Reasoning robot, involved tasks/modules and robot world representations

Václav Hlaváč

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Center for Machine Perception
`http://cmp.felk.cvut.cz/~hlavac, hlavac@fel.cvut.cz`

Outline of the talk:

- Artificial intelligence view.
- Symbol grounding.
- Applied AI approaches.
- Need for a robot control architecture.
- Path planning / trajectory generation.
- Robot world models.

Robot reasoning/controlling at large

- We have studied a single feedback control loop.
- How do we put together multiple feedback control loops?
  - At which level of generality?
  - In what order?
  - In what priority?
- How do we generate reliable and correct behavior of a robot?

What is intelligence?

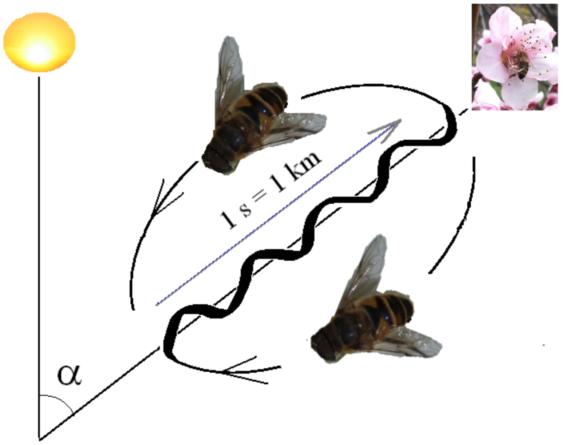
- How do we decide if a particular system or device (e.g., hardware and software combination) is intelligent?
- The Turing test (1950).
- ALVINN – Autonomous Land Vehicle in a Neural Network. Carnegie-Mellon Univ. 1993; Learning by observation of a human driver.
  - Inputs: stereo video, human control, laser range finders, radar, inertial navigation.
  - Outputs: steering, acceleration, braking.
  - Processing: neural network, 30×32 unit video input; 30 unit output (steering control), 5 hidden units, winner takes all output.

What behaviors are intelligent?

- Memory.
- Learning.
- Language.
- Planning.
- Eye gaze following.
- Predicting how someone will act based on her/his beliefs or knowledge.

A biological motivation

- Use animal biology as a metaphor for intelligence.
- Bees, rats, dolphins, slugs, chimpanzees, humans all exhibit forms of intelligence.
- A bee performs a waggle dance. (in Czech: osmičkový tanec)
- Dolphins and bats use echolocation (sonar).
- Chimpanzees can track another's gaze.
- Rats are excellent at remembering locations.



Intelligence – an evolutionary metaphor

- We consider a bee, a snake, a rat, to have forms of intelligence because they exhibit behaviors (including learning behaviors) adapted to particular environments.
- We can also consider various artificial systems to have forms of intelligence, e.g. ALVINN.

Intelligence

There is wide agreement among AI researcher that intelligence is required to do the following:

- Reason, use strategy, solve puzzles, and make judgments under uncertainty.
- Represent knowledge, including commonsense knowledge.
- Plan.
- Learn.
- Communicate in natural language.
- Integrate all these skills towards common goals.
- Ability to sense (e.g. see) and act (e.g. move and manipulate objects).
- Salience (in Czech významnost) – the capacity to recognize importance.

Interdisciplinary approaches to intelligence (cognitive science, computational intelligence) emphasise autonomy and imagination (ability to form concepts that were not programmed in).

Ethics of AI

Dealt with in science fiction

- Consciousness – to have subjective experience and thought.
- Self-awareness – to be aware of oneself as a separate individual, especially to be aware of one's own thoughts.
- Sentience – ability to feel perceptions or emotions subjectively
- Sapience – the capacity for wisdom.

A Czech language corner

- Consciousness – *vědomí*.
- Self-awareness – *uvědomění si sama sebe*.
- Sentience – *uvědomování si*.
- Sapience – *prozíravost, často užíváno ironicky*.

