ARTIFICIAL INTELLIGENCE

UNIT-V

SYLLABUS:

Robotics: Introduction, Robot Hardware, Robotic Perception, Planning to move, planning uncertain movements, Moving, Robotic software architectures, application domains.

Philosophical foundations: Weak AI, Strong AI, Ethics and Risks of AI, Agent Components, Agent Architectures, Are we going in the right direction, What if AI does succeed.

AI Unit-5.3: ROBOT:

Robots are physical agents that perform tasks by manipulating the physical world.

Effectors have a single purpose that to assert physical forces on the environment. Robots are also equipped with **sensors**, which allow them to perceive their environment.

Most of today's robots fall into one of three primary categories.

1.MANIPULATORS:

Manipulator motion usually involves a chain of controllable joints, enabling such robots to place their effectors in any position within the workplace. Few car manufacturers could survive without robotic manipulators, and some manipulators have even been used to generate original artwork.

2.MOBBILE ROBOT:

The second category is the **mobile robot**. Mobile robots move about their environment using wheels, legs, or similar mechanisms. They have been put to use delivering food in hospitals, moving containers at loading docks, and similar tasks. Other types of mobile robots include **unmanned air vehicles, Autonomous underwater vehicles etc..,**

3.MOBILE MANIPULATOR:

The third type of robot combines mobility with manipulation, and is often called a **mobile manipulator**. **Humanoid robots** mimic the human torso.

The field of robotics also includes prosthetic devices , intelligent environments and multibody systems, wherein robotic action is achieved through swarms of small cooperating robots. Robotics brings together many of the concepts we have seen earlier in the book, including probabilistic state estimation, perception, planning, unsupervised learning, and reinforcement learning.

ROBOT HARDWARE:

The robot hardware mainly depends on 1.sensors and 2.effectors

1.sensors:

Sensors are the perceptual interface between robot and environment.

PASSIVE SENSOR: Passive sensors, such as cameras, are true observers of the environment: they capture signals that are generated by other sources in the environment.

ACTIVE SENSOR: Active sensors, such as sonar, send energy into the environment. They rely on the fact that this energy is reflected back to the sensor.

Range finders are sensors that measure the distance to nearby objects. In the early days of robotics, robots were commonly equipped with **sonar sensors**. Sonar sensors emit directional sound waves, which are reflected by objects, with some of the sound making it back into the sensor.

Stereo vision relies on multiple cameras to image the environment from slightly different viewpoints, analyzing the resulting parallax in these images to compute the range of surrounding objects.

Other common range sensors include radar, which is often the sensor of choice for UAVs. Radar sensors can measure distances of multiple kilometers. On the other extreme end of range sensing are **tactile sensors** such as whiskers, bump panels, and touch-sensitive skin.

A second important class of sensors is **location sensors**. Most location sensors use range sensing as a primary component to determine location. Outdoors, the **Global Positioning System** (GPS) is the most common solution to the localization problem. GPS measures the distance to satellites that emit pulsed signals.

The third important class is **proprioceptive sensors**, which inform the robot of its own motion. To measure the exact configuration of a robotic joint, motors are often equipped with **shaft decoders** that count the revolution of motors in small increments.

Other important aspects of robot state are measured by **force sensors** and **torque sensors**. These are indispensable when robots handle fragile objects or objects whose exact shape and location is unknown.

EFFECTORS:

Effectors are the means by which robots move and change the shape of their bodies. To understand the design of effectors we use the concept of degree of freedom.

We count one degree of freedom for each independent direction in which a robot, or one of its effectors, can move. For example, a rigid mobile robot such as an AUV has six degrees of freedom, three for its (x, y, z) location in space and three for its angular orientation, known as *yaw*, *roll*, and *pitch*. These six degrees define the **kinematic state**2 or **pose** of the robot. The **dynamic state** of a robot includes these six plus an additional six dimensions for the rate of change of each kinematic dimension, that is, their velocities.

For nonrigid bodies, there are additional degrees of freedom within the robot itself. For example, the elbow of a human arm possesses two degree of freedom. It can flex the upper arm towards or away, and can rotate right or left. The wrist has three degrees of freedom. It can move up and down, side to side, and can also rotate. Robot joints also have one, two, or three degrees of freedom each. Six degrees of freedom are required to place an object, such as a hand, at a particular point in a particular orientation.

