Bharath
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Declared as Deemed - to - be - University under section 3 of UGC Act 1956)
BHARATH INSTITUTE OF SCIENCE AND TECHNOLOGY
No.173, Agharam Road, Selaiyur, Chennai - 600 073.

BEE009 - Robotics and Automation

Compiled by,

Dr.S.P.Vijayaragavan
Associate Professor
Department of EEE
BIHER .

Motivation

- Intelligent Environments are aimed at improving the inhabitants' experience and task performance
  - Automate functions in the home
  - Provide services to the inhabitants
- Decisions coming from the decision maker(s) in the environment have to be executed.
  - Decisions require actions to be performed on devices
  - Decisions are frequently not elementary device interactions but rather relatively complex commands
    - Decisions define set points or results that have to be achieved
    - Decisions can require entire tasks to be performed

Automation and Robotics in Intelligent Environments

- Control of the physical environment
  - Automated blinds
  - Thermostats and heating ducts
  - Automatic doors
  - Automatic room partitioning
- Personal service robots
  - House cleaning
  - Lawn mowing
  - Assistance to the elderly and handicapped
  - Office assistants
  - Security services

Robots

"A device with degrees of freedom that can be controlled."

- Class 1 : Manual handling device
- Class 2 : Fixed sequence robot
- Class 3 : Variable sequence robot
- Class 4 : Playback robot
- Class 5 : Numerical control robot
- Class 6 : Intelligent robot

A Brief History of Robotics

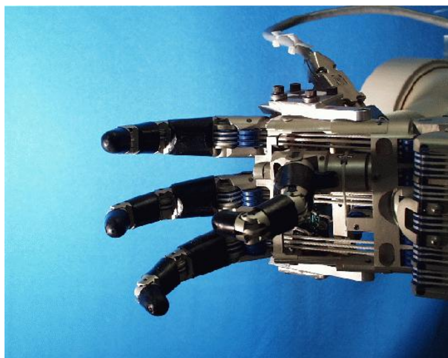
- Mechanical Automata
  - Ancient Greece & Egypt
    - Water powered for ceremonies
  - 14th – 19th century Europe
    - Clockwork driven for entertainment
- Motor driven Robots
  - 1928: First motor driven automata
  - 1961: Unimate
    - First industrial robot
  - 1967: Shakey
    - Autonomous mobile research robot
  - 1969: Stanford Arm
    - Dextrous, electric motor driven robot arm

