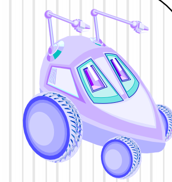


LECTURE 2 - 1

Representation and The Reactive Paradigm

Introduction to AI Robotics (Ch. 3)

Quote of the Week

"In the fifties, it was predicted that in 5 years robots would be everywhere.
In the sixties, it was predicted that in 10 years robots would be everywhere.
In the seventies, it was predicted that in 20 years robots would be everywhere.
In the eighties, it was predicted that in 40 years robots would be everywhere..."

Marvin Minsky

ANNOUNCEMENTS

- Lab 2 - Random Wander, Obstacle Avoidance demonstration is due on *Thursday, 3/18/10*
- The lab memo and code is due on Angel by midnight on *Thursday, 3/18/10*
- Quiz 3 on Lecture 2 – 1, Ch. 3 on *Tuesday, 3/16/10*

OBJECTIVES

Upon completion of this lecture the student should be able to:

- Describe representation and world model
- List the different types of world maps
- Describe an animal behavior
- Describe reflexive behaviors
- Define innate releasing mechanisms
- Describe the role of perception in behaviors
- Apply schema theory to implement robot behaviors

Representation

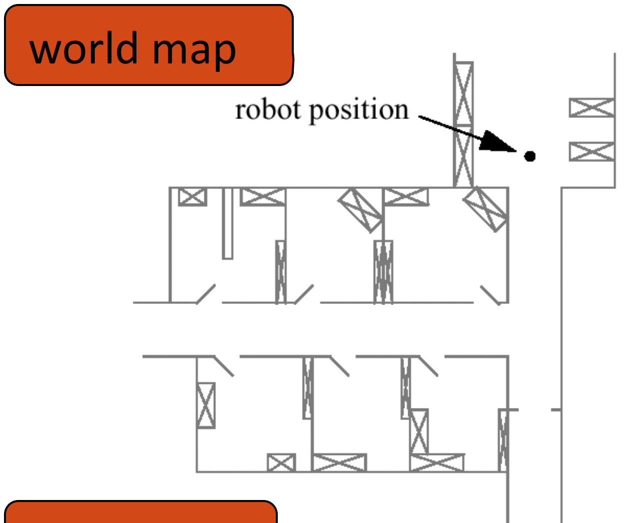
REPRESENTATION

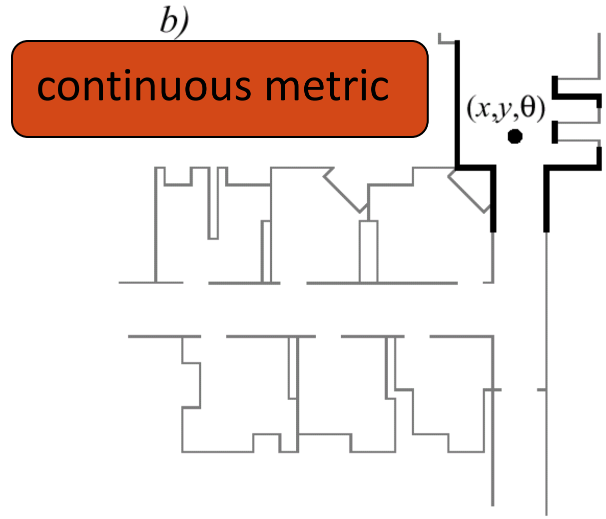
- What does the robot know and keep in its brain?
- ***Representation*** is the form in which information is stored or encoded in the robot
- The robot may need to remember what happened in the past or predict what will happen in the future
- The robot many need to store maps of the environment, images of people or places

