Lecture 6-1

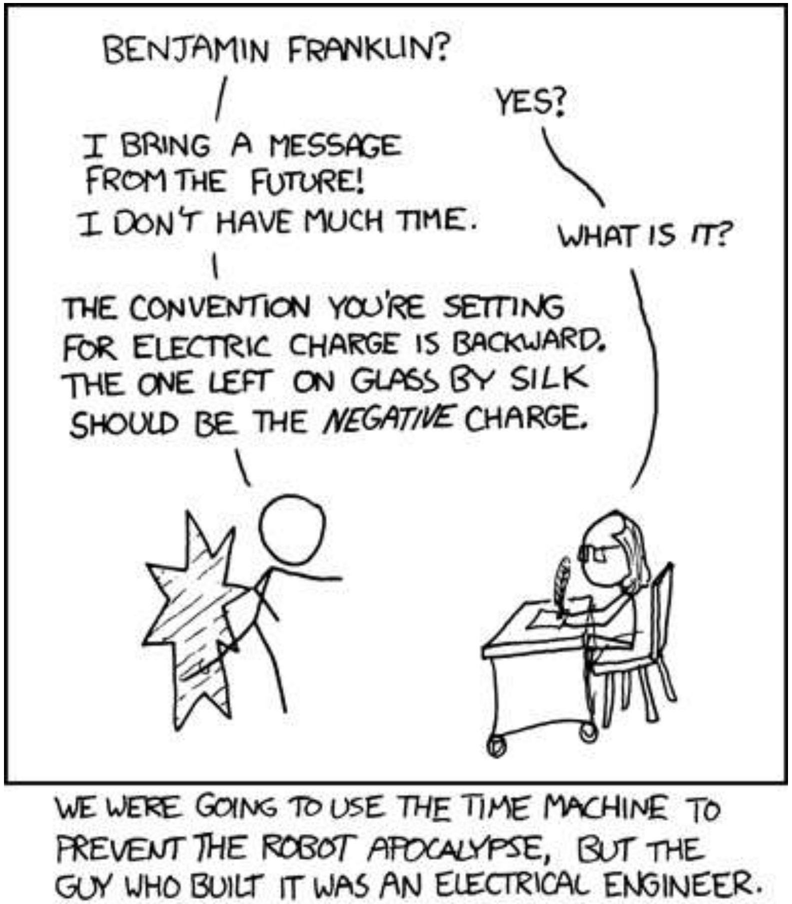
The Building Blocks of Control:
Control Architectures
What's in Your Head?
Representation

The Robotics Primer (Ch. 11, 12)

Course Announcements

- Quiz on Tuesday, 4/20/09 on **Control Architectures and Representation**
- Your robots will be returned on Tuesday
- Lab 4 Demo due Thursday, 4/23/09
- Lab 4 Memo and code due by midnight on Friday, 4/24/09
- Upload memo and code to Angel
- DO NOT unplug the network cables from the desktop computers or the walls

