ACDS Lecture Series

Lecture - 19

CSIR

Robotics/Intelligent Machine

G. N. Sastry and Team

ADVANCED COMPUTATION AND DATA SCIENCES (ACDS) DIVISION

CSIR-North East Institute of Science and Technology, Jorhat, Assam, India

19. Robotics/ Intelligent Machine

19.1 Introduction

What is artificial intelligence??

Artificial Intelligence is intelligence displayed by machines, in contrast with the natural intelligence displayed by humans and other animals.

- Uses of AI:
- Computer science
- Education
- Finance
- Toys and games

Advantages of AI

- Nonhuman intelligences
- No need for human testers.

Disadvantages of AI

- If AI programmed is corrupted it can something devastating.
- The AI is programmed to do something beneficial, but it develops a destructive method for achieving its goal.

- A robot is a machine which is capable of carrying out a complex series of actions automatically.

- USES:
  Factory
  Space
  Military
  Mining
  Healthcare,
  Home

Advantages of Robots

- Robots helps us in many ways.
- It can lift heavy loads.
- It can also do minor surgery.

Disadvantages of Robots

- It costs a leg and a arm.
- Robots need a continuous supply of power
- People can lose their jobs.
- Maintenance and repair are expensive.

Laws of Robotics

First Law :

A robot may not injure a human being, or, through inaction, allow a human being to come to harm, unless this would violate a higher order law

Second Law:

A robot must obey orders given it by human beings, except where such orders would conflict with a higher order law

Third Law :

A robot must protect its own existence as long as such protection does not conflict with a higher order law.

The First Robot

- Unimate was the first industrial robot, which worked on a General Motors assembly line at the Inland Fisher Guide Plant in Ewing Township, New Jersey, in 1961.

- It was invented by George Devol in the 1950s

19.1.1 Biological and cognitive paradigms for robot designs

Primary purpose:

- Robotics in biology gives high valuable experiments related to researchers and development of life science.
- Those experiments which include biological sample in large numbers.
- Those experiment involves the delivery and dispensation of biological solution/samples in large number each with very small volume.
- Robotics gives an important medical diagnosis.

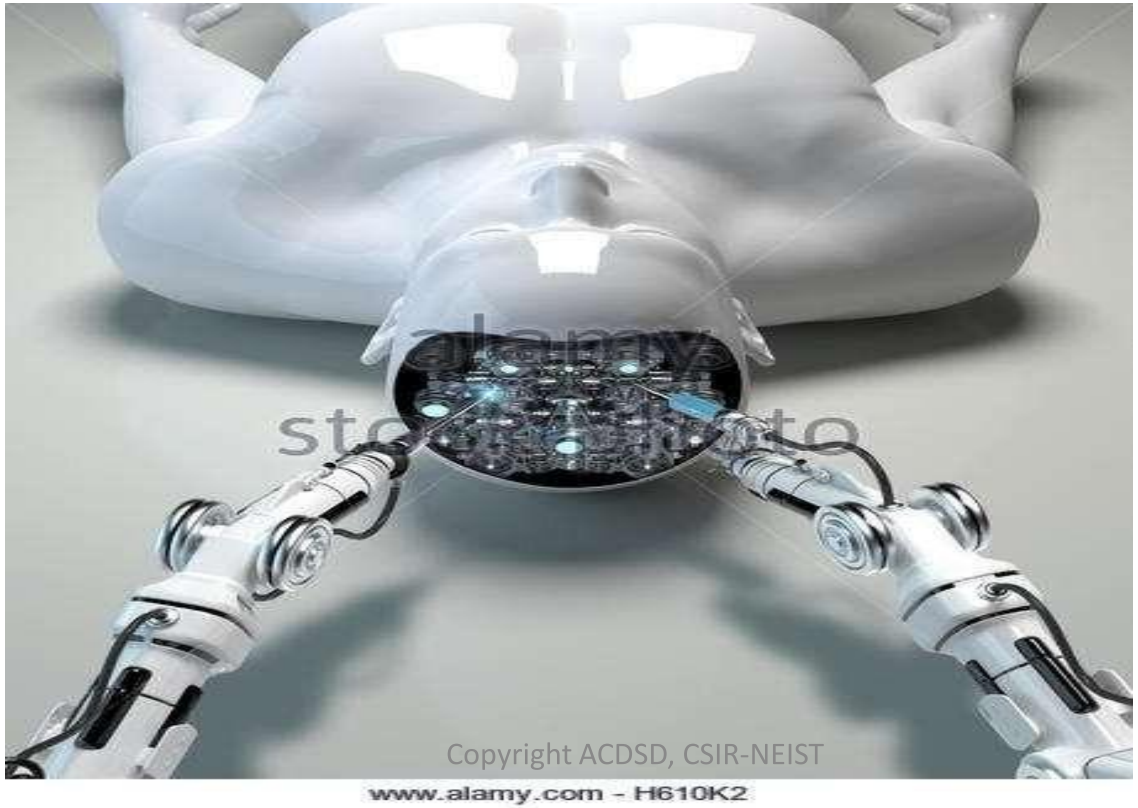
Secondary purpose:

- Robotics for biomedical applications is for effective HANDLINE
- exploration(penetrating) of molecular and cell biology.
- This type of application include cell manipulation(handling or controlling).
- It also includes prolonged DNA INSERTION,CELL injection.

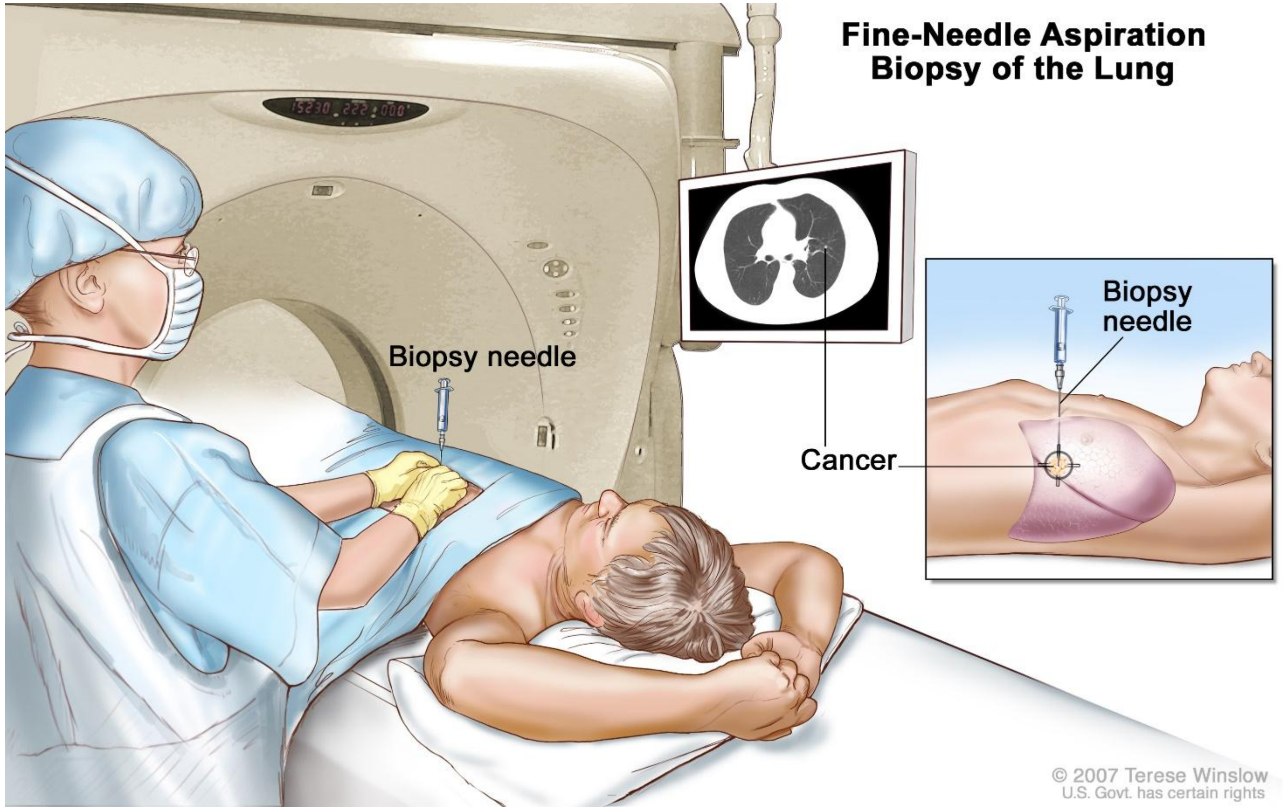
Robotics and automation technologies used in biological and medical applications

Cell manipulation : It shapes and manipulates the cell of any sort of an organism including the function of the cell . It circulates Oxygen and makes cell heal all sorts of wounds.