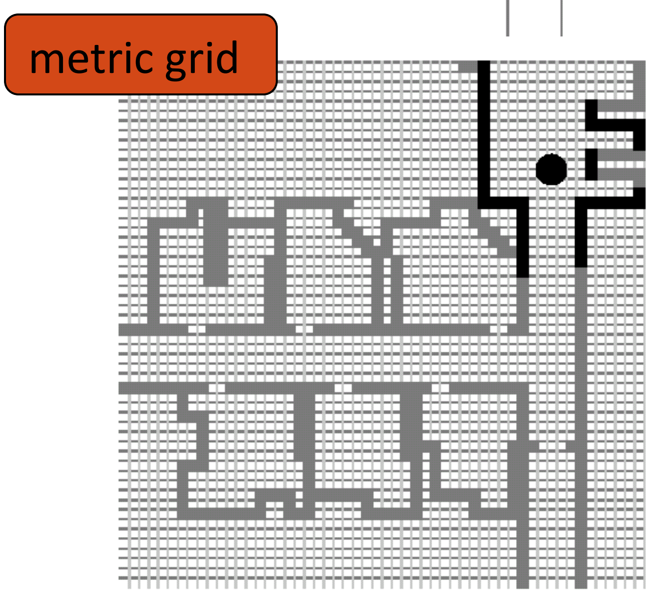
WORLD MODEL

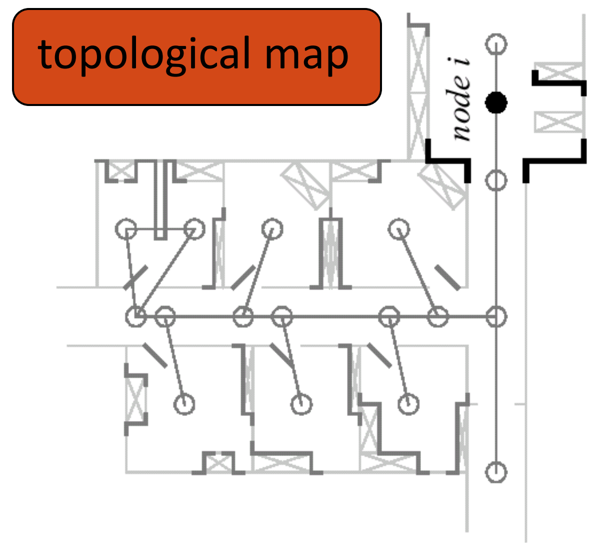
- Representation of the world is typically called a ***world model***
- The map is the most commonly used model of a world model
- The robot may use an ***odometric path*** to recall the route traveled
- The robot may use a ***landmark based path*** using salient features of the environment
- The robot may use a ***landmark based map*** which tells the robot what to do at each landmark regardless of order. A collection of landmarks with links is called a ***topological map***.
- The robot may recall a maze by drawing it by using exact lengths of corridors and distances to walls. This is a ***metric map***.