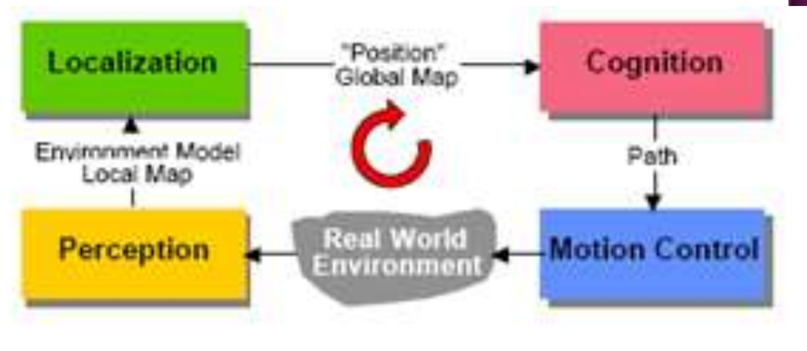
Cartoon of the Week

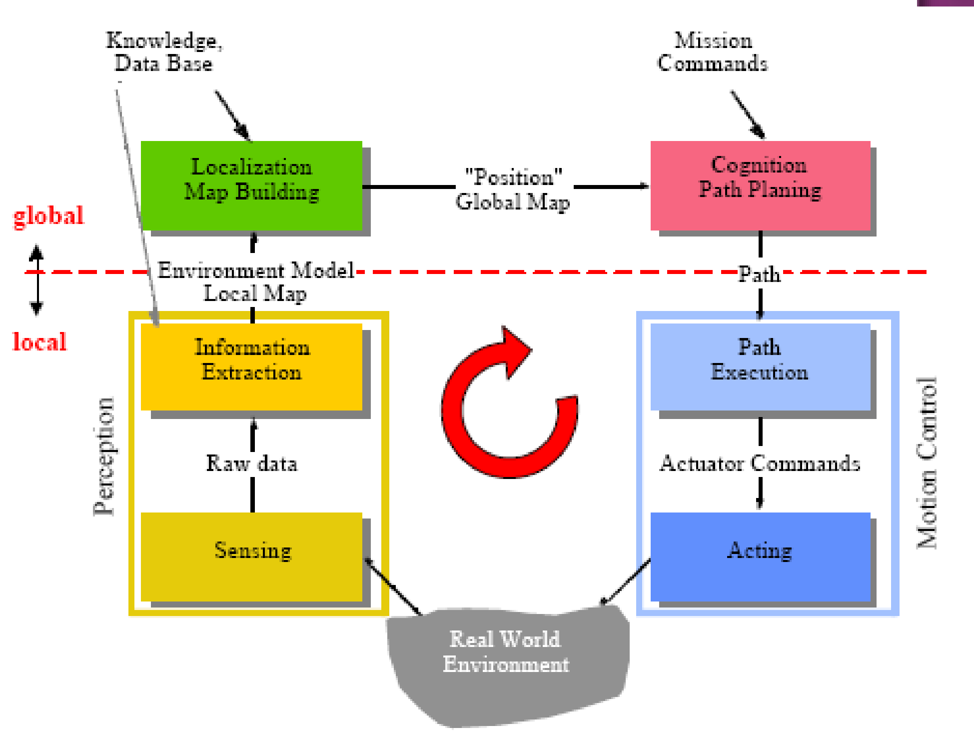


Control Architectures (Ch. 11)

Control of Mobile Robots

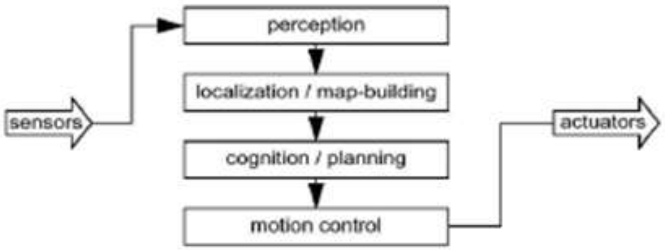
- Controls
  - Dynamically changing
  - No compact model available
  - Multiple sources of error
- Most functions are ‘local’ and do not involve localization or cognition
- Localization and global path planning are slower and should be performed only when needed
- This is similar to what human beings do