- **Stereotactic Brain Surgery:** It is used by neurosurgeons to locate the small targets in the brain.it is operated when the person is inside the MRI scan.



Lung Biopsy : Robotic system automatically takes lung biopsy sample under CT fluoroscopy.



Medicine : It is non-invasive surgeries and diagnosis precision and repeatable of robots means consistency and quality.

Robotic tools, devices and systems

- General purposes and Robotic devices
- Special purposes and robotic and devices
- Sensors:
  - Visual sensing: Used as vision based localisation,navigation, tracking are crucial for intelligent robots.
  - Force sensing: Gives input to the robot control which uses for modification of robot’s behaviour.
  - NEURO sensing: It controls the system based on the interaction of computers.

19.1.2 Robotics and AI

Definition of Robotics

- A robot is…

“An active artificial agent whose environment is the physical world”

--Russell and Norvig

“A programmable, multifunction manipulator designed to move material, parts, tools or specific devices through variable programmed motions for the performance of a variety of tasks”

--Robot Institute of America

Relevance to Artificial Intelligence

- Effectors
- Sensors
- Architecture
- Integration of various inputs
  - Hierarchy of information representation
- Emotions

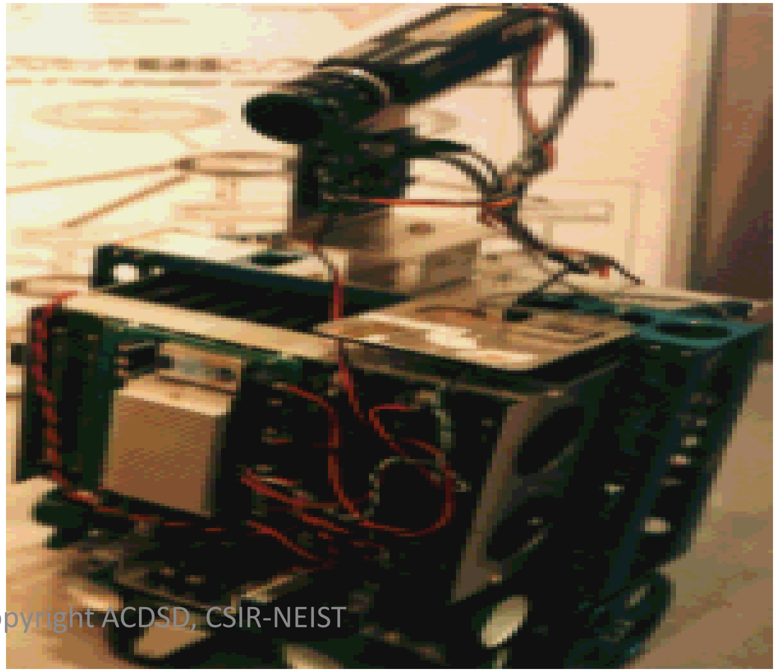
Effectors

- Effector vs. Actuator
- Degrees of freedom (d.f.)
  - 6 d.f. for free body in space
- Locomotion
  - Statically stable vs. Dynamically stable
- Manipulation
  - Rotary vs. Prismatic motion
  - End Effector

Four-finger Utah/MIT hand

Sensors

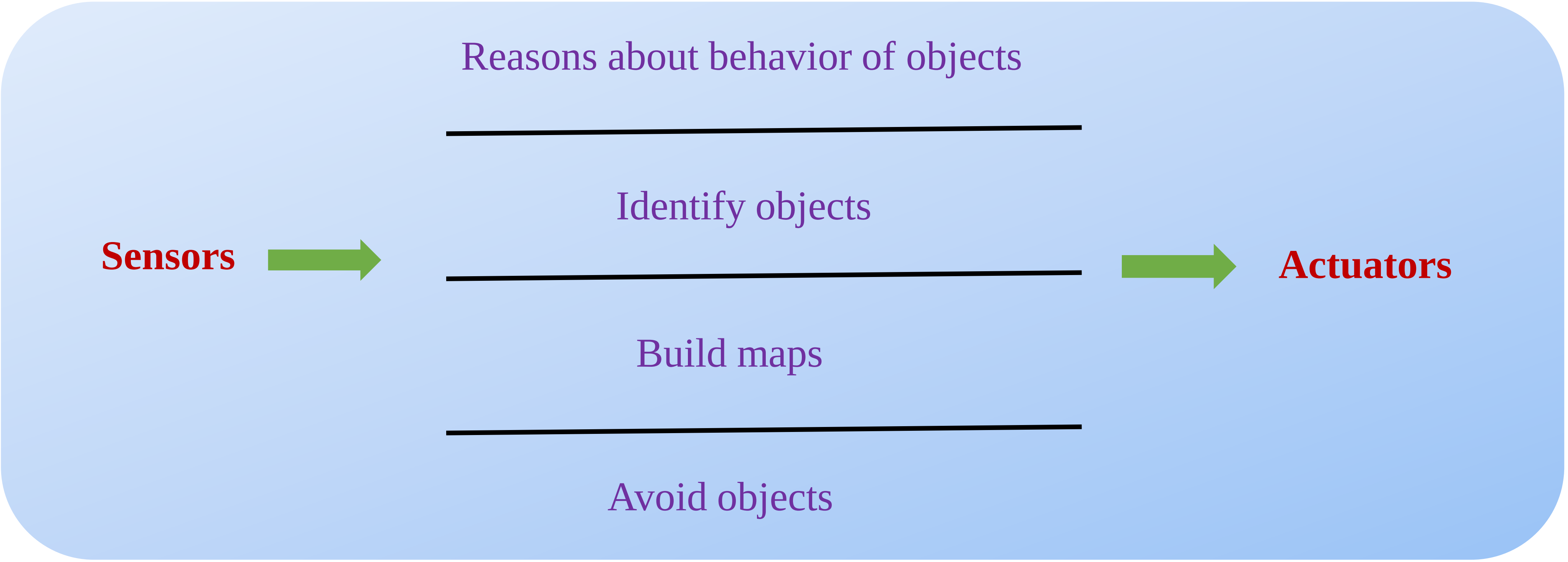
- Force-sensing
- Tactile-sensing
- Sonar
- Visual (camera)
- Proprioceptive



Robot with camera attached

Architecture

- Classical architecture
  -shortcomings
- Behavior-based architecture



Design for a behavior-based mobile robot

Current Developments

- Emotions
- Energy-efficiency
- Integration
  - Hierarchy of information representation
- Control structures
  - Synthesis of neural nets and fuzzy logic
- Robotic surgery
  - Telepresence
- Robot perception
  - Face and object recognition

Importance of Emotions

- Emotions help prevent people from repeating their mistakes (decisions that resulted in negative feelings)
- Recognizing emotions would allow robots to become more responsive to users' needs
- Exhibiting emotions would help robots interact with humans

Classification of Emotions

Continuous

- Emotions defined in multi-dimensional space of attributes
- Arousal-Valence Plane

Discrete

- Defines 5, 6, or more "basic" emotional states upon which more complex emotions are based

Future of Robotics

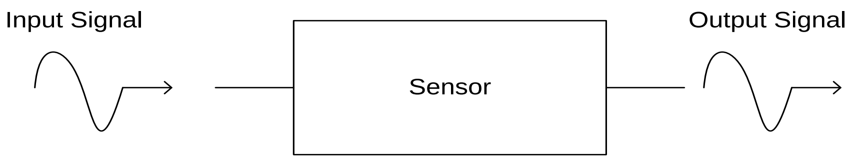
- **Downsizing**
  - Reduction in power needs and size
- **Synergism**
  - Greater integration of technologies
- **Greater intelligence**
  - More user-friendly interface
- **More environmentally friendly**
  - Robots easy to disassemble and destroy
  - Easily reusable or degradable parts

- Design robots to recognize presence, posture, and gaze
- Develop viable social exchange between robots and humans
- Design systems that can learn via reinforcement

19.2 Sensors and signal processing

What are Sensors?

