INTRODUCTION TO ROBOTICS

Autonomous agents and Autonomous robotics

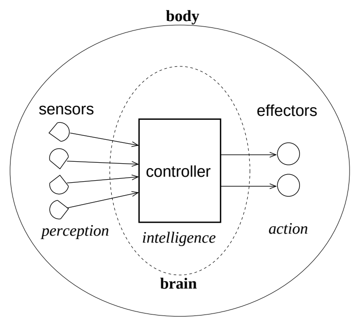
- We will be discussing *autonomous mobile robots*
- What is a robot?
  - "a programmable, multifunction manipulator designed to move material, parts, tools or specific devices through variable programmed motions for the performance of various tasks." [Robot Institute of America]
  - "an active, artificial *agent* whose environment is the physical world" [Russell&Norvig, p773]

- What is an agent?
  - "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." [Russell&Norvig, p32]
- What is autonomy?
  - no remote control!!
  - an agent makes decisions on its own, guided by feedback from its sensors; but you write the program that tells the agent how to make its decisions environment.

Our definition of a *robot*

- *robot* = *autonomous embodied agent*
- has a *body* and a *brain*
- exists in the physical world (rather than the virtual or simulated world)
- is a mechanical device
- contains *sensors* to perceive its own state
- contains *sensors* to perceive its surrounding environment
- possesses *effectors* which perform actions
- has a *controller* which takes input from the sensors, makes *intelligent* decisions about actions to take, and effects those actions by sending commands to motors

Our canonical agent



A bit of robot history

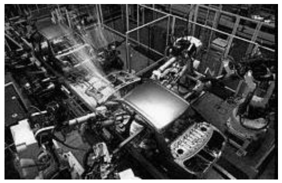
- The word *robot* came from the Czech word *robota*, which means *slave*
- Used first by playwrite Karel Capek, "Rossum's Universal Robots" (1923)
- Human-like automated devices date as far back as ancient Greece
- Modern view of a robot stems from science fiction literature

- Foremost author: Isaac Asimov, "I, Robot" (1950)
- The *Three Laws of Robotics*
  1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
  2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
  3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.
- Hollywood broke these rules: e.g., "The Terminator" (1984)
- Also see Iain Banks "Culture" novels for an intersting exploration on how we might interact with intelligent machines.

Effectors

- Comprise all the mechanisms through which a robot can *effect* changes on itself or its environment
- *Actuator* = the actual mechanism that enables the effector to execute an action; converts software commands into physical motion
- Types:
  - arm
  - leg
  - wheel
  - gripper
- Categories:
  - *manipulator*
  - *mobile*

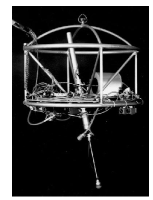
Some manipulator robots

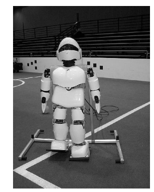


some manipulator robots

Mobile robots

- Classified by manner of locomotion:
  - *wheeled*
  - *legged*
- Stability is important
  - *static stability*
  - *dynamic stability*





Degrees of freedom

- Number of directions in which robot motion can be controlled
- Free body in space has 6 degrees of freedom:
  - three for position (x, y, z)
  - three for orientation $(roll, pitch, yaw)$
    - yaw refers to the direction in which the body is facing i.e., its orientation within the xy plane
    - $roll$ refers to whether the body is upside-down or not i.e., its orientation within the yz plane
    - $pitch$ refers to whether the body is tilted i.e., its orientation within the xz plane
- If there is an actuator for every degree of freedom, then all degrees of freedom are controllable ⇒ *holonomic*
- Most robots are *non-holonomic*

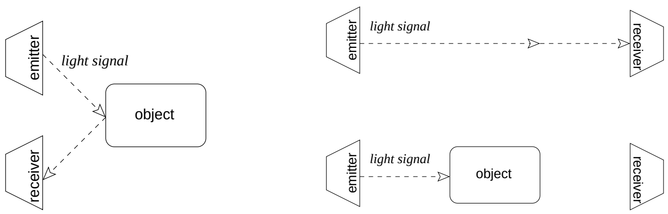
Sensors

- ⇒ Perception
  - *Proprioceptive:* know where your joints/sensors are
  - *Odometry:* know where you are
- Function: to convert a physical property into an electronic signal which can be interpreted by the robot in a useful way

Property being sensed	type of sensor
contact	bump, switch
distance	ultrasound, radar, infra red (IR)
light level	photo cell, camera
sound level	microphone
temperature	thermal
rotation	encoder

More on sensors

- Operation
  - *Passive*: read a property of the environment
  - *Active*: act on the environment and read the result



reflectance *break-beam*

More on sensors

- noise
  - *internal*: from inside the robot
  - *external*: from the robot's environment
  - *calibration*: can help eliminate/reduce noise

Environment

- *Accessible* vs *inaccessible*
  - robot has access to all necessary information required to make an informed decision about to do next
- *Deterministic* vs *nondeterministic*
  - any action that a robot undertakes has only one possible outcome.
- *Episodic* vs *non-episodic*
  - the world proceeds as a series of repeated episodes.

Environment

- *Static* vs *dynamic*
  - the world changes by itself, not only due to actions effected by the robot
- *Discrete* vs *continuous*
  - sensor readings and actions have a discrete set of values.

State

- Knowledge about oneself and one's environment
  - *Kinematics* = study of correspondance between actuator mechanisms and resulting motion
    - motion:
      - rotary
      - linear
  - Combines sensing and acting
  - *Did I go as far as I think I went?*
- But one's environment is full of information
- For an agent, what is relevant?