Some philosophy

Rationalism

- “any view appealing to reason as a source of knowledge or justification”, “ the criterion of the truth is not sensory but intellectual and deductive”.
- Intelligence stems from (logical) reasoning.
- Many rationalists believe that some part of human knowledge is innate.
- *E.g., René Descartes (1596–1650), Baruch Spinoza (1632–1677), Gottfried Leibniz (1646–1716).*

Empiricism

- The idea that intelligence stems from the senses “An empiricist holds that experience is the source of all human knowledge”
- E.g., all knowledge must be grounded or based in the sensory world.
- *E.g. Aristotle (384 BC–322 BC), William of Ockham (1288–1348), Francis Bacon (1561-1626), John Locke (1632-1704), George Berkeley (1685-1753), David Hume (1711-1776), Hermann von Helmholtz (1821-1894), and Karl Popper (1902-1994).*

Classical AI

- Formal mathematical logic approach (“Laws of thought approach”).
- Making correct inferences.
- Program input:

  Description of the problem in a formal, logical notation.
- Reasoning:

  Logical inference. Find the solution to the problem, if the solution exists.

A physical symbol system hypothesis

- A set of arbitrary “physical tokens”, e.g. pen strokes on a paper, holes on a tape, events in a digital computer, that are manipulated on the basis of “explicit rules” that are likewise physical tokens and strings of tokens.
- The rule-governed symbol-token manipulation is based purely on the shape of the symbol tokens (not their “meaning”), i.e., it is purely syntactic, and consists of “rulefully combining” and recombining symbol tokens.
- There are primitive atomic symbol tokens and composite symbol-token strings.
- The entire system and all its parts – the atomic tokens, the composite tokens, the syntactic manipulations both actual and possible and the rules – are all “semantically interpretable”.
- The syntax can be systematically assigned a meaning e.g., as standing for objects, as describing states of affairs.

Harnad, S. (1990) The Symbol Grounding Problem. Physica D 42: 335-346.

Symbol grounding

- Computationalism (one cognition theories) states: cognition (i.e., thinking) is just a form of computation.
- Symbols are manipulated according to rules that are based on the symbols' shapes, not their meanings.
- How are those symbols (e.g., the words in our heads) connected to the things they refer to?
- The symbols in an autonomous hybrid symbolic+sensorimotor system would be grounded. (Importance of the perception action cycle.)
- But whether its symbols would have meaning rather than just grounding is something cognitive science itself—cannot determine, or explain.

Weak and strong Artificial Intelligence

1. Weak Artificial Intelligence (also called Applied AI)
   - The goal is to achieve practical outcomes:
   - Examples: ALVINN, Deep blue – a chess program.
2. Strong Artificial Intelligence
   - The goal is to make machines truly intelligent (to match or exceed human intelligence), to have minds and be conscious.
   - Example: whole brain simulation (called also mind uploading).

A distinction between two positions in the philosophy of AI given by John Searle as part of his Chinese room argument.

Chinese room argument, J. Searle 1980

- There is a program that gives a computer the ability to carry on an intelligent conversation in written Chinese.
- If the program is given to someone who speaks only English to execute the instructions of the program by hand, then in theory, the English speaker would also be able to carry on a conversation in written Chinese.
- However, the English speaker would not be able to understand the conversation because she/he does not know how to interpret symbols.
- J. Searle concludes, a computer executing the program would not understand the conversation either.

The argument is directed against the philosophical positions of functionalism and computationalism, which hold that the mind may be viewed as an information processing system operating on formal symbols. The argument applies only to digital computers and does not apply to machines in general.

Practical AI approaches

1. Engineering approach
   - Building intelligent entities.
   - The most important goal is often behavior and not mechanism.
2. Cognitive modeling approach
   - Tries to achieve “psychological reality”, e.g. use studies from psychology on how humans think.
   - Trace of programs reasoning steps should follow human steps in reasoning. E.g. General Problem Solver (A. Newell & H.A. Simon, 1959), the first computer program which separated its knowledge of problems (rules represented as input data) from its strategy of how to solve problems (a generic solver engine, mathematical logic).

AI issues in robotics

- A state space – a set of possible configurations of a robot, usually a discrete one, e.g. a position on a occupancy grid.
- Representation of a robot environment (a practical implementation of the state space), e.g. the robot world model.
- A reasoning engine – usually some type of symbol manipulation mechanism used for search/planning in a state space.
- Ontology *(next slide)*.