ENVIRONMENT REPRESENTATION



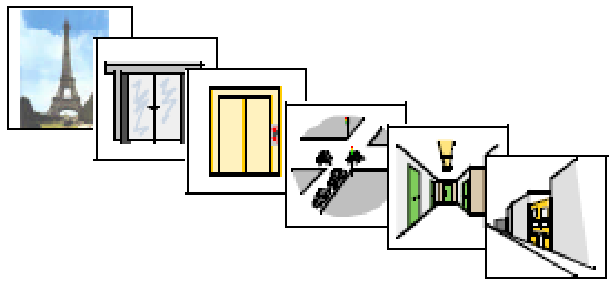




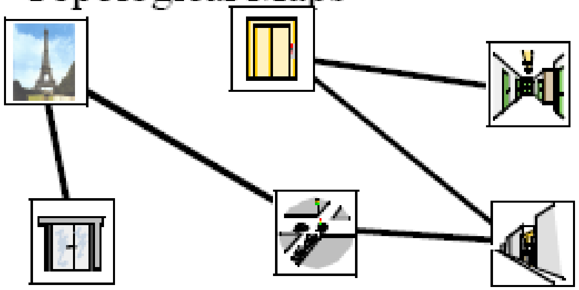


MAP CATEGORIES

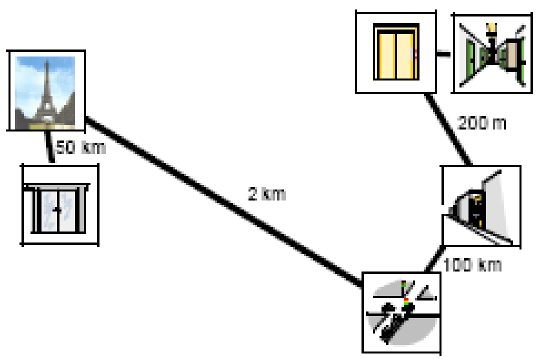
- Recognizable Locations



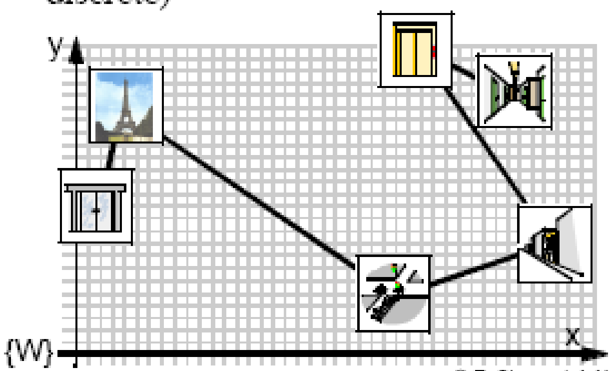
- Topological Maps



- Metric Topological Maps



- Fully Metric Maps (continuos or discrete)



WORLD MODEL COMPARISONS

- The ***odometric map*** is only useful if the world does not change and the robot is able to accurately keep track of distances and turns
- The ***landmark-based map*** does not require the robot to be accurate but the world still cannot change
- The ***fully metric map*** is the most complicated and most useful because the robot has to take many measurements and store more information

OTHER REPRESENTATIONS

- Self
  - proprioception, self-limitations, goals, sensors, intentions, plans
- Environment
  - Navigable spaces, structures, maps
- Objects
  - People, doors, other robots, detectable things in the world
- Actions
  - Outcomes of specific actions in the environment
- Tasks
  - What needs to be done, where, in what order ,how fast

REPRESENTATION TIMELINE

- Keeping a model updated takes sensing, computation and memory
- Some models take a long time to construct and may be kept around for the lifetime of the robot’s task (i.e. detailed metric maps)
- Some models may be quickly constructed and soon discarded (ie. odometric path)

REPRESENTATION AND CONTROL ARCHITECTURES

- Different architectures have ***centralized world models*** or ***distributed world models***
- ***Deliberative Control*** must have an accurate world model
- ***Reactive control*** does not facilitate the use of models
- ***Hybrid control*** uses multiples types of models
- ***Behavior-based control*** imposes constraints on the time and space models used
- Certain control architectures and representations are more appropriate for a given robot and task

Reactive Control

Introduction to AI Robotics (Ch. 3)

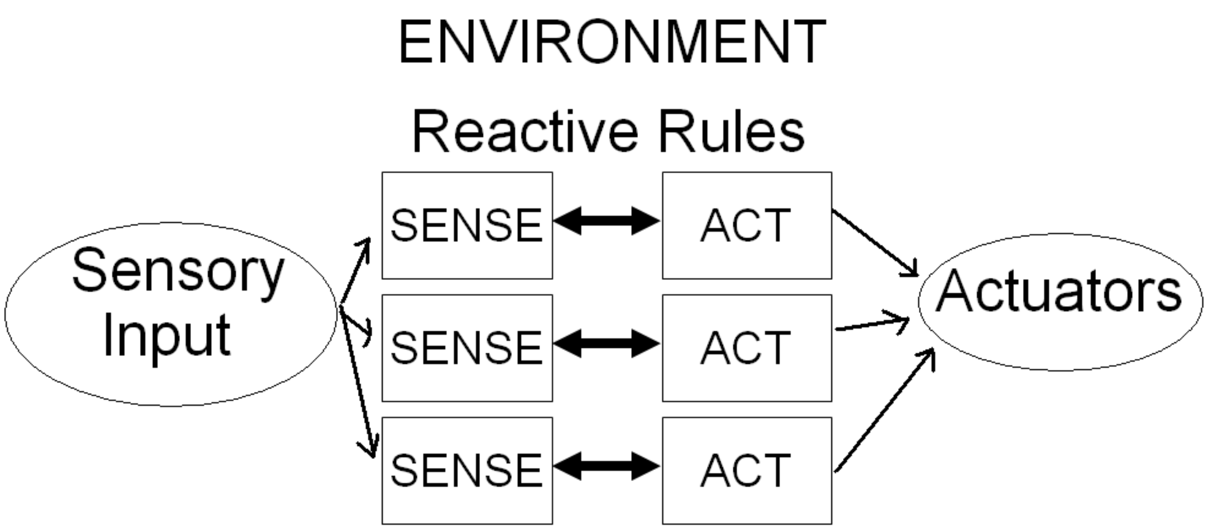
REACTIVE CONTROL

- Reactive Control is one of the mostly commonly used methods for robot control and is based on a tight connection between the robot’s sensors and effectors
- They do not use any internal representations and do not look ahead at the possible outcomes of their actions
- They operate on a short time scale and react to the current sensory information
- They have reactive rules (i.e. reflexes) to specific sensory input
- Complex computation is removed entirely in favor of fast, stored pre-computed responses