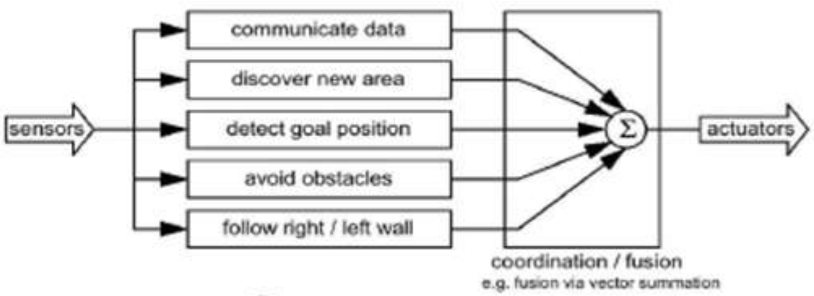


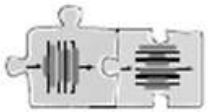


Control Approaches

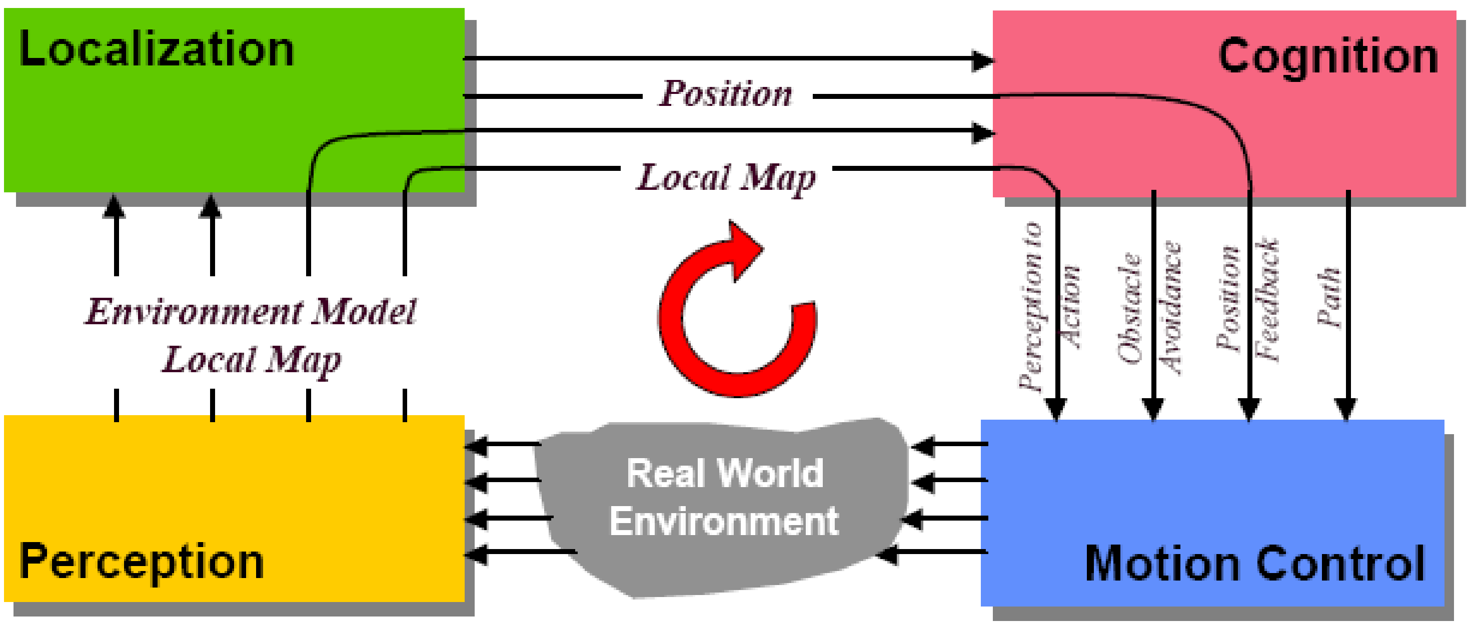
- Classical AI Control
  - Complete modeling
  - Function based
  - Horizontal decomposition
- New AI Control
  - Sparse or no modeling
  - Behavior-based
  - Vertical decomposition (bottom up)
- Possible solution
  - Combine approaches







Mixed Approach



Control Architectures

- The controller is the brains of the robot. Robot control is the means by which sensing and action of a robot are coordinated.
- Feedback control is good for low level control of actuators (wall following, obstacle avoidance)
- More complex tasks require a control architecture
  - This may involve using multiple controllers
  - This may involve arbitration of controllers for complex tasks
- A robot **control architecture** provides the guiding principles and constraints for organizing a robot's control system (i.e. brain)

Software versus Hardware

- Robot control can take place in hardware or software
- More complex controllers are typically implemented in software
- Hardware is good for fast and specialized uses
- Software is good for flexible, more general programs
- Brains use programs to solve problems and to achieve goals
- Solving a problem using a finite step by step procedure is called an algorithm

Four Types of Control Architectures

- Deliberative Control
- Reactive Control
- Hybrid Control
- Behavior-based Control
- These four architectures differ in the way they handle
  - Time
  - Modularity
  - Representation

Robot Paradigms and Primitives

- A *paradigm* is a philosophy or set of assumptions and/or techniques which characterize an approach to a class or problems
- The three commonly accepted primitives of robotics are: *sense, plan, act*