- American National Standards Institute (ANSI) Definition
  - A device which provides a usable output in response to a specified measure and



- A sensor acquires a physical parameter and converts it into a signal suitable for processing (e.g. optical, electrical, mechanical)
- A transducer
  - Microphone, Loud Speaker, Biological Senses (e.g. touch, sight,…ect)

Detectable Phenomenon

Stimulus	Quantity
Acoustic	Wave (amplitude, phase, polarization), Spectrum, Wave Velocity
Biological & Chemical	Fluid Concentrations (Gas or Liquid)
Electric	Charge, Voltage, Current, Electric Field (amplitude, phase, polarization), Conductivity, Permittivity
Magnetic	Magnetic Field (amplitude, phase, polarization), Flux, Permeability
Optical	Refractive Index, Reflectivity, Absorption
Thermal	Temperature, Flux, Specific Heat, Thermal Conductivity
Mechanical	Position, Velocity, Acceleration, Force, Strain, Stress, Pressure, Torque

Physical Principles

- **Amperes's Law**
  - A current carrying conductor in a magnetic field experiences a force (e.g. galvanometer)

- **Curie-Weiss Law**
  - There is a transition temperature at which ferromagnetic materials exhibit paramagnetic behavior

- **Faraday's Law of Induction**
  - A coil resist a change in magnetic field by generating an opposing voltage/current (e.g. transformer)

- **Photoconductive Effect**
  - When light strikes certain semiconductor materials, the resistance of the material decreases (e.g. photoresistor)

Need for Sensors

- Sensors are omnipresent. They embedded in our bodies, automobiles, airplanes, cellular telephones, radios, chemical plants, industrial plants and countless other applications.

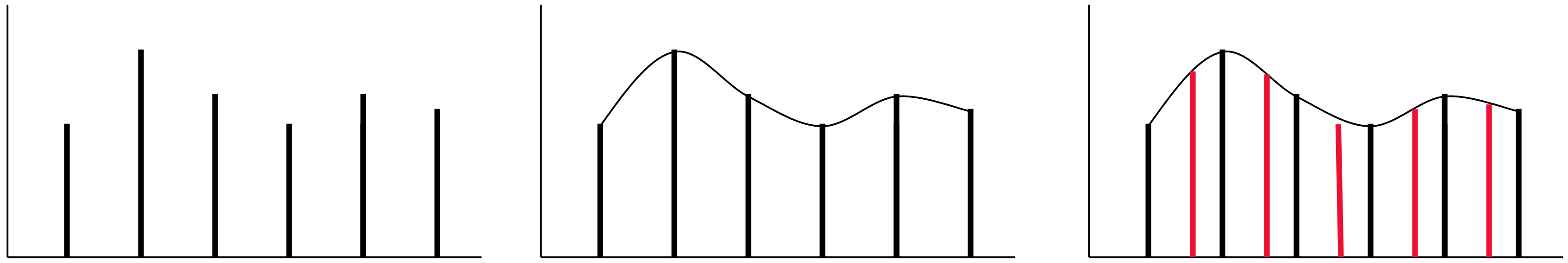
- Without the use of sensors, there would be no automation !!
  - Imagine having to manually fill Poland Spring bottles

Choosing a Sensor

Environmental Factors	Economic Factors	Sensor Characteristics
Temperature range	Cost	Sensitivity
Humidity effects	Availability	Range
Corrosion	Lifetime	Stability
Size		Repeatability
Overrange protection		Linearity
Susceptibility to EM interferences		Error
Ruggedness		Response time
Power consumption		Frequency response
Self-test capability		

Images as Samples of Functions

- We can view an image as a set of *samples* from an ideal function
  - This is particularly true if we are plotting a function, but it is also the case that a digital photograph is a sample of some function
- If we knew what the function was, we could enlarge the image by *resampling* the function
- Failing that, we can *reconstruct* the function and then resample it



Why Signal Processing?

- Signal processing is the analysis, interpretation, and manipulation of signals like sound, images time-varying measurement values and sensor data ete.
- Sampling:
  - How do we know how many pixels are required, and where to place them?
- Reconstruction:
  - What happens when a monitor converts a set of points into a continuous picture?
  - How do we enlarge, reduce, rotate, smooth or enhance images?
- Jaggies, Aliasing
  - How do we avoid jagged, pixelated lines

- For example biological datasuch as electrocardiograms, control system signals, telecommunication transmission signals such as radio signals, and many others.

19.3 Planning approaches to robot control

- The controller is the robot's brain and controls the robot's movements. It’s usually a computer of some type which is used to store information about the robot and the work environment and to store and execute programs which operate the robot.

Robot controllers

Programming languages in industrial robotics	
Robot brand	Language name
ABB	RAPID
Comau	PDL2
Fanuc	Karel
Kawasaki	AS
Kuka	KRL
Staubli	VAL3
Yaskawa	Inform

Types of robot control

- Point-to-point (PTP) control robot
- Continuous path control robot
- Controlled path robot

Point-to-point (PTP) control robot

- The PTP robot is capable of moving from one point to another point. The locations are recorded in the control memory.
- PTP robots do not control the path to get from one point to the next point.
- Common applications include:
  - ➢ Component insertion
  - ➢ Spot welding
  - ➢ Loading and unloading

Continuous path controlled robot

- The CP robot is capable of performing movements along the controlled path. With CP from one control, the robot can stop at any specified point along the controlled path.
- All the points along the path must be stored explicitly in the robot's control memory.
- Typical applications include:
  - ➢ Spray painting
  - ➢ Arc welding
  - ➢ Gluing

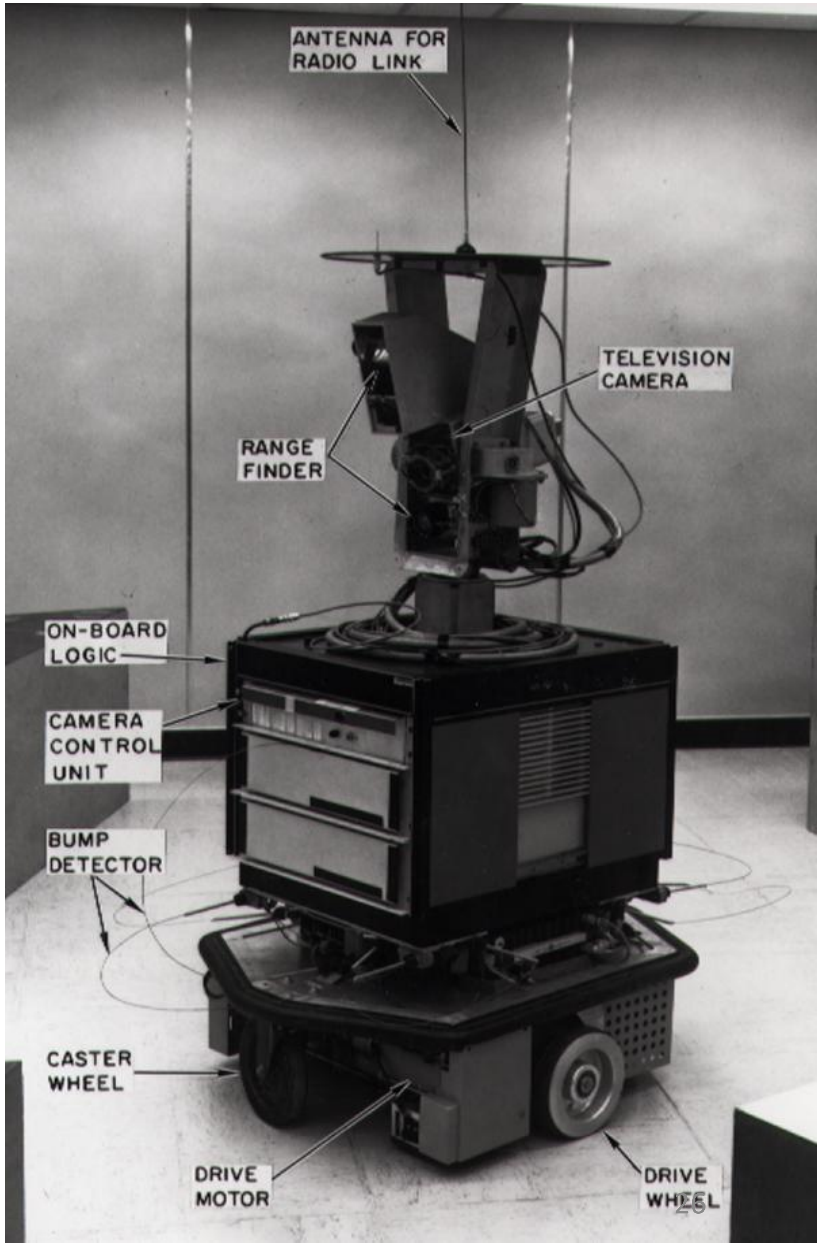
CONTROLLED PATH ROBOT

- In controlled-path robots, the control equipment can generate paths of different geometry such as straight lines, circles, and interpolated curves with a high degree of accuracy.
- Only the start and finish points and the path definition function must be stored in the robot's control memory.
- All controlled-path robots have a servo capability to correct their path.