Maillardet's Automaton

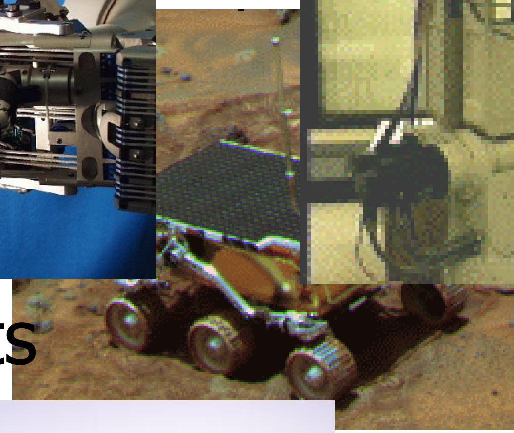
Unimate

Robots

- Robot Manipulators

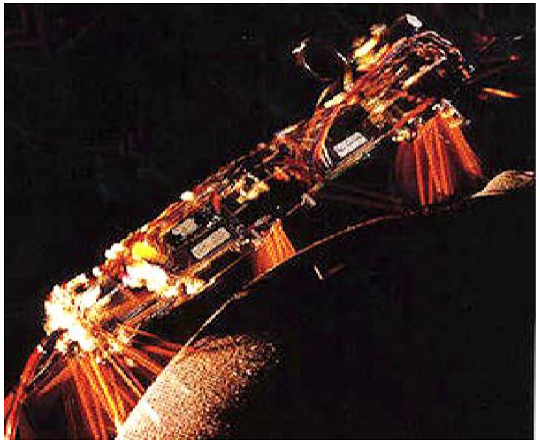


- Mobile Robots

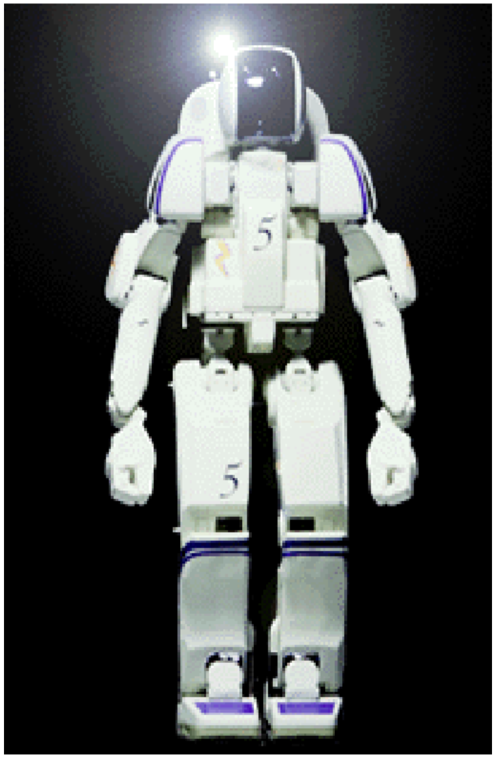


Robots

- Walking Robots



- Humanoid Robots



Autonomous Robots

- The control of autonomous robots involves a number of subtasks
  - Understanding and modeling of the mechanism
    - Kinematics, Dynamics, and Odometry
  - Reliable control of the actuators
    - Closed-loop control
  - Generation of task-specific motions
    - Path planning
  - Integration of sensors
    - Selection and interfacing of various types of sensors
  - Coping with noise and uncertainty
    - Filtering of sensor noise and actuator uncertainty
  - Creation of flexible control policies
    - Control has to deal with new situations

Traditional Industrial Robots

- Traditional industrial robot control uses robot arms and largely pre-computed motions
  - Programming using “teach box”
  - Repetitive tasks
  - High speed
  - Few sensing operations
  - High precision movements
  - Pre-planned trajectories and task policies
  - No interaction with humans

Problems

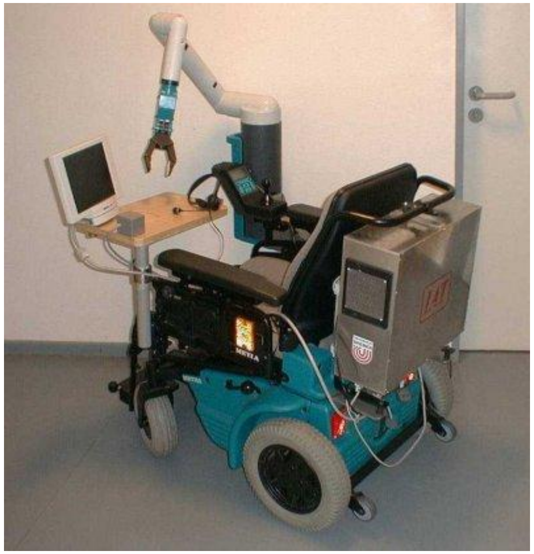
- Traditional programming techniques for industrial robots lack key capabilities necessary in intelligent environments
  - Only limited on-line sensing
  - No incorporation of uncertainty
  - No interaction with humans
  - Reliance on perfect task information
  - Complete re-programming for new tasks

Requirements for Robots in Intelligent Environments

- Autonomy
  - Robots have to be capable of achieving task objectives without human input
  - Robots have to be able to make and execute their own decisions based on sensor information
- Intuitive Human-Robot Interfaces
  - Use of robots in smart homes can not require extensive user training
  - Commands to robots should be natural for inhabitants
- Adaptation
  - Robots have to be able to adjust to changes in the environment

Robots for Intelligent Environments

- Service Robots
  - Security guard
  - Delivery
  - Cleaning
  - Mowing
- Assistance Robots
  - Mobility
  - Services for elderly and People with disabilities

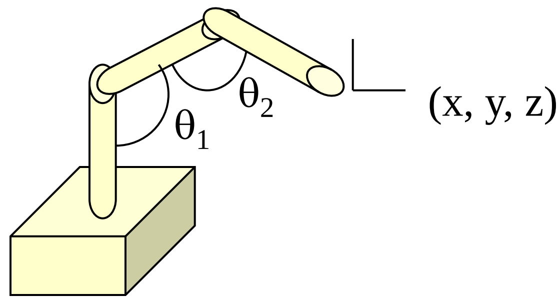


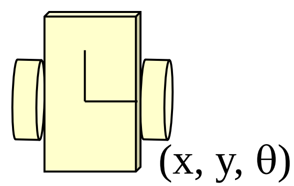
Autonomous Robot Control

- To control robots to perform tasks autonomously a number of tasks have to be addressed:
  - Modeling of robot mechanisms
    - Kinematics, Dynamics
  - Robot sensor selection
    - Active and passive proximity sensors
  - Low-level control of actuators
    - Closed-loop control
  - Control architectures
    - Traditional planning architectures
    - Behavior-based control architectures
    - Hybrid architectures