Ontology

- In philosophy:

  Def: Ontology is the philosophical study of the nature of being, existence, or reality, as well as the basic categories of being and their relations.

- In computer and information science:

  Def: Ontology formally represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts.

  - It can be used to model a domain and support reasoning about entities.
  - Ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it.

Need for a robot control architecture

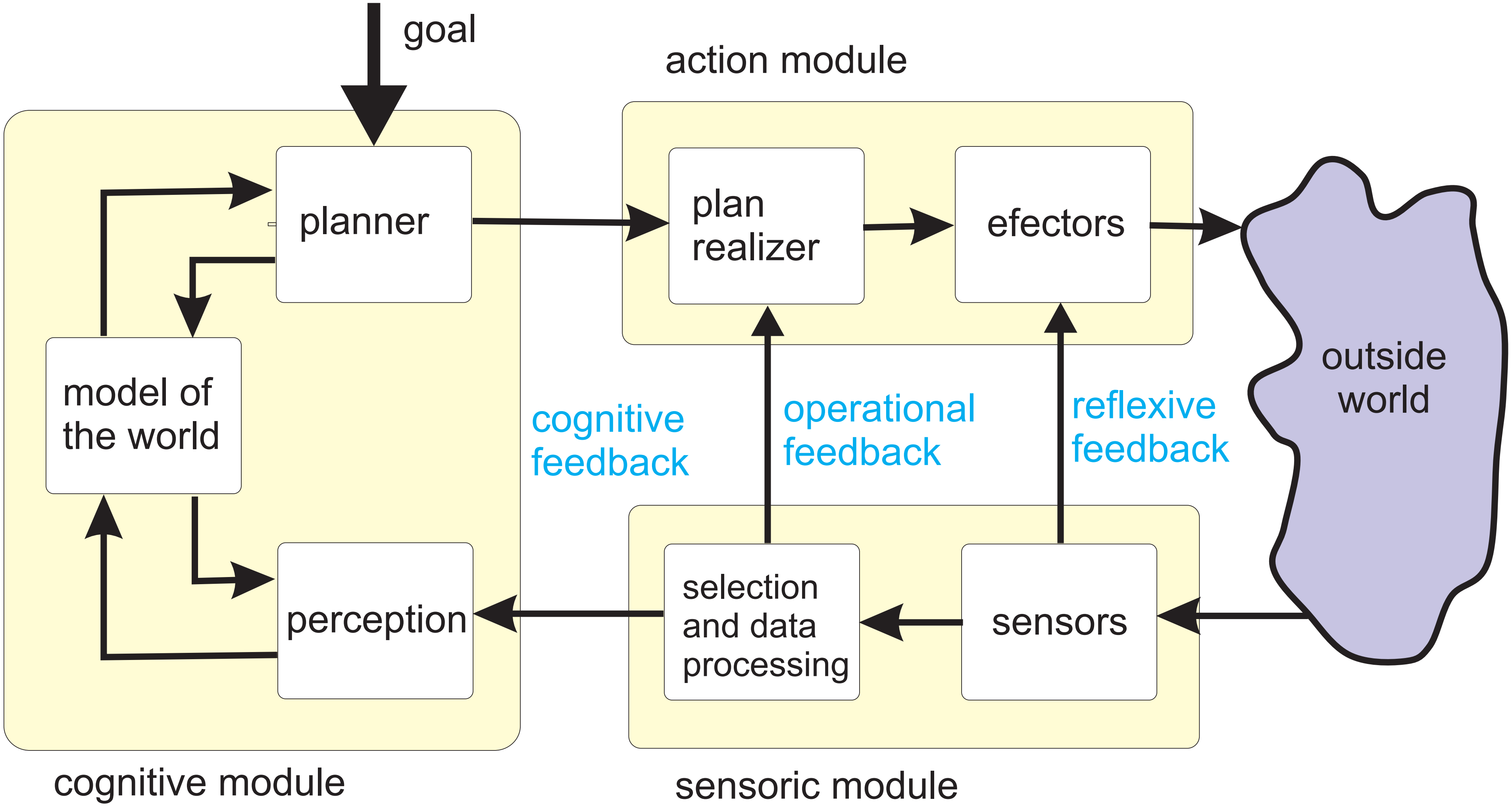
- A robot control architecture provides principles for organizing robot reasoning and robot world representation, sensing and perception, and actuation modules (both hardware and software), which are called jointly “a robot control architecture”.
- The control architecture specifies building blocks and their interrelations.
- The control architecture provides:
  - Structure.
  - Constraints.
- There have been four control architectures mainly used:
  1. Deliberative.
  2. Reactive.
  3. Behavior-based.
  4. Hybrid.