19.3.1 STRIPS and SHAKEY

SHAKEY the First Intelligent Robot

- The first general-purpose mobile robot to be able to reason about its own actions.
- Due to its nature, the project combined research in robotics, computer vision, and natural language processing. Because of this, it was the first project that melded logical reasoning and physical action.
- Shakey was developed at the Artificial Intelligence Center of Stanford Research Institute (now called SRI International).
- Shakey has had a substantial influence on present-day AI and robotics.
- The Shakey project ran from 1966 to 1972 and was seen as a way of combining the various fields of artificial intelligence — logic, formulating plans, and executing plans — with computer vision and navigation.



19.3.2 Robot manipulator kinematics

- A Robot manipulator is designed to perform a task in the 3-D space.
- The tool or end effector is required to follow a planned trajectory to manipulate objects or carry out the task in the workspace.
- Kinematic model describes the spatial position of joints and links, position and orientation of end effector .
- The motion can be either unconstrained, if there is no physical interaction between the end- effector and the environment.
- The motion can be constrained if contact forces arise between the end effector and the environment.

Kinematic analysis:

- Kinematic analysis of a manipulator structure concerns the description of the manipulator motion with respect to a fixed reference.
- Cartesian frame by ignoring the forces and moments that cause the motion of the structure.
- Kinematics describes the analytical relationship between the joint positions and the end effector position and orientation.
- The formulation of the kinematics relationship allows studying two key problems of robotics.

1. The direct kinematics problem which is concerned with the determination of systematic, general method to describe the end- effector position and orientation as a function of the joint values by means of linear algebra tools.
2. The inverse kinematics problem which is concerned to transform a desired position and orientation of the end-effector in the workspace into the corresponding joint values.

Object Location

- Robotic manipulation, by definition, implies that parts and tools will be moved around in space by some sort of mechanism.
- This naturally leads to the need of representing positions and orientations i.e. the location of the parts, tools, and of the mechanism itself.
- To define and manipulate mathematical quantities which represent position and orientation we must define co-ordinate systems and develop conventions for representation.

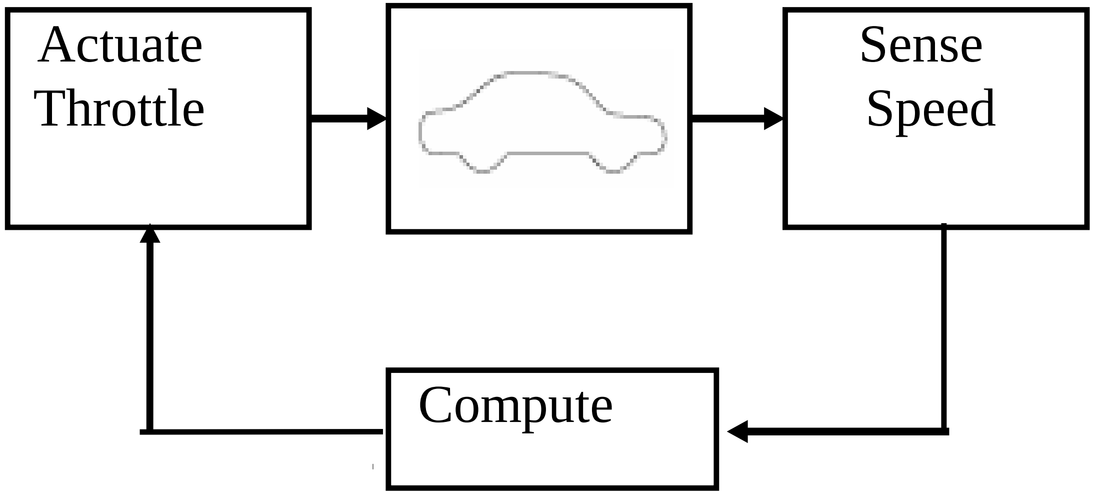
19.4 Control Theory

- Ensure stability

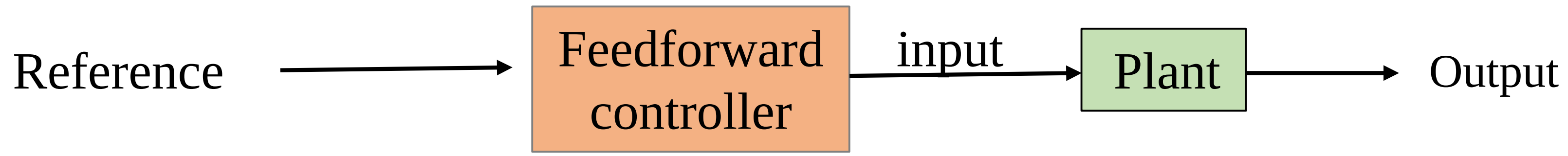
  - System maintains desired operating point (hold steady speed)
- Improve performance

  - System responds rapidly to changes (accelerate to 6 m/sec)
- Guarantee robustness

  - System tolerates perturbations in dynamics (mass, drag, etc)



19.4.1 Feedforward Control

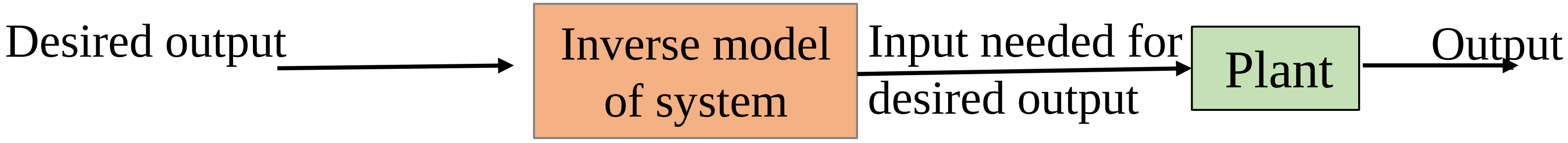


Control element responds to change in command or measured disturbance in a pre-defined way

Based on prediction of plant behavior (requires model) Can react before error actually occurs

- Overcome sluggish dynamics and delays
- Does not jeopardize stability

One implementation of feedforward



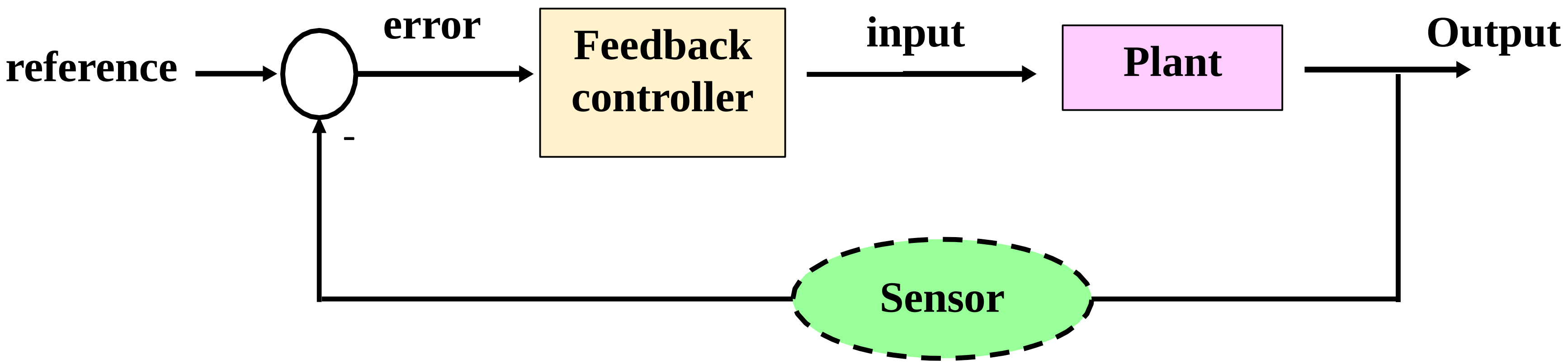
Model-based prediction of input

- Ideally consists of exact inverse model of the plant
- Can compensate for known plant dynamics, delays (before you get errors)
- No sensors needed
- System response must be predictable