Modeling the Robot Mechanism

- Forward kinematics describes how the robots joint angle configurations translate to locations in the world





- Inverse kinematics computes the joint angle configuration necessary to reach a particular point in space.
- Jacobians calculate how the speed and configuration of the actuators translate into velocity of the robot

Mobile Robot Odometry

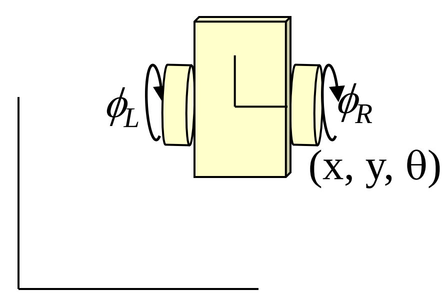
- In mobile robots the same configuration in terms of joint angles does not identify a unique location
  - To keep track of the robot it is necessary to incrementally update the location (this process is called odometry or dead reckoning)

$$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}^{t+\Delta t} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}^{t} + \begin{pmatrix} v_x \\ v_y \\ \varpi \end{pmatrix} \Delta t$$

- Example: A differential drive robot

$$v_x = \cos(\theta)\frac{r(\dot{\phi}_L + \dot{\phi}_R)}{2}, v_y = \sin(\theta)\frac{r(\dot{\phi}_L + \dot{\phi}_R)}{2}$$

$$\varpi = \frac{r}{d}\left(\dot{\phi}_L - \dot{\phi}_R\right)$$



Actuator Control

- To get a particular robot actuator to a particular location it is important to apply the correct amount of force or torque to it.
  - Requires knowledge of the dynamics of the robot
    - Mass, inertia, friction
    - For a simplistic mobile robot: $F = m\,a + B\,v$
  - Frequently actuators are treated as if they were independent (i.e. as if moving one joint would not affect any of the other joints).
  - The most common control approach is PD-control (proportional, differential control)
    - For the simplistic mobile robot moving in the x direction:

$$F = K_P\left(x_{desired} - x_{actual}\right) + K_D\left(v_{desired} - v_{actual}\right)$$

Robot Navigation

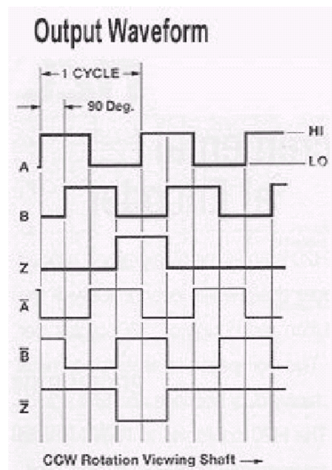
- Path planning addresses the task of computing a trajectory for the robot such that it reaches the desired goal without colliding with obstacles
  - Optimal paths are hard to compute in particular for robots that can not move in arbitrary directions (i.e. nonholonomic robots)
  - Shortest distance paths can be dangerous since they always graze obstacles
  - Paths for robot arms have to take into account the entire robot (not only the endeffector)

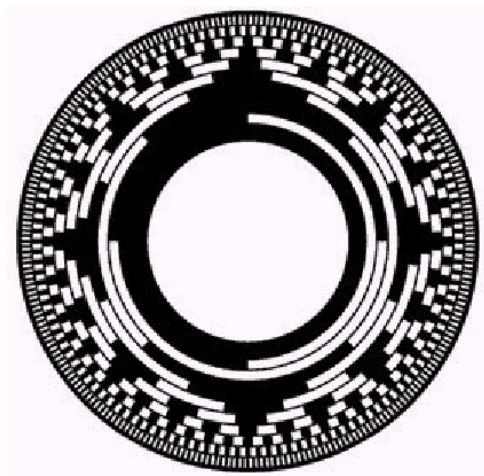
Sensor-Driven Robot Control

- To accurately achieve a task in an intelligent environment, a robot has to be able to react dynamically to changes ion its surrounding
  - Robots need sensors to perceive the environment
  - Most robots use a set of different sensors
    - Different sensors serve different purposes
  - Information from sensors has to be integrated into the control of the robot