Brief comparison of four control architectures

Architecture	*Properties*
Deliberative	Lots of states. Maps of the robot environment. Look ahead.
Reactive	No states. No maps. No look ahead.
Behavior-based	Some states. Look ahead only while acting. Reactive + state.
Hybrid	States. Look ahead but act. Combines long and short time scales.

Motivating example, modules: A cognitive robot with a deliberative architecture

The deliberative architecture, the most sophisticated and oldest one from the artificial intelligence point of view, is used here to motivate main involved building blocks.



This lecture introduces two issues only

Only two blocks will be introduced in this lecture:

- Motion planner (only briefly, to motivate the second block).

  The higher level task planning, needed in some more complicated scenarios, is not discussed here for a moment.

  E.g. a mobile robot, which should play chess and be one of the players physically acting on a chessboard.

  Only motion of a robot would be of concern in this lecture.

- Model of the robot world (called often a map in mobile robotics).

Planning for manipulators and mobile robots

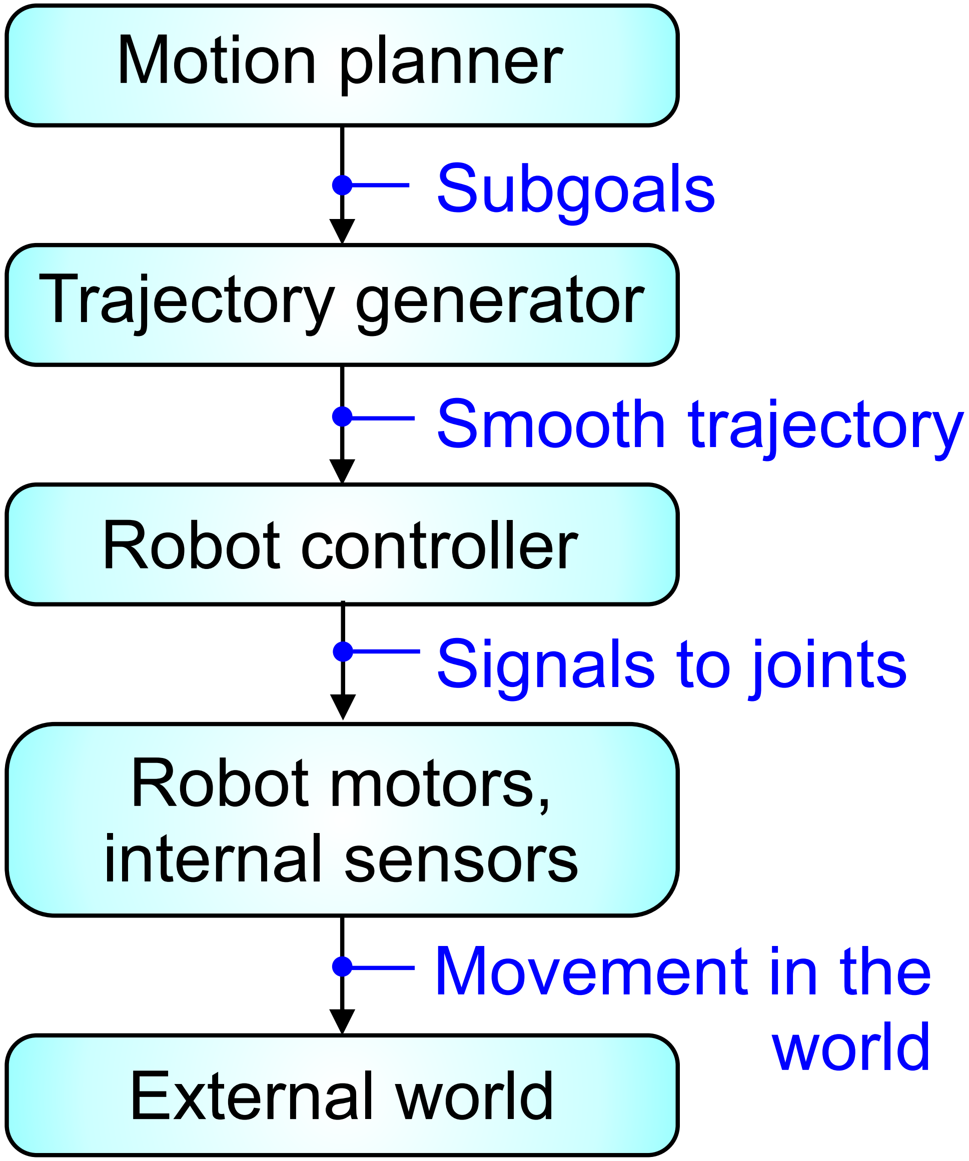
- The intention is to use a single formalism for planning in both most practically useful tasks:
  - Open kinematic chain manipulators.
  - Mobile robots.
- Involved tasks:
  - Localization. *Where am I?*

