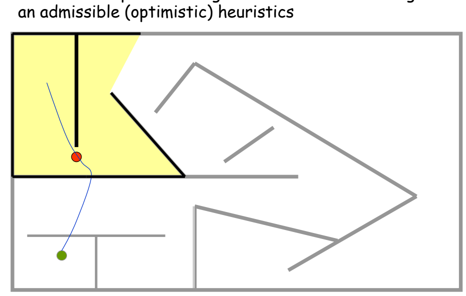
50

Assuming no obstacles in the unknown region and taking the shortest path to the goal is similar to searching with an admissible (optimistic) heuristics



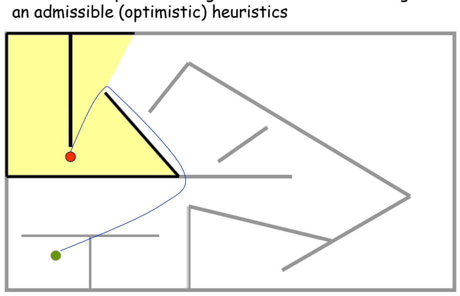
51

Assuming no obstacles in the unknown region and taking the shortest path to the goal is similar to searching with an admissible (optimistic) heuristics



52

Assuming no obstacles in the unknown region and taking the shortest path to the goal is similar to searching with an admissible (optimistic) heuristics



Just as with classical search, on-line search may detect dead-ends and move to a more promising position (~ node of search tree)

53

Suggestion

- It's time to refresh your memory on probability theory:
  - axioms of probability
  - random variable
  - joint distributions
  - conditioning
  - independence

  [R&N: Chap. 13, Sect. 13.3-6]
- We will be using probabilities in a few lectures

54