Robot Sensors

- Internal sensors to measure the robot configuration
  - Encoders measure the rotation angle of a joint

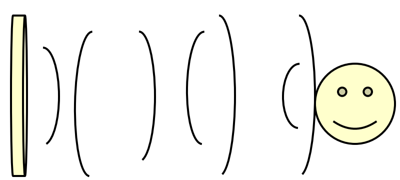




  - Limit switches detect when the joint has reached the limit

Robot Sensors

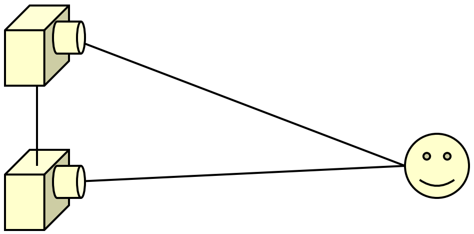
- Proximity sensors are used to measure the distance or location of objects in the environment. This can then be used to determine the location of the robot.
  - Infrared sensors determine the distance to an object by measuring the amount of infrared light the object reflects back to the robot
  - Ultrasonic sensors (sonars) measure the time that an ultrasonic signal takes until it returns to the robot



  - Laser range finders determine dista measuring either the time it takes for a laser beam to be reflected back to the robot or by measuring where the laser hits the object

Robot Sensors

- Computer Vision provides robots with the capability to passively observe the environment
  - Stereo vision systems provide complete location information using triangulation



  - However, computer vision is very complex
    - Correspondence problem makes stereo vision even more difficult

Uncertainty in Robot Systems

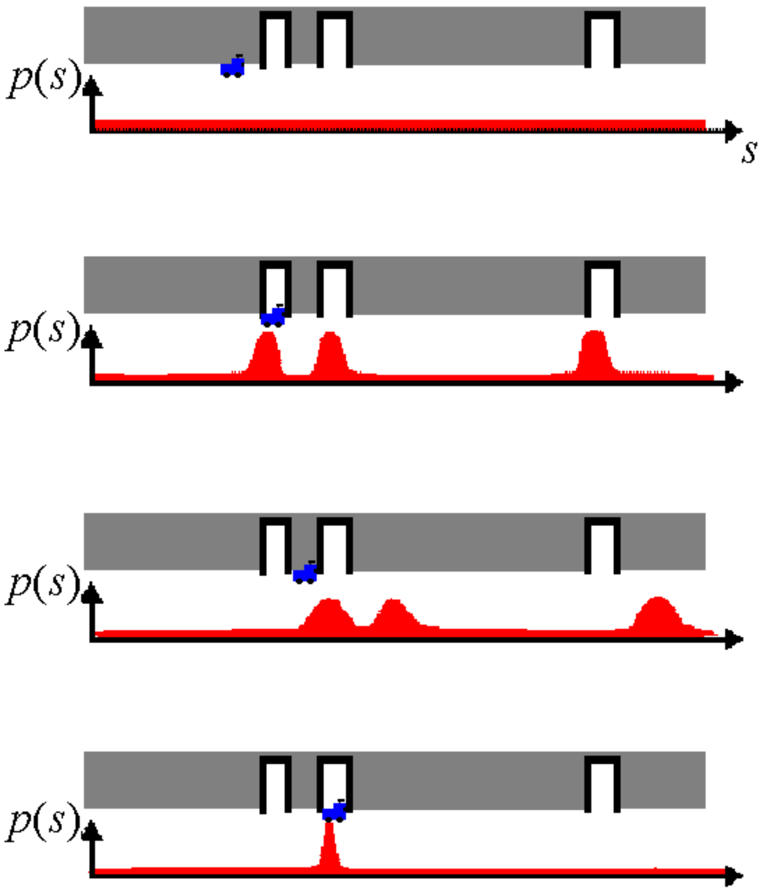
- Robot systems in intelligent environments have to deal with sensor noise and uncertainty
  - Sensor uncertainty
    - Sensor readings are imprecise and unreliable
  - Non-observability
    - Various aspects of the environment can not be observed
    - The environment is initially unknown
  - Action uncertainty
    - Actions can fail
    - Actions have nondeterministic outcomes