    E.g. odometry, localization with respect to external sensors (as GPS), beacons or landmarks.
  - Planning/trajectory generation.

    Where am I going? (= goal)

    How do I get there? (= plan or trajectory).
- Complications: errors in maps, sensing, control, changing world, unexpected obstacles, . . .

Motion planning



Global path vs. local trajectory

- Finding a path/trajectory between the robot initial configuration and its desired final configuration can formulated as a single optimization task in theory.

- However, this task has a prohibitively high computational complexity in most cases.

  Why? The robot has several degrees of freedom, which induces a high dimensional search space. The search algorithms (as finding the shortest path in a graph) have exponential complexity.

- The pragmatic solution (divide and conquer) is to separate the path/trajectory finding into two subtasks:

  - Global path planning.

  - Local trajectory generation (local obstacle avoidance in mobile robotics).

Terminology: path vs. trajectory

- *Note: Terms path, trajectory are often confused. They are used as synonyms informally.*
- **Path** is an ordered locus of points in the space (either joint or operational), which the robot should follow.
  - Path provides a pure geometric description of motion.
  - Path is usually planned globally taking into account obstacle avoidance, traversing a complicated maze, etc.
- **Trajectory** is a path augmented with velocities and accelerations in its each point.
  - A design of a trajectory does not need global information, which simplifies the task significantly.
  - The trajectory is specified and designed locally. Parts of a path are covered by individual trajectories.
  - It is often required that pieces of trajectories join smoothly, which induces that a single trajectory design takes into account neighboring trajectories from the path only.

Path/trajectory and a robot world map

- Path planning is strongly related with the global world map and is usually performed off-line (because of the above mentioned high computational complexity).
- Trajectory generation (or local obstacle avoidance) can be regarded locally as independent of a global map. It is usually performed on-line.
- For global path planning, the robot world map should be given (e.g. by a floor plan of a building).
  - If it is not given, a map is obtained off-line before path planning.
  - Sometimes, map building and localization can be done simultaneously. It is called as SLAM (Simultaneous Localization and Mapping).
- With industrial robots, path planning is not usually solved by a computer because a human operator plans the paths.