Robot Primitives	Input	Output
SENSE	Sensor data	Sensed information
PLAN	Information (sensed and/or cognitive)	Directives
ACT	Sensed information or directives	Actuator commands

Time

- How fast do things happen?
- Do all components of the controller run at the same speed?
- Time refers to how fast the robot responds to the environment compared with how quickly it can sense and think
- *Deliberative control* looks into the future so it works on a long time-scale
- *Reactive control* responds to the immediate, real- time demands of the environment without looking into the past or the future
- *Hybrid control* combines the long time scale of deliberative control and the short time scale of reactive control
- *Behavior-based control* works to bring the time-scales together in a different way than hybrid control

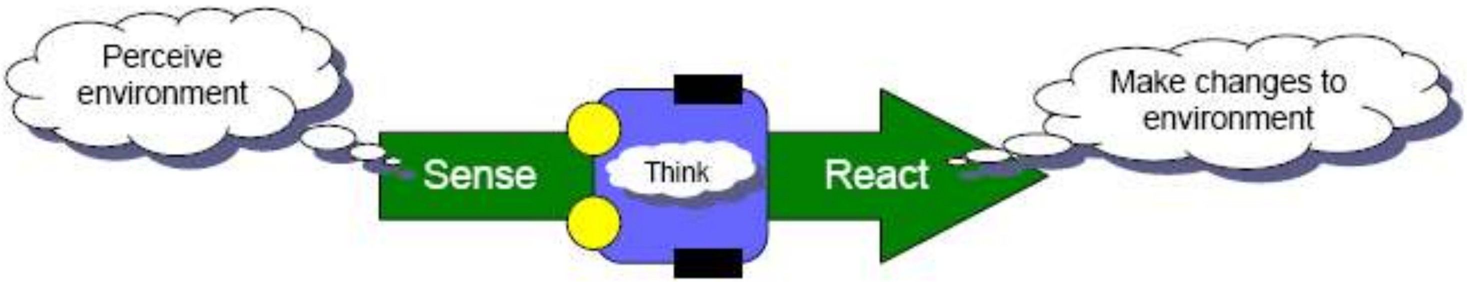
Modularity

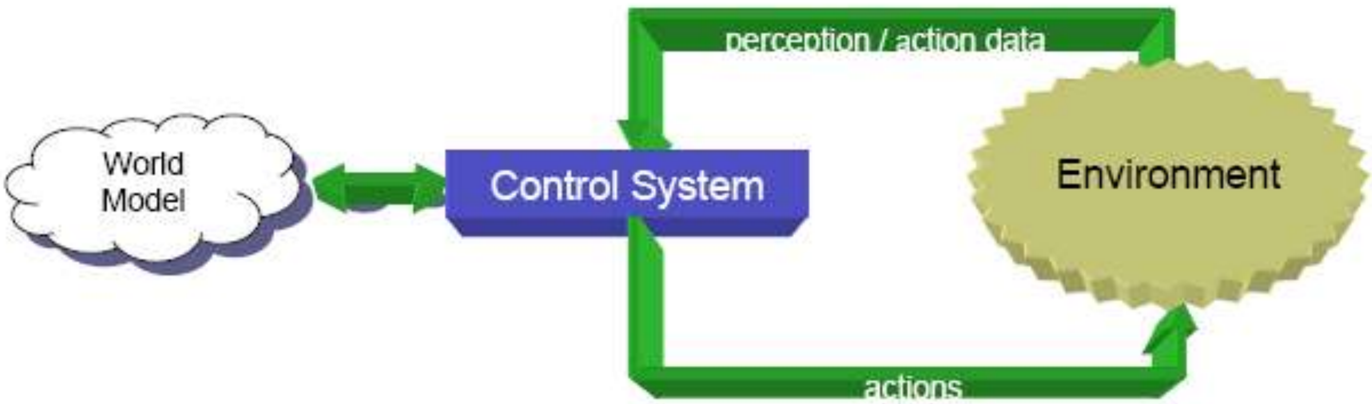
- What are the components of the control system?
- What can talk to what?
- *Modularity* refers to the way the control system is broken into pieces, components, or modules.
- *Modularity* also refers to how the modules interact with each other to product the robot's overall behavior
- *Deliberative control* has a control system with sensing, planning and acting modules that work in sequence
- In *reactive control*, multiple modules are all active in parallel and can send messages to each other in various ways
- In *hybrid control*, the three main modules are the deliberative, reactive and the connection in between
- In *behavior-based control*, there are more than 3 modules and they work in parallel and talk to each other