Probabilistic Robot Localization

- Explicit reasoning about Uncertainty using Bayes filters:

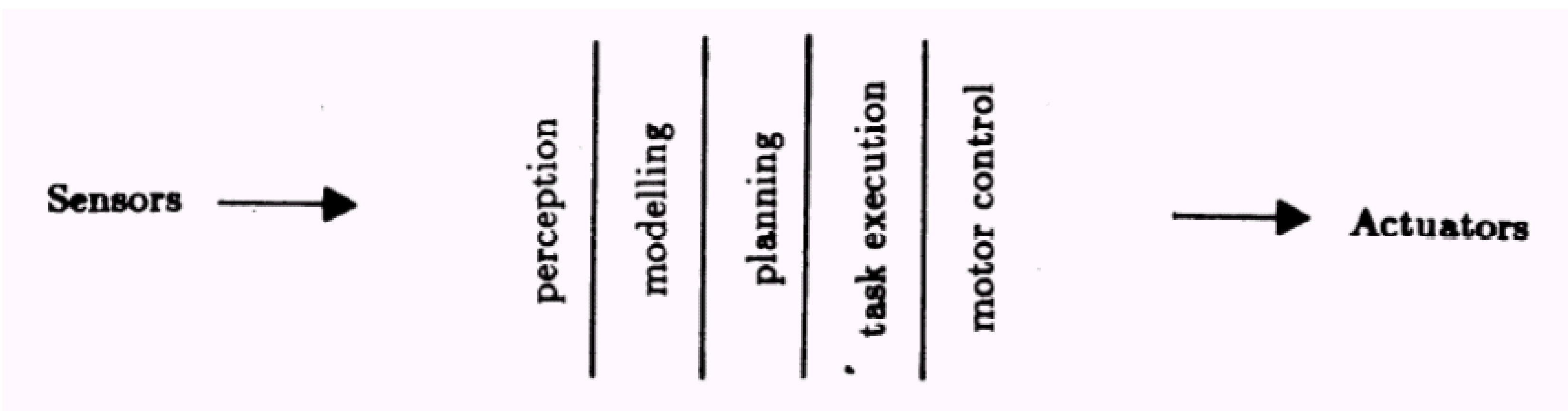
$$b(x_t) = \eta \; p(o_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1}) \, b(x_{t-1}) \, dx_{t-1}$$

- Used for:
  - Localization
  - Mapping
  - Model building



Deliberative Robot Control Architectures

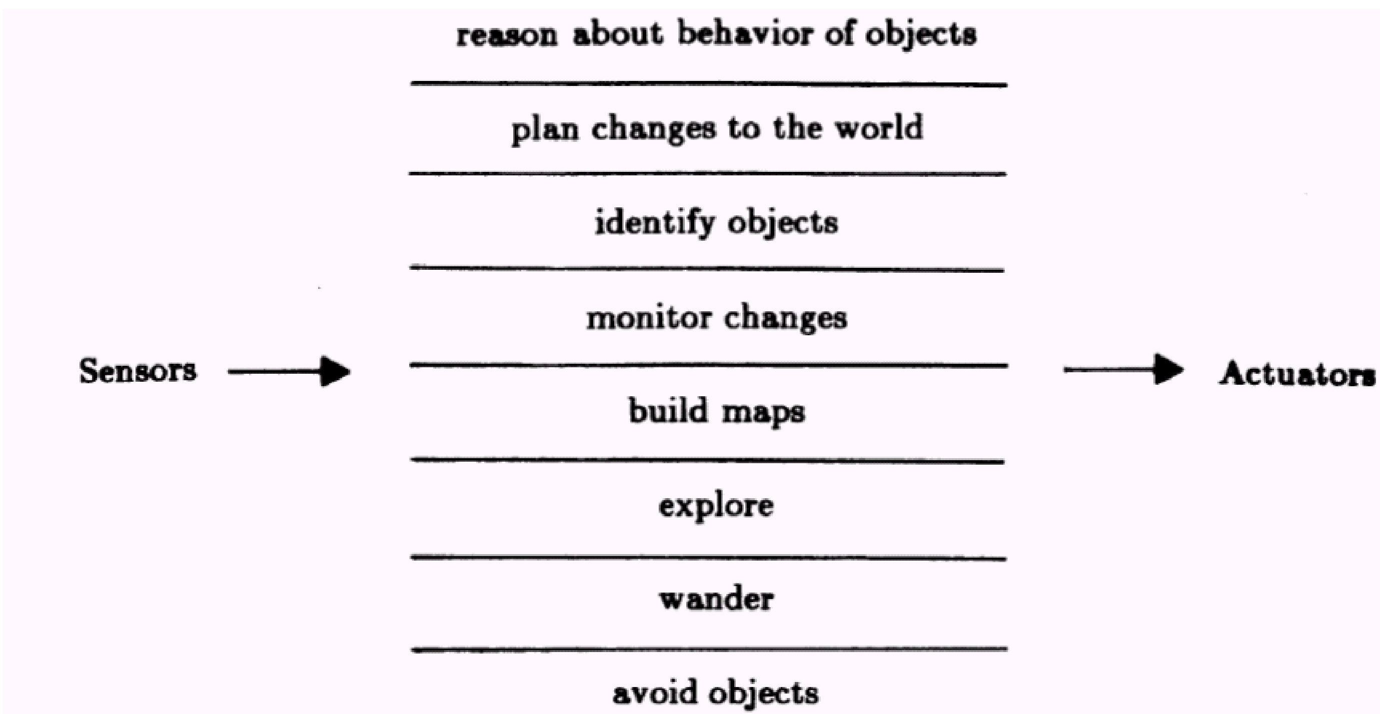
- In a deliberative control architecture the robot first plans a solution for the task by reasoning about the outcome of its actions and then executes it