Limitations of feedforward control

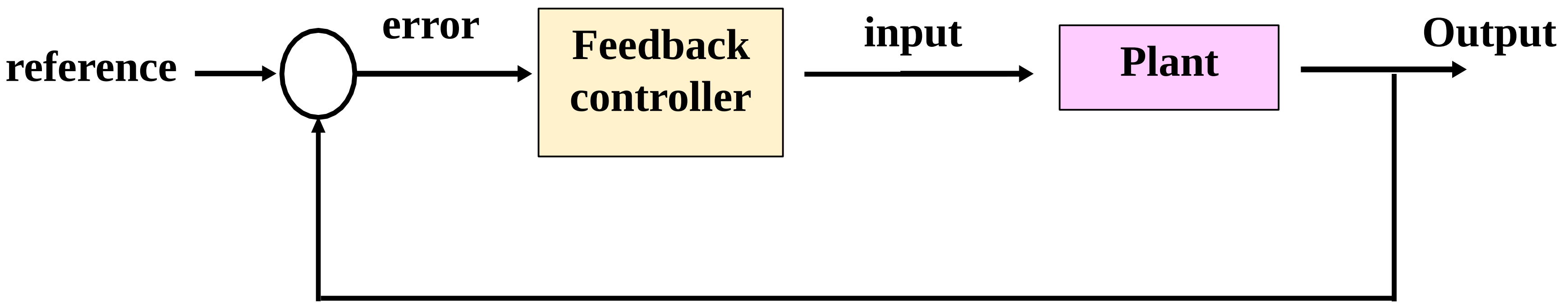
- Effects of disturbance or command input must be predictable
- May not generalize to other conditions
- Will not be accurate if the system changes
- The disturbance variables must be measured on-line. In many applications, this is not feasible.
- To make effective use of feedforward control, at least an approximate process model should be available.
- Ideal feedforward controllers that are theoretically capable of achieving perfect control may not be physically realizable. Fortunately, practical approximations of these ideal controllers often provide very effective control.

19.4.2 Feedback control



Plant	System to be controlled
Reference	Desired value of output (also ‘set point’)
Controller	Computes compensatory command to the plant based on error
Sensor	(implied)

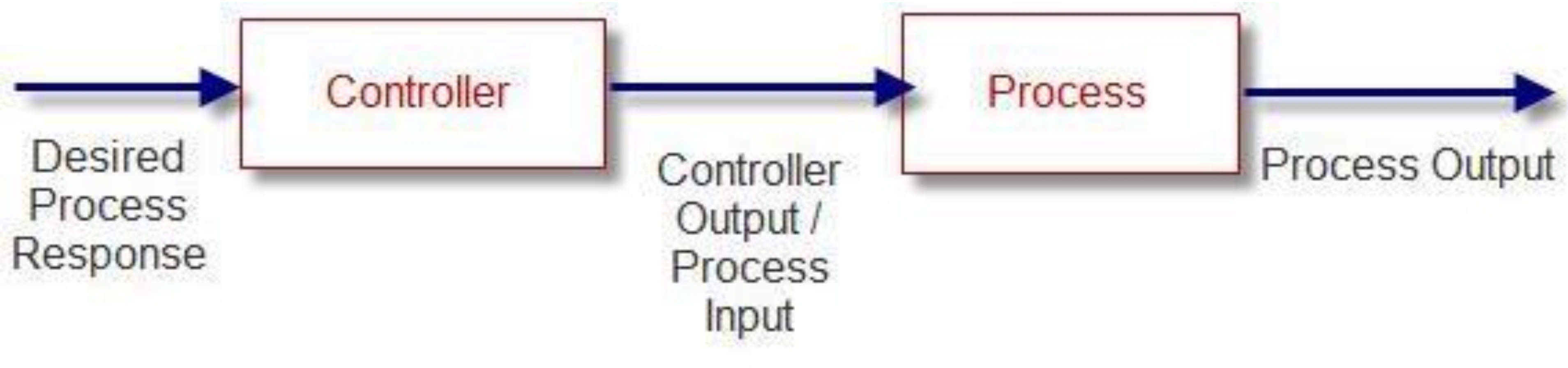
Features of feedback



- Reactive / Error-driven
- Automatically compensates for disturbances (controller acts on error)
- Automatically follows change in desired state (set point can change)
- Can improve undesirable properties of system/plant
- Can be very simple

Open loop control systems

- Any physical system which does not automatically correct for variation in its output, is called an open-loop system.
- Such a system may be represented by the block diagram as shown in Fig.



- In these systems, output is dependent on input but controlling action or input is totally independent of the output or changes in output of the system.
- In these systems the output remains constant for a constant input signal **provided the external conditions remain unaltered.**

Open loop control systems

Advantages: The advantages of open loop control system are,

- Such systems are simple in construction.
- Very much convenient when output is difficult to measure.
- Such systems are easy from maintenance point of view.
- Generally these are not troubled with the problems of stability.
- Such systems are simple to design and hence economical.

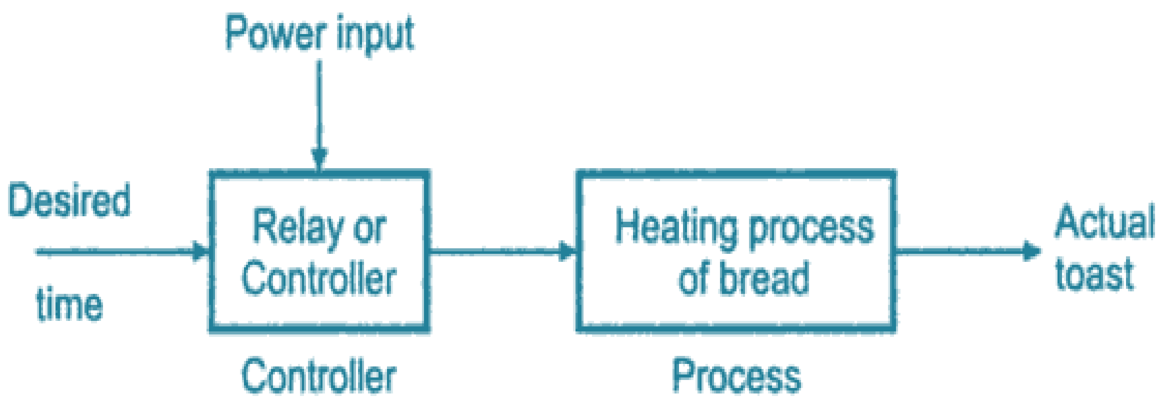
Disadvantages: The disadvantages of open loop control system are,

- These systems are inaccurate and unreliable because accuracy of such systems are totally dependent on the accurate pre-calibration of the controller.
- These systems give inaccurate results if there are variations in the external environment.
- Recalibration of the controller is necessary, time to time to maintain the quality and accuracy of the desired output.

Examples of open loop system

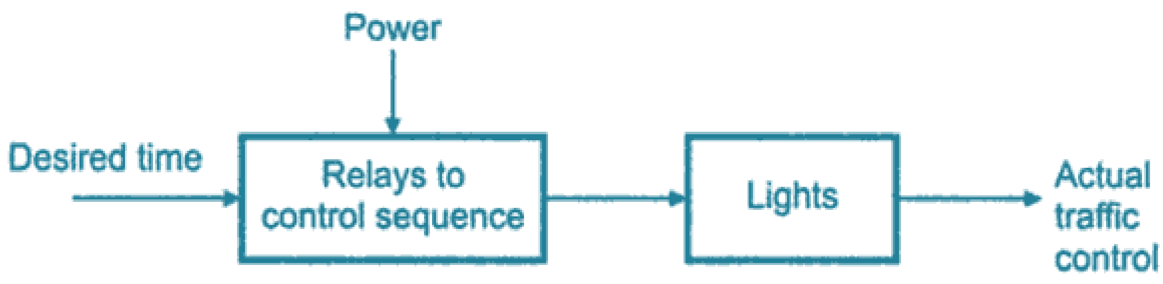
1) Automatic Toaster System

- In this system, the quality of toast depends upon the time for which the toast is heated.
- Depending on the time setting, bread is simply heated in this system.
- The toast quality is to be judged by the user and has no effect on the inputs.