Control

- Autonomy
- Problem solving
- Modeling
  - knowledge
  - representation
- Control architectures
- Deliberative control
- Reactive control
- Hybrid control

Autonomy

- To be truly autonomous, it is not enough for a system simply to establish direct numerical relations between sensor inputs and effector outputs
- A system must be able to accomplish *goals*
- A system must be able to *solve problems*
- ⇒ Need to represent problem space
  - which contains goals
  - and intermediate states
- There is always a trade-off between *generality* and *efficiency*
  - more specialized ⇒ more efficient
  - more generalized ⇒ less efficient

Problem solving: example

- GPS = General Problem Solver [Newell and Simon 1963]
- Means-Ends analysis

operator	*preconditions*	*results*
$PUSH(obj, loc)$	$at(robot, obj) \wedge large(obj) \wedge$ $clear(obj) \wedge armempty()$	$at(obj, loc) \wedge$ $at(robot, loc)$
$CARRY(obj, loc)$	$at(robot, obj) \wedge small(obj)$	$at(obj, loc) \wedge$ $at(robot, loc)$
$WALK(loc)$	$none$	$at(robot, loc)$
$PICKUP(obj)$	$at(robot, obj)$	$holding(obj)$
$PUTDOWN(obj)$	$holding(obj)$	$\neg holding(obj)$
$PLACE(obj1, obj2)$	$at(robot, obj2) \wedge holding(obj1)$	$on(obj1, obj2)$

Modeling the robot's environment

- Modeling
  - the way in which *domain knowledge* is embedded into a control system
  - information about the environment stored internally: *internal representation*
  - e.g., maze: robot stores a *map* of the maze "in its head"

- Knowledge
  - information in a context
  - organized so it can be readily applied
  - understanding, awareness or familiarity acquired through learning or experience
  - physical structures which have correlations with aspects of the environment and thus have a predictive power for the system

Memory

- Divided into 2 categories according to duration
- *Short term memory (STM)*
  - transitory
  - used as a buffer to store only recent sensory data
  - data used by only one behaviour
  - examples:
    - *avoid-past*: avoid recently visited places to encourage exploration of novel areas
    - *wall-memory*: store past sensor readings to increase correctness of wall detection

Memory

- *Long term memory (LTM)*
  - persistent
  - *metric maps*: use absolute measurements and coordinate systems
  - *qualitative maps*: use landmarks and their relationships
  - examples:
    - *Markov models*: graph representation which can be augmented with probabilities for each action associated with each sensed state

Knowledge representation

- Must have a relationship to the environment (temporal, spatial)
- Must enable predictive power (look-ahead), but if inaccurate, it can deceive the system
- *Explicit*: symbolic, discrete, manipulable
- *Implicit*: embedded within the system
- *Symbolic*: connecting the meaning (semantics) of an arbitrary symbol to the real world
- Difficult because:
  - sensors provide signals, not symbols
  - symbols are often defined with other symbols (circular, recursive)
  - requires interaction with the world, which is noisy

Components of knowledge representation

- *State*
  - totally vs partially vs un- observable
  - discrete vs continuous
  - static vs dynamic
- *Spatial*: navigable surroundings and their structure; metric or topological maps
- *Objects*: categories and/or instances of detectable things in the world

Components of knowledge representation

- *Actions*: outcomes of specific actions on the self and the environment
- *Self/ego*: stored proprioception (sensing internal state), self-limitations, capabilities
  - *perceptive*: how to sense
  - *behaviour*: how to act
- *Intentional*: goals, intended actions, plans
- *Symbolic*: abstract encoding of state/information

Types of representations

- maps
  - *Euclidean map*
    - represents each point in space according to its metric distance to all other points in the space
  - *Topological map*
    - represents locations and their connections, i.e., how/if they can be reached from one another; but does not contain exact metrics
  - *Cognitive map*
    - represents behaviours; can store both previous experience and use for action
    - used by animals that forage and home (animal navigation)
    - may be simple collections of vectors

Control architecture

- A control architecture provides a set of principles for organizing a control system
- Provides structure
- Provides constraints
- Refers to software control level, not hardware!
- Implemented in a programming language
- Don't confuse "programming language" with "robot architecture"
- Architecture guides how programs are structured

Classes of robot control architectures

- *Deliberative*
  - look-ahead; think, plan, then act
- *Reactive*
  - don't think, don't look ahead, just react!
- *Hybrid*
  - think but still act quickly
- *Behaviour-based*
  - distribute thinking over acting

Deliberative control

- Classical control architecture (first to be tried)
- First used in AI to reason about actions in non-physical domains (like chess)
- Natural to use this in robotics at first
- Example: Shakey (1960's, SRI)
  - state-of-the-art machine vision used to process visual information
  - used classical planner (STRIPS)

- Planner-based architecture
  1. sensing (S)
  2. planning (P)
  3. acting (A)
- Requirements
  - lots of time to think
  - lots of memory
  - (but the environment changes while the controller thinks)

Reactive control

- Operate on a short time scale
- Does not look ahead
- Based on a tight loop connecting the robot's sensors with its effectors
- Purely reactive controllers do not use any internal representation; they merely react to the current sensory information
- Collection of rules that map situations to actions
  - simplest form: divide the perceptual world into a set of mutually exclusive situations recognize which situation we are in and react to it
  - (but this is hard to do!)

Hybrid control

- Use the best of both worlds (deliberative and reactive)
- Combine open-loop and closed-loop execution
- Combine different time scales and representations
- Typically consists of three layers:
  1. reactive layer
  2. planner (deliberative layer)
  3. integration layer to combine them
  4. (but this is hard to do!)