  – Control process goes through a sequence of sensing, model update, and planning steps

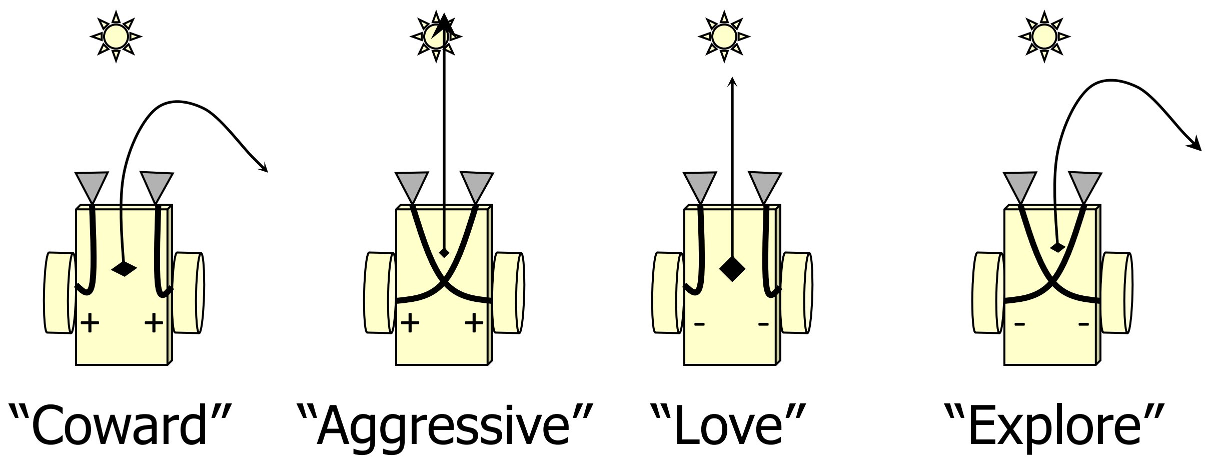
Behavior-Based Robot Control Architectures

- In a behavior-based control architecture the robot's actions are determined by a set of parallel, reactive behaviors which map sensory input and state to actions.