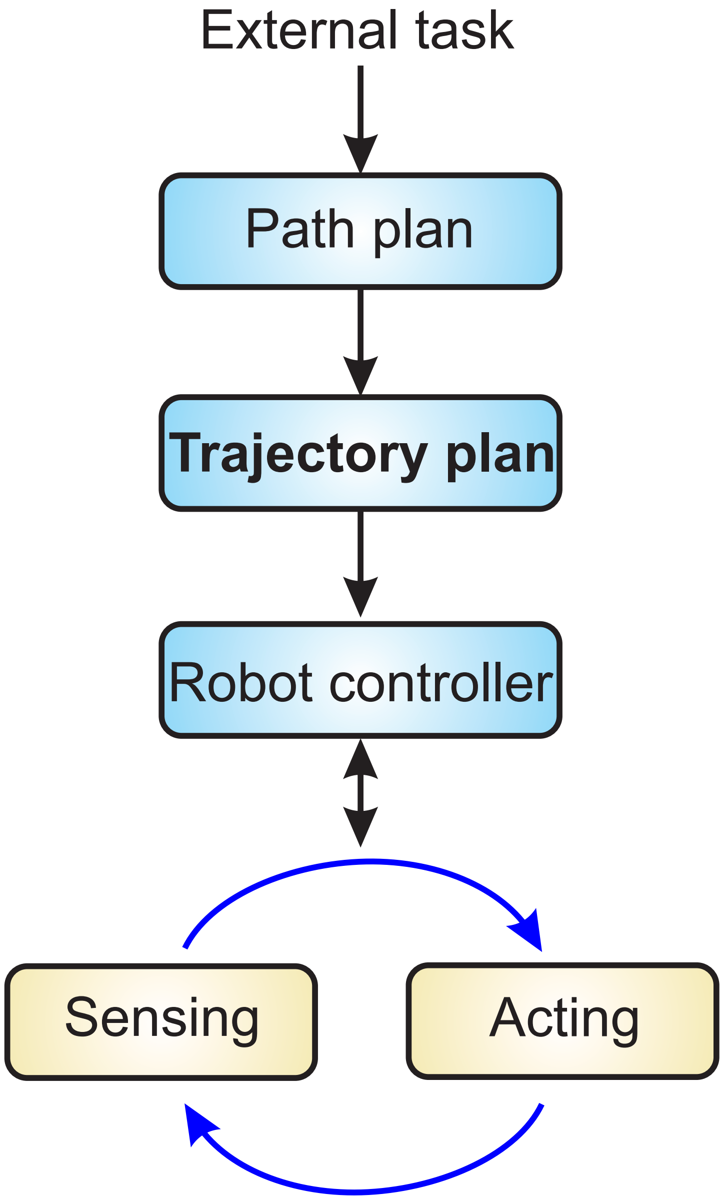
Robot motion planning, an overview

Path planning

- Geometric path.
- Issues: obstacle avoidance, shortest path.

Trajectory generating

- The path planning provides the input – the chunk of a path usually given as a set of points defining the trajectory.
- “Approximate” the desired path chunk by a class of polynomial functions and
- generate a sequence of time-based “control set points” for the control of manipulator from the initial configuration to its destination.



Robot & representations

- 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.

Robot world model (robot environment model)

- Robot representation of its environment is called world model (or environment model).
- Representation types:
  - Continuous: dynamic relations between subsets of a space (e.g. 3D Euclidean space), i.e. function of several variables depending on time.
  - Discrete: e.g., a cell x, y on the floor plan (occupancy grid) contains an obstacle.
  - Symbolic: e.g., this particular manipulation task needs that type of a gripper.

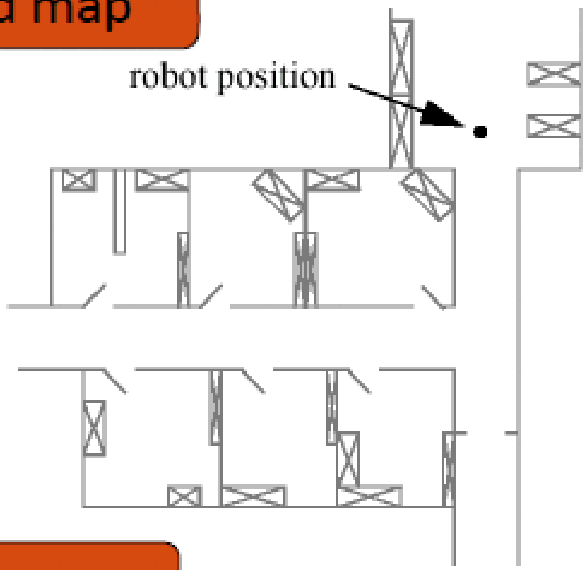
    The relation between symbols is usually expressed using graphs.