2) Traffic Light Controller

- A traffic flow control system used on roads is time dependent.
- The traffic on the road becomes mobile or stationary depending on the duration and sequence of lamp glow.
- The sequence and duration are controlled by relays which are predetermined and not dependent on the rush on the road.



19.5 Probability Based Approaches

There are three ways to assign a probability , $P(O_i)$, to an outcome, O_i , namely:

Classical approach: based on equally likely events.

Example: If an experiment has n possible outcomes, this method would assign a probability of 1/n to each outcome. It is necessary to determine the number of possible outcomes.

- Experiment: Rolling a die
- Outcomes {1, 2, 3, 4, 5, 6}
- Probabilities: Each sample point has a 1/6 chance of occurring 6.3 .

Relative frequency: assigning probabilities based on experimentation or historical data.

Example: Bits & Bytes Computer Shop tracks the number of desktop computer systems it sells over a month (30 days):

Desktop Sold	# of Days
0	1
1	2
2	10
3	12
4	5

For example,
10 days out of 30
2 desktops were sold

➢ From this we can construct the probabilities of an event (i.e. the # of desktop sold on a given day)…

Desktops Sold	# of Days	Desktops Sold
0	1	1/30 = .03
1	2	2/30 07
2	10	10/30 = .33
3	12	12/30 = .40
4	5	5/30 = .17
		∑ = 100

“There is a 40% chance Bits & Bytes will sell 3 desktops on any given day”

19.5 Probability Based Approaches

Subjective approach: assigning probabilities based on the assignor's(subjective) judgement.
"In the subjective approach we define probability as the degree of belief that we hold in the occurrence of an event"

Example: weather forecasting's "P.O.P."
"Probability of Precipitation" (or P.O.P.) is defined in different ways by different forecasters, but basically it's a subjective probability based on past observations combined with current weather conditions. 6.7 POP 60% – based on current conditions, there is a 60% chance of rain (say).

Interpreting Probability…
No matter which method is used to assign probabilities all will be interpreted in the relative frequency approach.

For example, a government lottery game where 6 numbers (of 49) are picked. The classical approach would predict the probability for any one number being picked as 1/49=2.04%.
We interpret this to mean that in the long run each number 6.8 We interpret this to mean that in the long run each number will be picked 2.04% of the time.

- Also thought of as the “memoryless” property.
- A stochastic process is said to have the Markov property if the probability of state X_{n+1} having any given value depends only upon state X_n
- Very much depends on description of states.

Markov Property Example

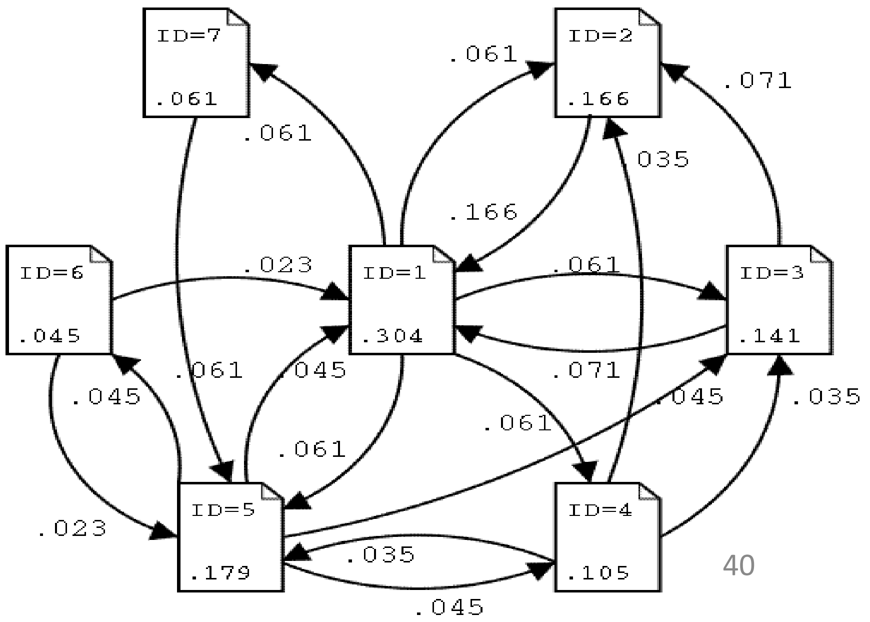
Checkers:

- Current State: The current configuration of the board
- Contains all information needed for transition to next state
- Thus, each configuration can be said to have the Markov property

Markov Chain

- Discrete-time stochastic process with the Markov property
- Industry Example: Google’s PageRank algorithm
- Probability distribution representing likelihood of random linking ending up on a page

$$\boxed{PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).}$$



19.5.2 Partially Observable Markov Decision Processes(POMDPs)

- Formal Specification and example

  - Belief State

  - Belief State Update

- Policies and Optimal Policy

  - Two Methods

- A special case of the Markov Decision Process(MDP). In an MDP, the environment is fully observable ,and with the Markov assumption for the transition model, the optimal policy depends only on the current state.

- For POMDPs, the environment is only partially observable.

POMDP Implications

- Since current state is not necessarily known, agent cannot execute the optimal policy for the state.

- A is defined by the following :
  - Set of states ***S***, set of actions ***A***, set of observations ***O***.
  - Transition model ***T(s, a, s')***
  - Reward model ***R(s)***
  - Observation model ***O(s, o)*** - probability of observing observation ***s*** in state ***o***

POMDP Solutions

- Solving POMDP on a physical state space is equivalent to solving an MDP on the believe state space.
- However state space is continuous and very high dimensional, so solutions are different to compute.
- Even finding approximately optimal solutions is PSPACE-hard.

Why study POMDPs ?

- In spite of the difficulties, POMDPs are still very important.
  - Many real-world problems and situations are not fully observable, but the Markov assumption is often valid.
- Active area of research
  - Google search on “POMDP” returns ≈ 5000 results
  - A number of current papers on the topic

Examples

- **Pattern Recognition tasks**

- SA-POMDP(single-action POMDP) – only decision is whether to change state or not.

- Model constructed to recognize words within text to which noise was added- i.e. individual letters within the words were

- SA-POMDP outperformed a pattern recognizer based on Hidden Markov Models, and exhibited better immunity to noise.

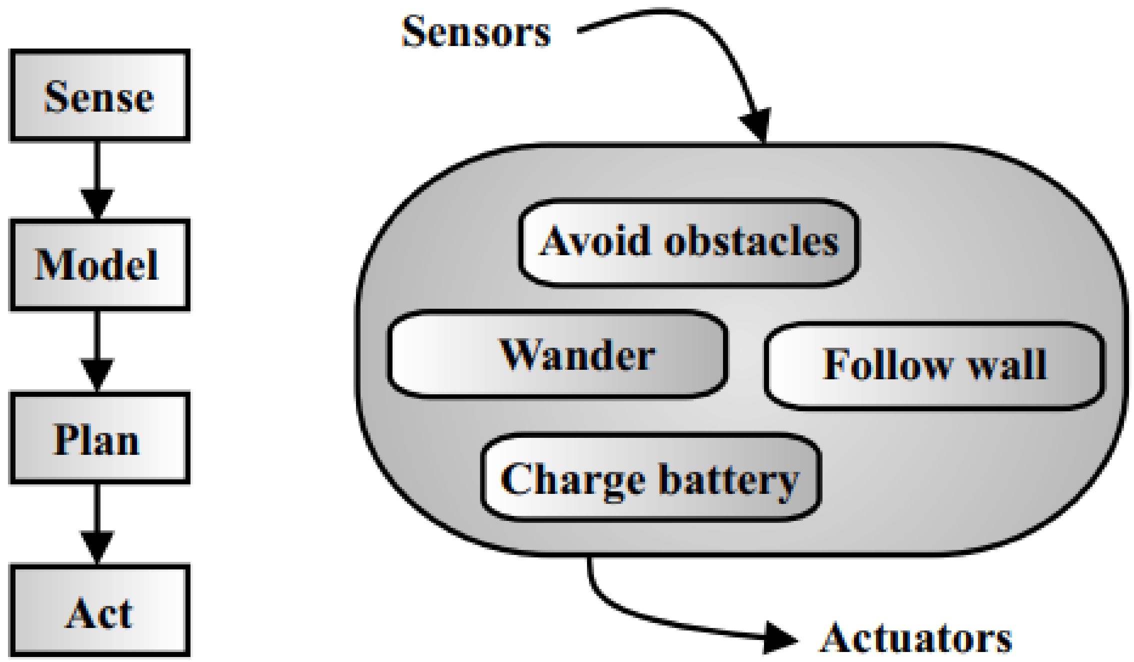
- **Robotics**

- Mission Planning

- Robot Navigation
  - POMDP used to control the movement of an autonomous robot within a crowded environment.
  - Used to predict the motion of other objects within the robot's environment.
  - Decompose state space into hierarchy, so individual POMDPs have a computationally tractable task.