Deliberative vs. Reactive Control

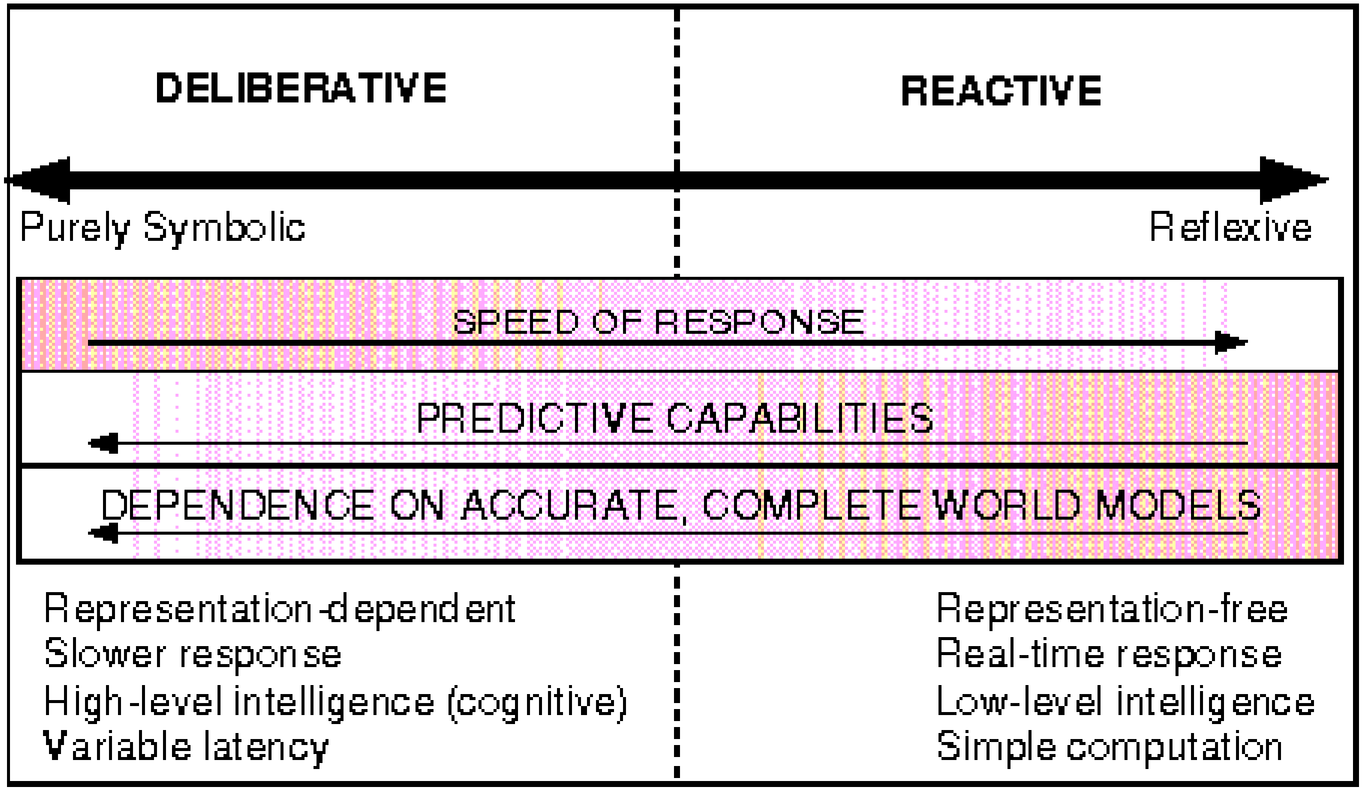
- Deliberative control *(Think hard then act)*
  - *Sense $\rightarrow$ Plan $\rightarrow$ Act*
  - Involves planning to avoid bad solutions
  - Flexible for increasing complexity
  - It is slow and speed decreases with complexity
  - Requires large amounts of accurate information
- Reactive Control *(Don't think, react)*
  - *Sense $\rightarrow$ Act*
  - Fast, regardless of complexity
  - Built-in or learned from looking in the past
  - Limited flexibility for increased complexity