STIMULI AND BEHAVIORS

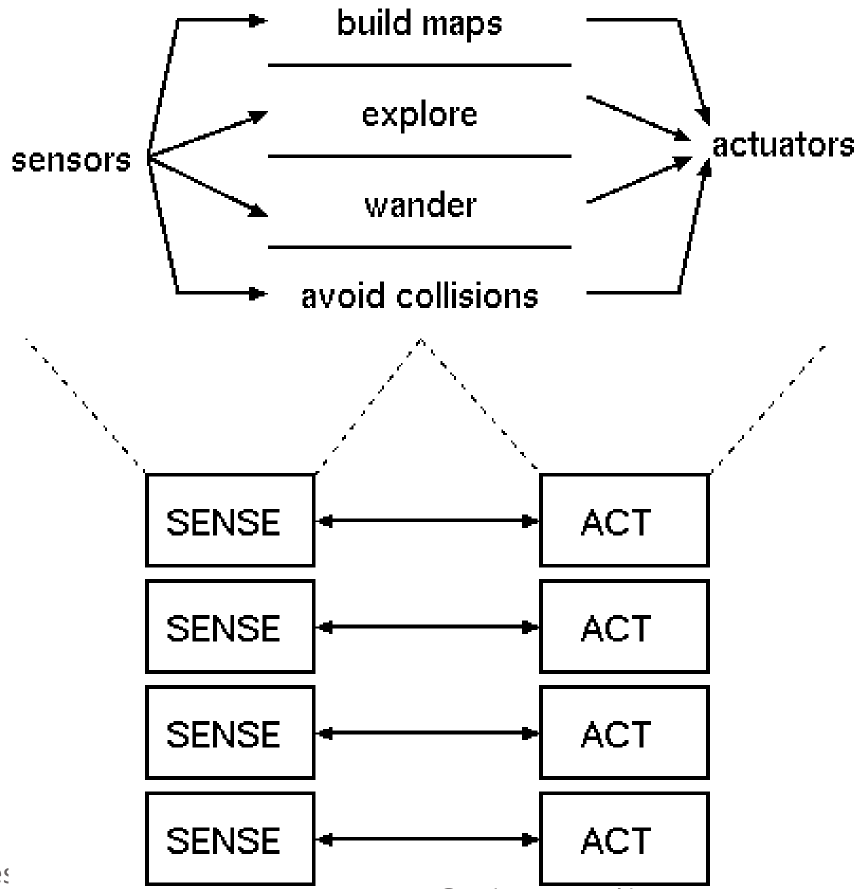
- The robot system has a set of situations (stimuli or coordinations) and a set of actions (responses, actions, behaviors)

- The situations may be based on sensory inputs or on internal state
- Examples are obstacle avoidance or random wander



VERTICAL DECOMPOSITION

Biological systems are more vertical



MUTUALLY EXCLUSIVE CONDITIONS

- To keep a reactive system simple have one unique behavior for each stimuli. The conditions are ***mutually exclusive*.**
- As the sensor state space grows the space may become unwieldy or intractable
- Coming up with the complete set of rules for the state space is typically done at design time not run time
- Typically there are only rules for important events and a default response for all others

STUCK SITUATIONS

- A reactive controller may be get stuck if there is a default rule that covers some of the states. In wall following this may be resolved by the following:
  - Introduce randomness to get the robot unstuck from a corner by having it turn by a random angle instead of a fixed one
  - Keep a history and remember the direction the robot turned last and use that information to make the decision about the turn direction

ACTION SELECTION

- ***Action Selection*** is the process of deciding among multiple possible behaviors when they are not mutually exclusive
- ***Command arbitration*** is the process of selecting one behavior from multiple candidates
- ***Command fusion*** is the process of combining multiple candidate behaviors into a single output behavior for the robot
- Reactive systems must support parallelism and the program must be able to multitask