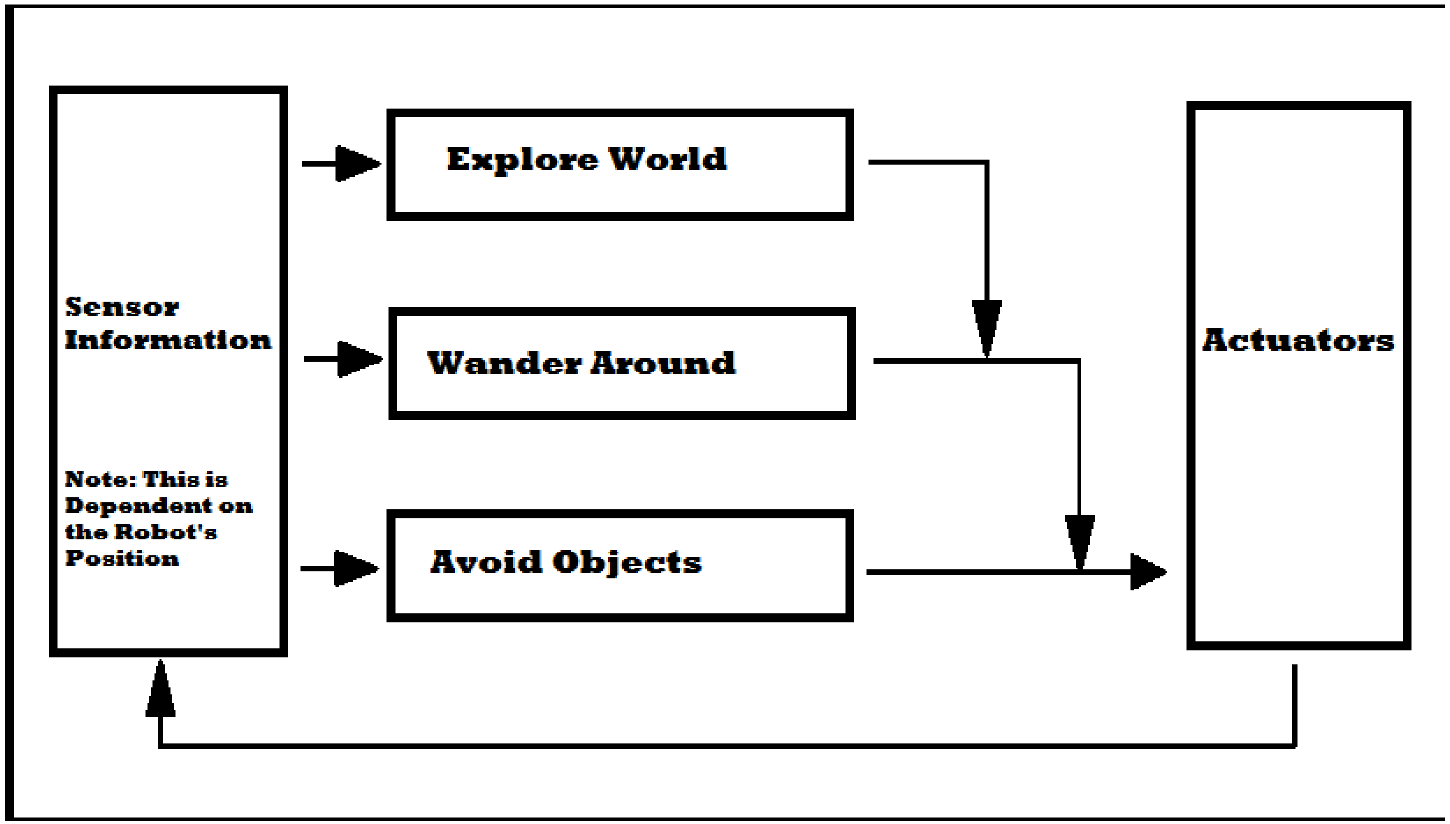
19.6 Behaviour-Based Control

Behavior-based robotics (**BBR**) or **behavioral robotics** is an approach in robotics that focuses on robots that are able to exhibit complex-appearing behaviors despite little internal variable state to model its immediate environment, mostly gradually correcting its actions via sensory-motor links.



19.6.1 The subsumption architecture

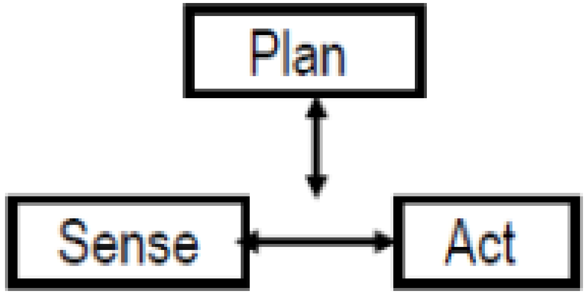
Subsumption architecture is a reactive robotic architecture heavily associated with behavior-based robotics which was very popular in the 1980s and 90s. The term was introduced by Rodney Brooks and colleagues in 1986.Subsumption has been widely influential in autonomous robotics and elsewhere in real-time AI.



19.6.2 Hybrid architectures

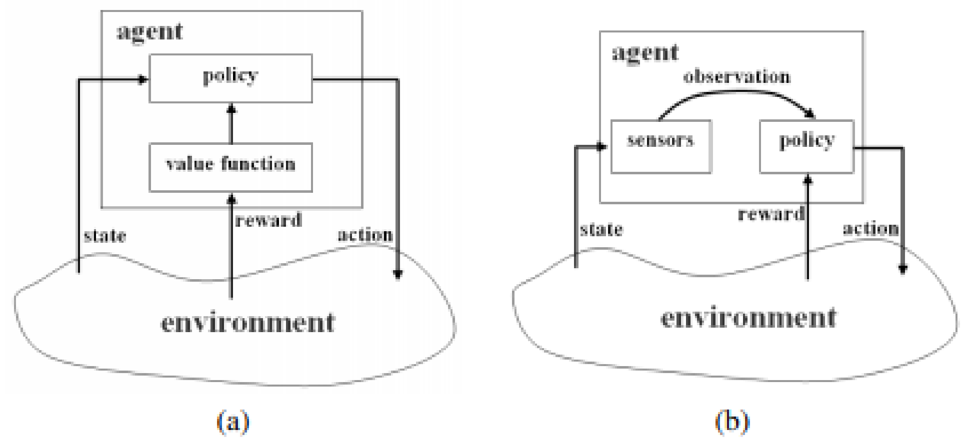
- Best general architecture solution because:
  - Use of asynchronous processing techniques (multi-tasking, threads, etc) allow deliberative functions to execute independently of reactive behaviors
- Provides responsiveness, robustness, and flexibility of purely reactive systems
  - Good software modularity allows subsystems or objects in Hybrid architectures to be mixed and matched for specific applications

Hybrid deliberative/reactive:



19.7.1 Reinforcement learning for control

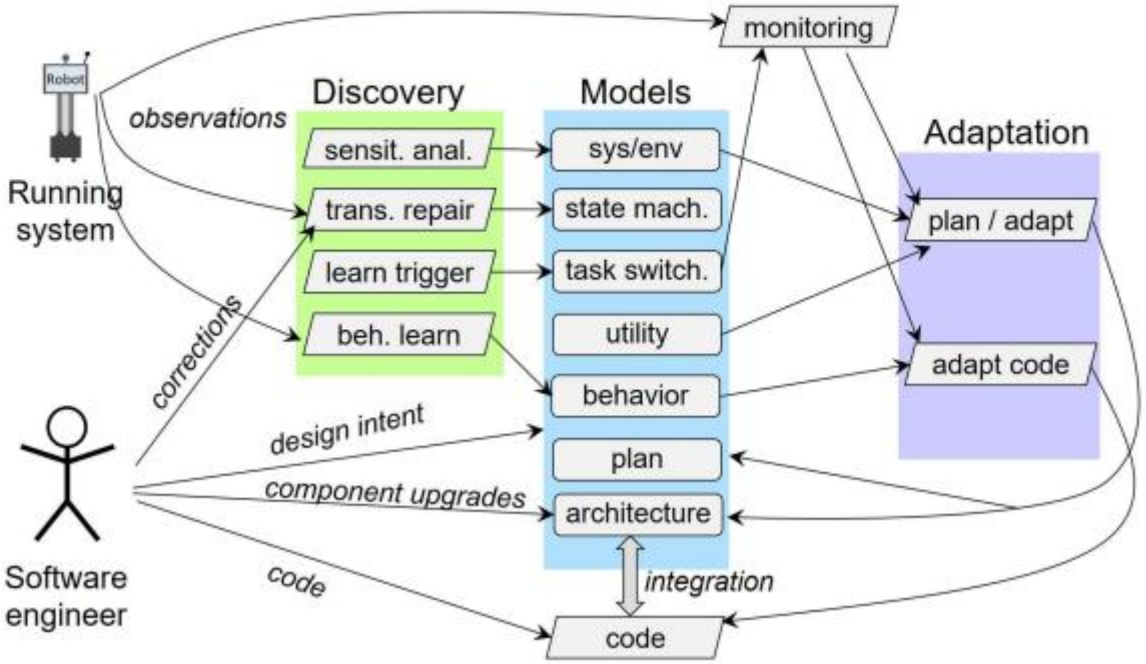
Consider the class of problems in which an agent, such as a robot, has to achieve some task by undertaking a series of actions in the environment, as shown in figure 1a. The agent perceives the state of the environment, selects an action to perform from its repertoire and executes it, thereby affecting the environment which transitions to a new state. The agent also receives a scalar signal indicating its level of performance, called the reward signal. The problem for the agent is to find a good policy for selecting actions given the states



The reinforcement learning framework. (a) In a fully observable world, the agent can estimate a value function for each state and use it to select its actions. (b) In a partially observable world, the agent does not know which state it is in due to sensor limitations; instead of a value function, the agent updates its policy parameters directly.

19.7.2 Model Based learning approaches to control

Intelligent ModelBased Adaptation (Figure 1), an approach in which developer-specified and automaticallydiscovered models are leveraged to autonomously adapt robotics software to a changing ecosystem. Our models capture the intent of the system and its components at a higher level of abstraction compared to source code. For example, a model of software architecture represents the high-level decomposition of the system into components and captures how those components communicate, providing a tool for reasoning about systemlevel adaptation. Other models represent the system and its environment, state machine control, triggers for replanning, utility, system behavior, and the current plan.



19.7.3 Learning maps

A **learning map** is a graphic organizer that highlights the knowledge, skills, and big ideas that students should get from a lesson, unit, or course. The **map** depicts the most important information to be learned and how the different pieces of **learning** are connected.

19.7.4 Evolutionary approaches

The first proposal that Darwinian selection could generate efficient control systems can be attributed to Alan Turing in the 1950s. He suggested that intelligent machines capable of adaptation and learning would be too difficult to conceive by a human designer and could instead be obtained by using an evolutionary process with mutations and selective reproduction .The development of computer algorithms inspired by the process of natural evolution followed shortly after ,but the first experiments on the evolution of adaptive behaviours for autonomous robots were done only in the early 1990s ,leading to the birth of the field of evolutionary robotics

19.8 Navigation and motion planning

Motion planning, also **path planning** (also known as the **navigation problem** or the **piano mover's problem**) is a computational problem to find a sequence of valid configurations that moves the object from the source to destination. The term is used in computational geometry, computer animation, robotics and computer games.

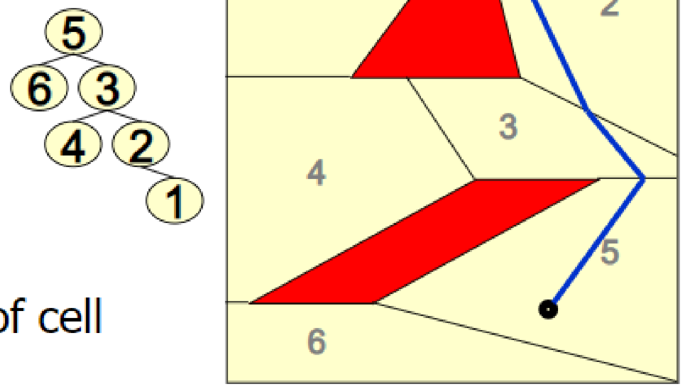
For example, consider navigating a mobile robot inside a building to a distant waypoint. It should execute this task while avoiding walls and not falling down stairs. A motion planning algorithm would take a description of these tasks as input, and produce the speed and turning commands sent to the robot's wheels. Motion planning algorithms might address robots with a larger number of joints (e.g., industrial manipulators), more complex tasks (e.g. manipulation of objects), different constraints (e.g., a car that can only drive forward), and uncertainty (e.g. imperfect models of the environment or robot).

19.8.1 Cell decomposition

Decompose the workspace into freespace and obstacle cells.

- Within freespace cells the robot can move freely
- Path planning reduces to:
- Finding (searching for) a sequence of adjacent freespace cells with the first containing the current location and the final one containing the goal
- Determining a strategy to move within and between freespace cells

- Cell Construction
  - Construct convex freespace cells
- Path Search
  - Label cells and search for path
- Path Construction
  - Connect centers of cell connections



19.8.2 Skeletonization

- Skeletonization Collapse the configuration space into a one- dimensional subset, or skeleton.
- Paths lie along the skeleton.
- Skeleton: A web with a finite number of vertices, and paths within the skeleton can be computed using graph search methods.
- Generally simpler than cell decomposition, because they provide a “minimal” description of free space.

Skeletonization : To be complete for motion planning, skeletonization methods must satisfy two properties:

1. If S is a skeleton of free space F, then S should have a connected piece within each connected region of F.
2. 2. For any point p in F, it should be “easy” to compute a path from p to the skeleton.
3. Skeletonization methods:

   1. Visibility graphs.
   2. Voronoi diagrams.
   3. Roadmaps.

19.8.3 Bounded-error planning

Bounded-error Planning (Fine-motion Planning) Planning small,precise motions for assembly. Sensor and actuator uncertainly.

Plan consists of a series of guarded motions.

1. Motion command.
2. 2. Termination condition.

Bounded-error Planning :

•Fine-motion planner takes as input the configuration space description, the angle of velocity uncertainty cone, and a specification of what sensing is possible for termination.

• Should produce a multi-step conditional plan or policy that is guaranteed to succeed, if such a plan exists.

•Plans are designed for the worst case outcome.

•Extremely high complexity.

19.8.4 Landmark- based navigation

Landmark Based Navigation

• Assume the environment contains easily recognizable, unique landmarks.

•A landmark is surrounded with a circular field of influence.

•Robot's control is assumed to be imperfect.

• The environment is know at planning time, but not the robot's position.

•Plan backwards from the goal using backprojection.

• Polynomial complexity.

19.8.5 Online algorithm

•Online Algorithms

• Environment is poorly known.

• Produce conditional plan.

• Need to be simple

•. Very fast and complete, but almost always give up any guarantee of finding the shortest path.

• Competitive ratio.

References

- https://www.clear.rice.edu/engi128/Handouts/Lec10-Control.pdf
- file:///C:/Users/ACDSD/Downloads/unit-1-introductiontocontrolsystems-170821111427.pdf
- https://www.slideserve.com/haroun/what-are-partially-observable-markov-decision-processes
- http://cs.brown.edu/research/ai/pomdp/tutorial/index.html

Thank you