Reactive vs. Deliberative Control





Spectrum of Robot Control



From "Behavior-Based Robotics" by R. Arkin, MIT Press, 1998

Hybrid Control

Think and act independently and concurrently

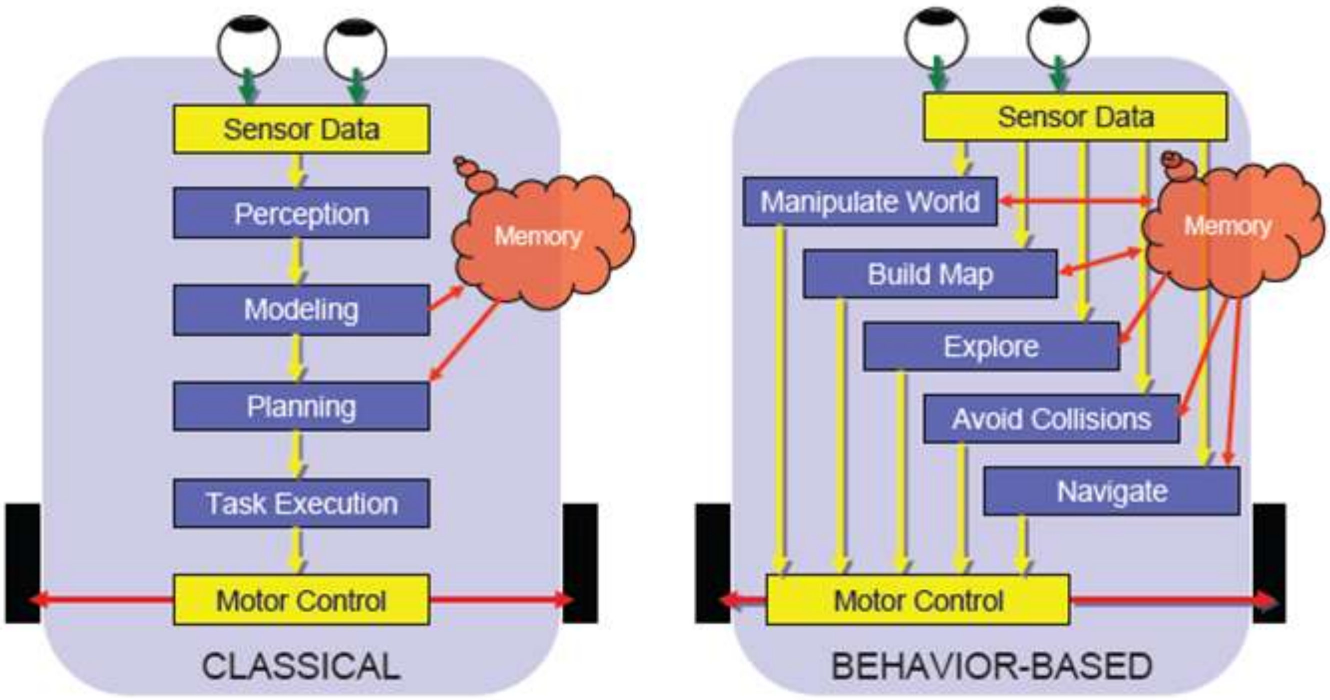
- *Plan, Sense → Act*
- Combination of reactive and deliberative control
- Reactive layer is the bottom level and deals with immediate reaction
- Deliberative layer is the top level and creates plans
- The middle layer connects the two layers
- Typically called the “three-layer system”
- Reactive and deliberative layers have different time-scales and representations (signals, symbols)
- Hybrid control is one of the two dominant control paradigms in robotics

Behavior-Based Control (BBC)

Think the way you act

- *Sense* $\rightarrow$ *Act*
- Inspired by biology and has the same capabilities as hybrid control
- BBC acts reactively and deliberately
- BBC is built from layers but no intermediate layer
- BBC has a uniform representation and time-scale
- **Behaviors** are concurrent processes that take inputs from sensors and other behaviors and send outputs to the robot's actuators or other behaviors to achieve some goals
- Thinking performed through a network of behaviors and uses distributed representations
- Responds in real-time (reactive) and allows for a variety of behavior coordination mechanisms