Complex Behavior from Simple Elements: Braitenberg Vehicles

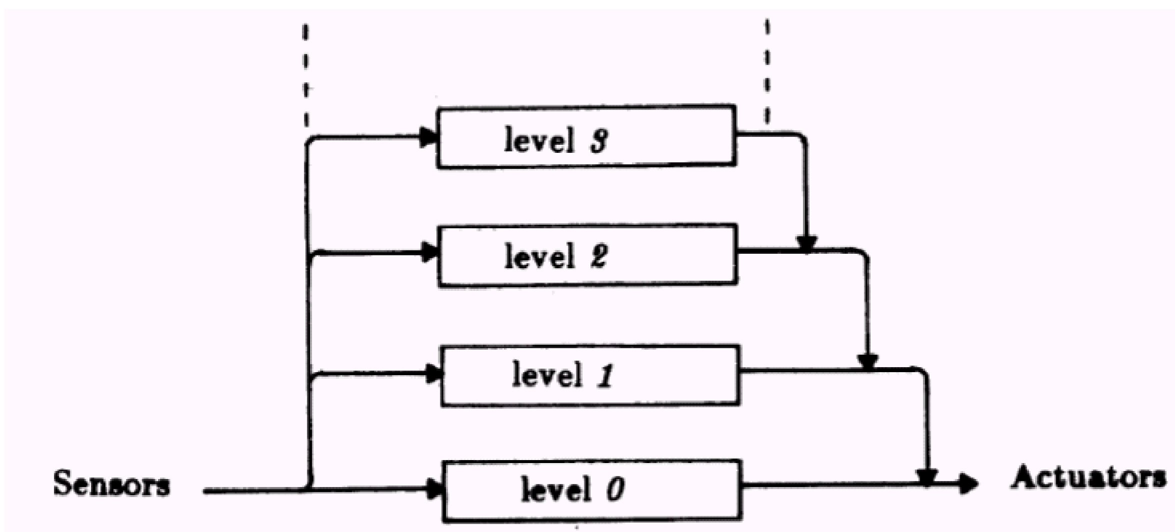
- Complex behavior can be achieved using very simple control mechanisms
  - Braitenberg vehicles: differential drive mobile robots with two light sensors



- Complex external behavior does not necessarily require a complex reasoning mechanism

Behavior-Based Architectures: Subsumption Example

- Subsumption architecture is one of the earliest behavior-based architectures
  - Behaviors are arranged in a strict priority order where higher priority behaviors subsume lower priority ones as long as they are not inhibited.

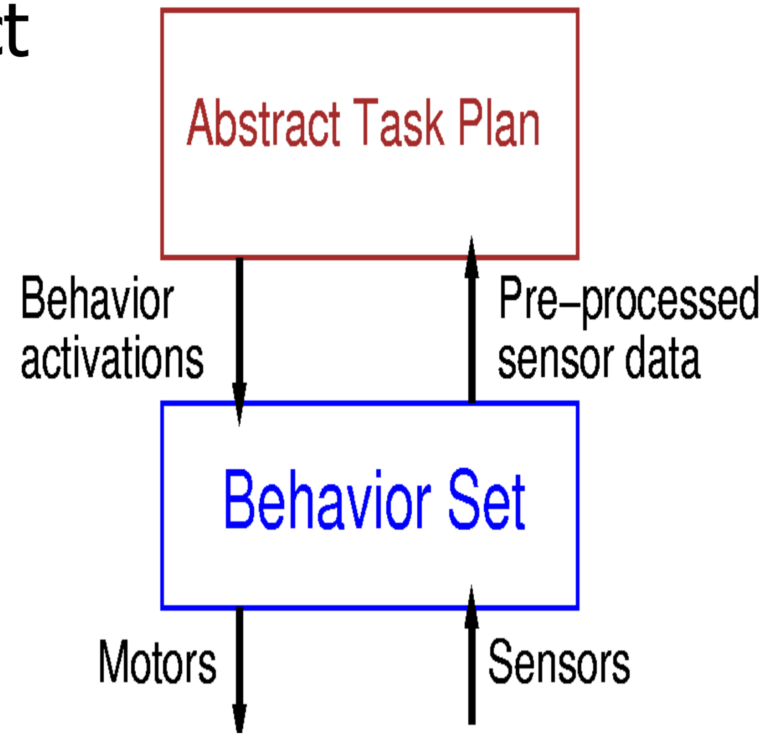


Reactive, Behavior-Based Control Architectures

- Advantages
  - Reacts fast to changes
  - Does not rely on accurate models
    - “The world is its own best model”
  - No need for replanning
- Problems
  - Difficult to anticipate what effect combinations of behaviors will have
  - Difficult to construct strategies that will achieve complex, novel tasks
  - Requires redesign of control system for new tasks