Examples: world maps in mobile robotics

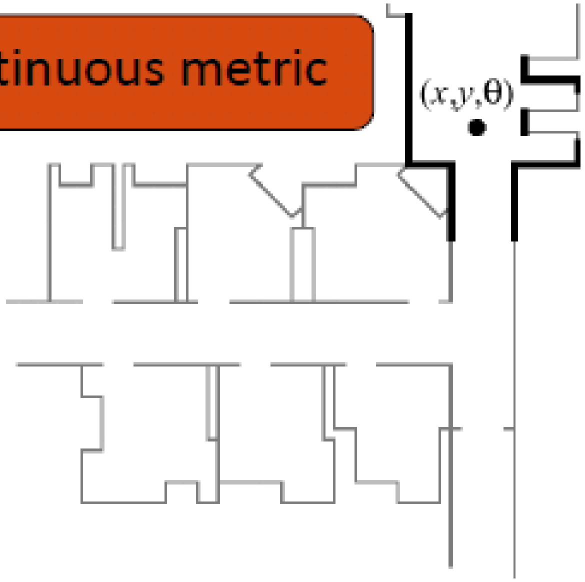
- An odometric path to recall the route traveled.
- A metric map, e.g. a robot may recall a maze by drawing it by using exact lengths of corridors and distances to walls.
- A landmark-based map, which tells the robot what to do at each landmark regardless of order.
- A topological map is a collection of landmarks with links.