Classical vs. Behavior-Based Control



Representation (Ch. 12)

Representation

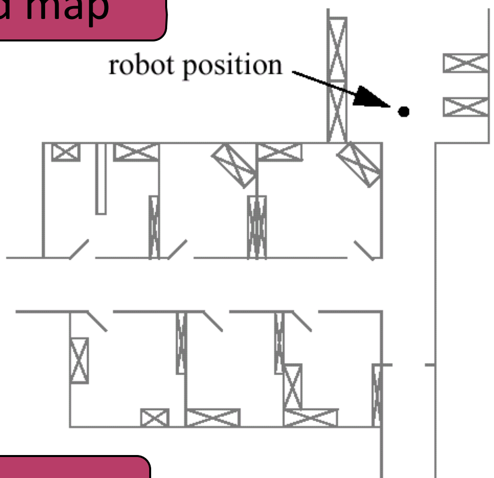
- What does the robot know and keep in its brain?
- *Representation* is the form in which information is stored or encoded in the robot
- The robot may need to remember what happened in the past or predict what will happen in the future
- The robot many need to store maps of the environment, images of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

World Model

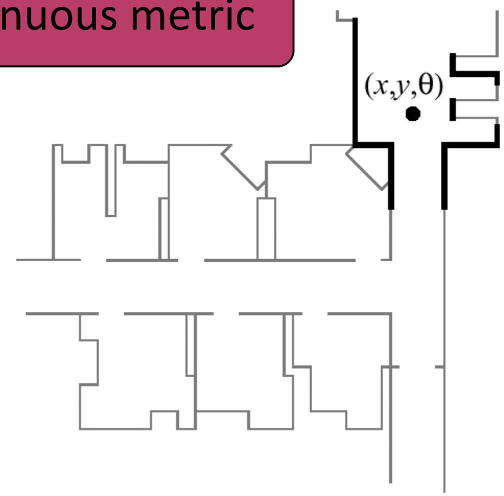
- Representation of the world is typically called a *world model*
- The map is the most commonly used model of a world model
- The robot may use an odometric path to recall the route traveled
- The robot may use a *landmark based path* using salient features of the environment
- The robot may use a *landmark 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