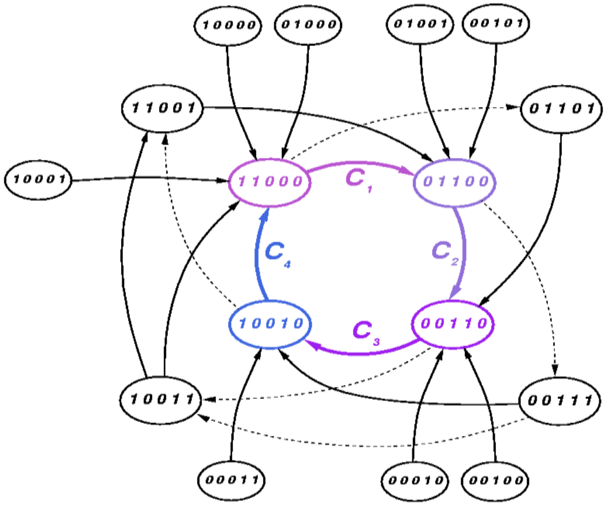
Hybrid Control Architectures

- Hybrid architectures combine reactive control with abstract task planning
  - Abstract task planning layer
    - Deliberative decisions
    - Plans goal directed policies
  - Reactive behavior layer
    - Provides reactive actions
    - Handles sensors and actuators

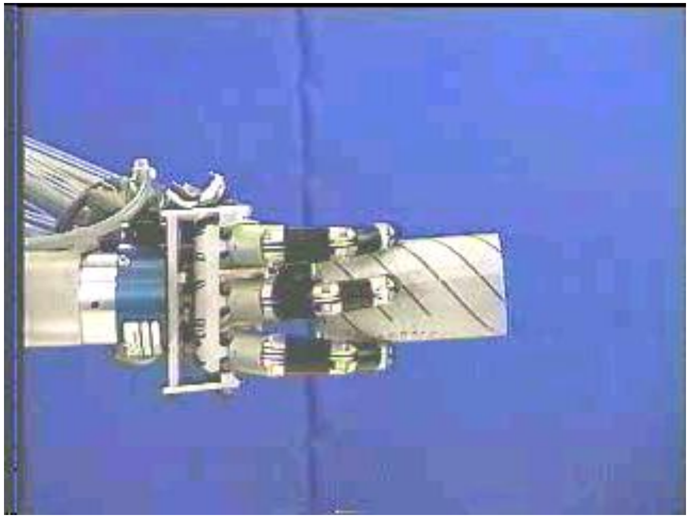


Hybrid Control Policies

Task Plan:



Behavioral Strategy:



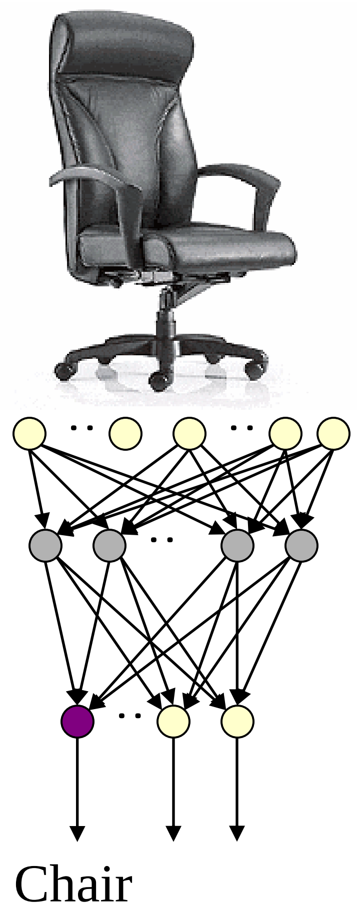
Example Task: Changing a Light Bulb

Traditional Human-Robot Interface: Teleoperation

- Remote Teleoperation: Direct operation of the robot by the user
  - User uses a 3-D joystick or an exoskeleton to drive the robot
    - Simple to install
    - Removes user from dangerous areas
  - Problems:
    - Requires insight into the mechanism
    - Can be exhaustive
    - Easily leads to operation errors

Learning Sensory Patterns

- Learning to Identify Objects
  - How can a particular object be recognized ?
    - Programming recognition strategies is difficult because we do not fully understand how we perform recognition
    - Learning techniques permit the robot system to form its own recognition strategy
  - Supervised learning can be used by giving the robot a set of pictures and the corresponding classification
    - Neural networks
    - Decision trees



Example: Reinforcement Learning in a Hybrid Architecture

- Policy Acquisition Layer
  - Learning tasks without supervision
- Abstract Plan Layer
  - Learning a system model
  - Basic state space compression
- Reactive Behavior Layer
  - Initial competence and reactivity