World maps illustrated pictorially 1

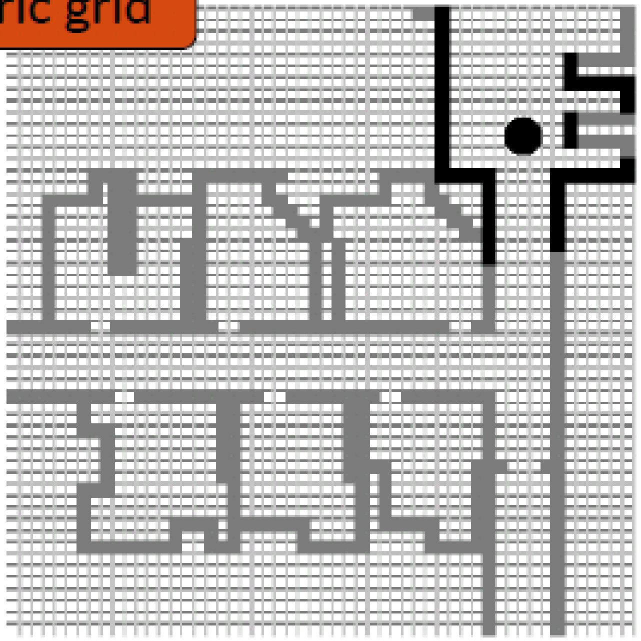
world map



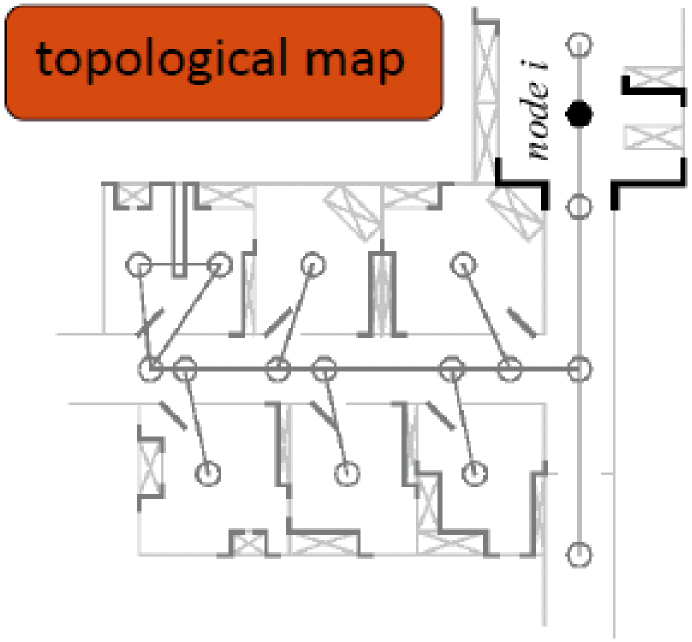
continuous metric



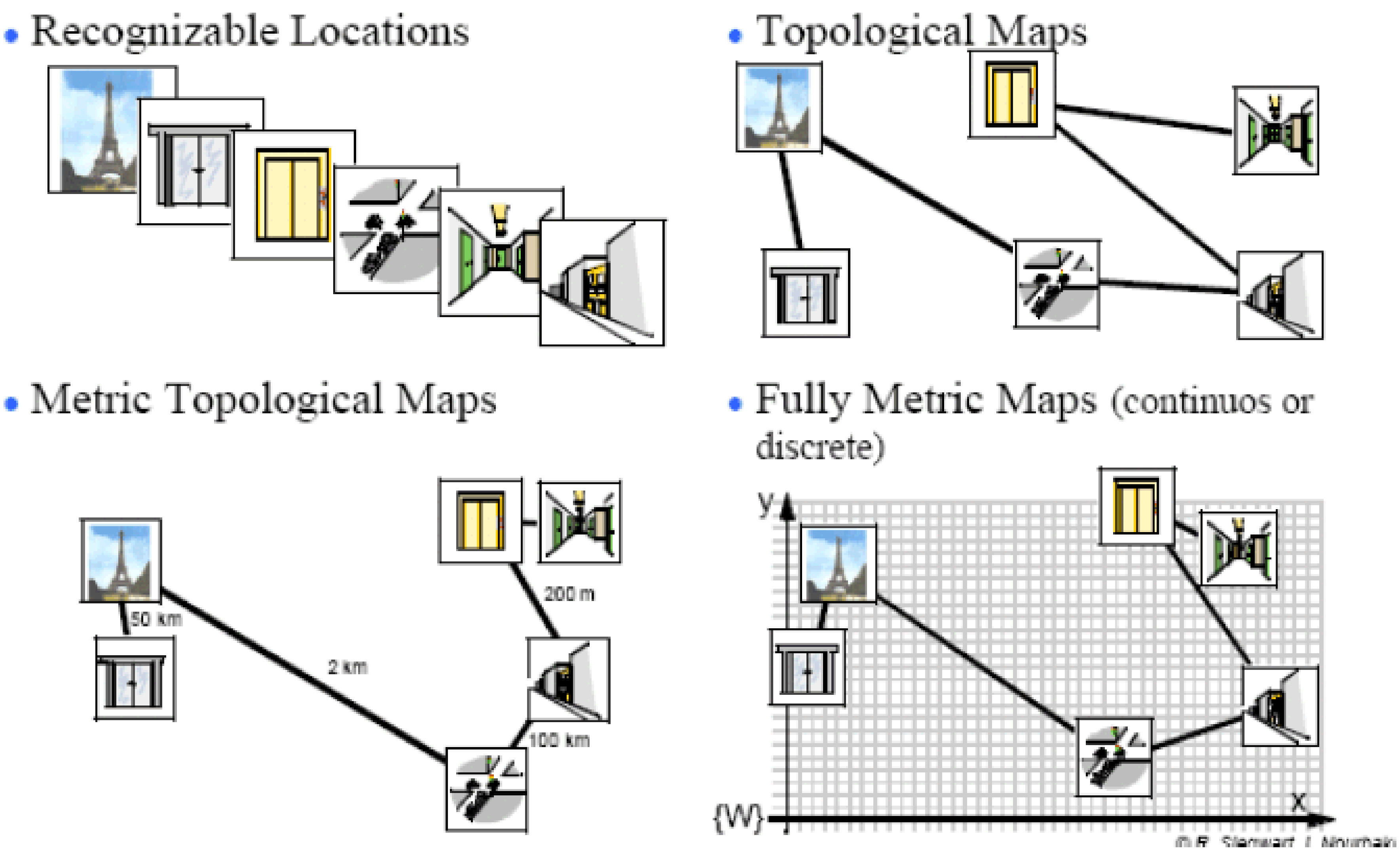
metric grid



topological map



World maps illustrated pictorially 2



- Recognizable Locations
- Topological Maps
- Metric Topological Maps
- Fully Metric Maps (continuos or discrete)

© R. Siegwart, I. Nourbakhsh

Other representations used in robotics

- Robot itself

  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.

Robotic representation & timeline

- Keeping a model updated requires 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 (e.g. detailed metric maps).
- Some models may be quickly constructed and soon discarded (e.g. odometric path).