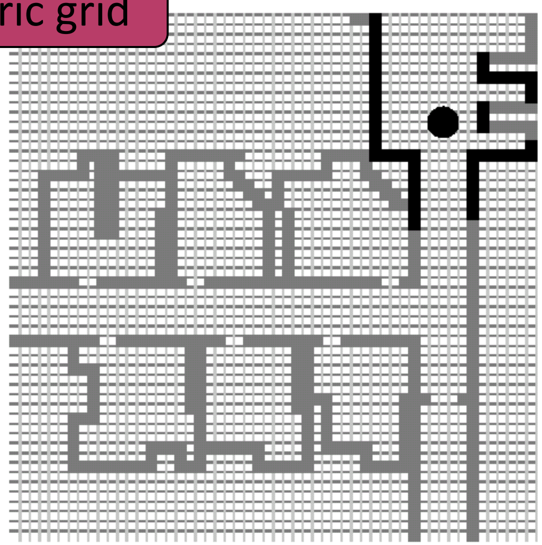
world map



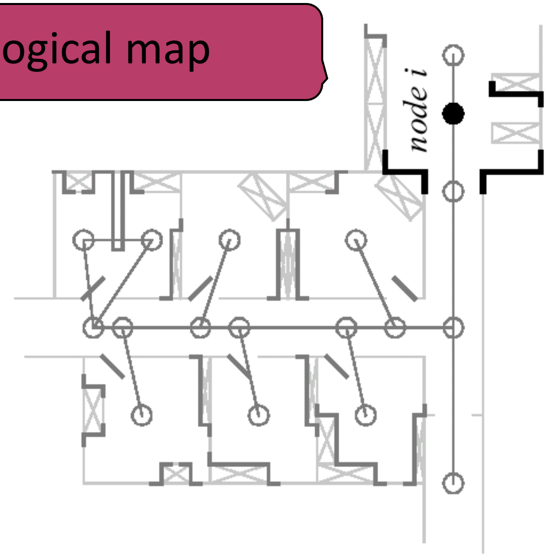
continuous metric



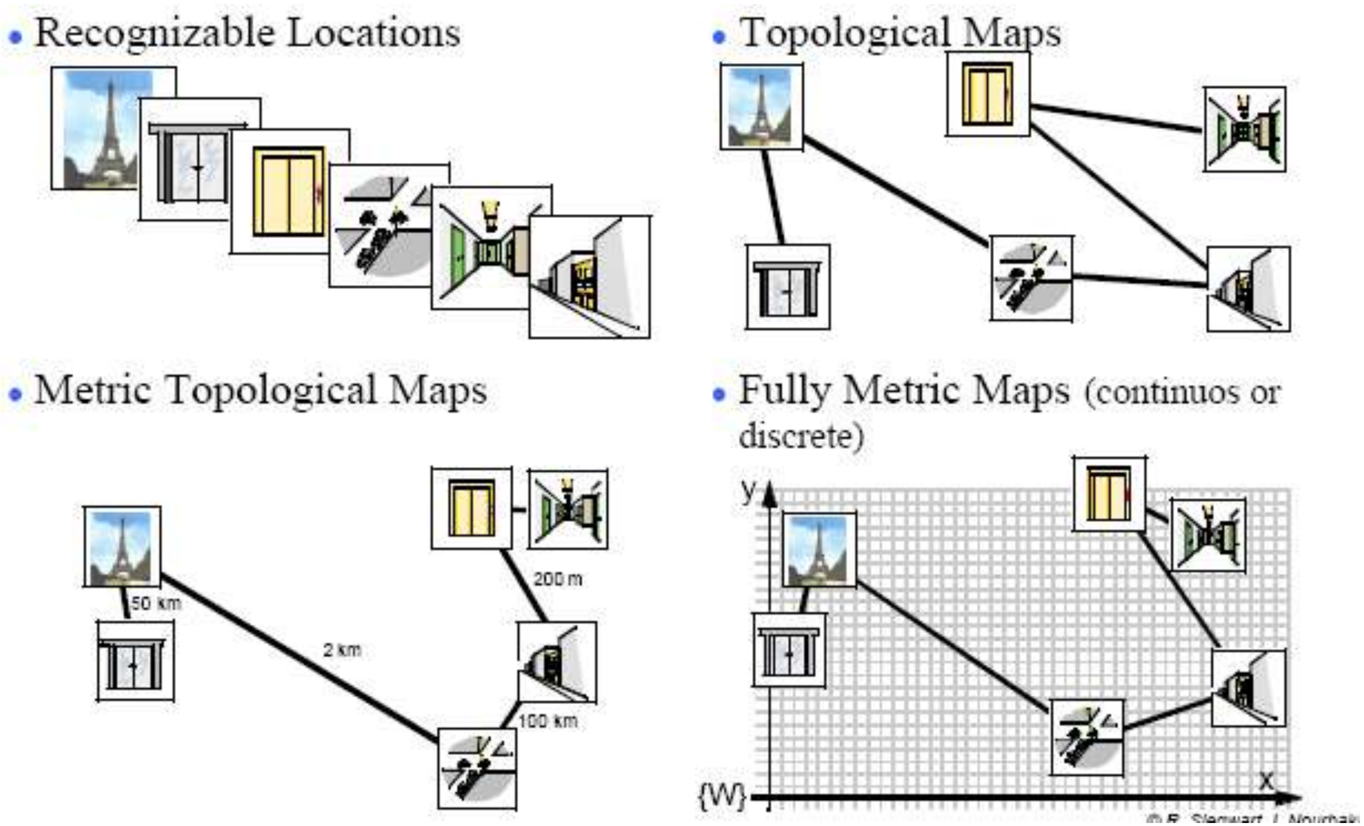
metric grid



topological map



Map Categories



World Model Comparisons

- The *odometric map* is only useful if the world does not change and the and the robot is able to accurately keep track of distances and turns
- The *landmark-based path* does not require the robot to be accurate but the world still cannot change
- The *landmark-based map* also does not require the robot to be accurate and the world cannot change
- The *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