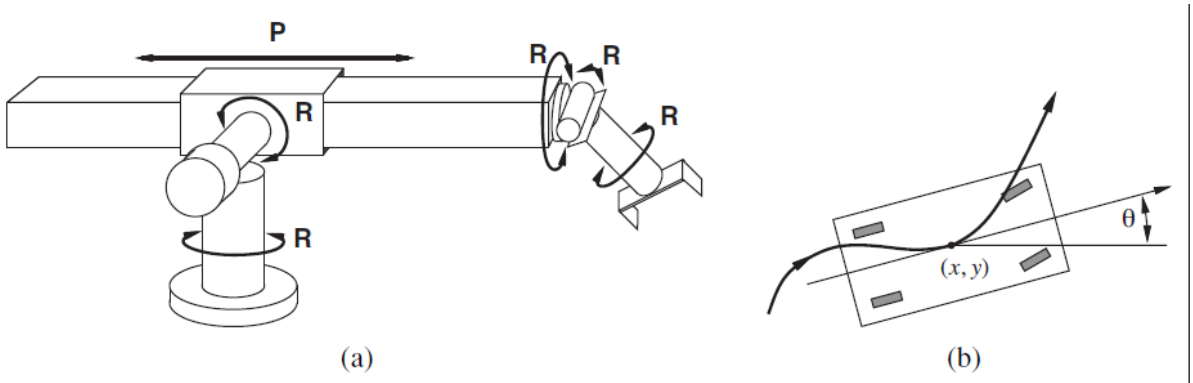


Figure 25.4 (a) The Stanford Manipulator, an early robot arm with five revolute joints (R) and one prismatic joint (P), for a total of six degrees of freedom. (b) Motion of a nonholonomic four-wheeled vehicle with front-wheel steering.

In the fig 4(a) has exactly six degrees of freedom, created REVOLUTE JOINT by five **revolute joints** that generate rotational motion and one **prismatic joint** that generates sliding motion

For mobile robots, the DOFs are not necessarily the same as the number of actuated elements.
Consider, for example, your average car: it can move forward or backward, and it can turn, giving it two DOFs. In contrast, a car's kinematic configuration is three-dimensional: on an open flat surface, one can easily maneuver a car to any (x, y) point, in any orientation. (See Figure 25.4(b).) Thus, the car has three **effective degrees of freedom** but two **control label degrees of freedom**. We say a robot is **nonholonomic** if it has more effective DOFs than controllable DOFs and **holonomic** if the two numbers are the same.

Sensors and effectors alone do not make a robot. A complete robot also needs a source of power to drive its effectors. The **electric motor** is the most popular mechanism for both manipulator actuation and locomotion, but **pneumatic actuation** using compressed gas and **Hydraulic actuation** using pressurized fluids also have their application niches.

ROBOTIC PERCEPTION:

Perception is the process by which robots map sensor measurements into internal representations of the environment. Perception is difficult because sensors are noisy, and the environment is partially observable, unpredictable, and often dynamic.

As a rule of thumb, good internal representations for robots have three properties: they contain enough information for the robot to make good decisions, they are structured so that they can be updated efficiently, and they are natural in the sense that internal variables correspond to natural state variables in the physical world.

For robotics problems, we include the robot's own past actions as observed variables in the model. Figure 25.7 shows the notation used in this

chapter: $\mathbf{X}t$ is the state of the environment (including the robot) at time t, $\mathbf{Z}t$ is the observation received at time t, and At is the action taken after the observation is received.

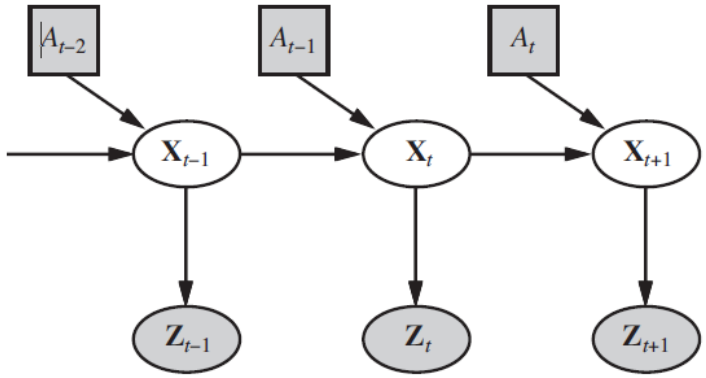


Figure 25.7 Robot perception can be viewed as temporal inference from sequences of actions and measurements, as illustrated by this dynamic Bayes network.

We would like to compute the new belief state, $\mathbf{P}(\mathbf{X}t+1 \mid \mathbf{z}1{:}t+1, a1{:}t)$, from the current belief state $\mathbf{P}(\mathbf{X}t \mid \mathbf{z}1{:}t, a1{:}t-1)$ and the new observation $\mathbf{z}t+1$.
Thus, we modify the recursive filtering equation (15.5 on page 572) to use integration rather than summation:

$$\mathbf{P}(\mathbf{X}t+1 \mid \mathbf{z}1{:}t+1, a1{:}t)$$
$$= \alpha\, \mathbf{P}(\mathbf{z}t+1 \mid \mathbf{X}t+1) \int \mathbf{P}(\mathbf{X}t+1 \mid xt, at)\, P(\mathbf{x}t \mid \mathbf{z}1{:}t, a1{:}t-1)\, d\mathbf{x}t \quad (25.1)$$

This equation states that the posterior over the state variables $\mathbf{X}$ at time $t+1$ is calculated recursively from the corresponding estimate one time step earlier. This calculation involves the previous action at and the current sensor measurement $\mathbf{z}t+1$. The probability $\mathbf{P}(\mathbf{X}t+1 \mid \mathbf{x}t, at)$ is called the **transition model** or **motion model**, and $\mathbf{P}(\mathbf{z}t+1 \mid \mathbf{X}\,t+1)$ is the **sensor model**.

1.Localization and mapping

Localization is the problem of finding out where things are—including the robot itself.

Knowledge about where things are is at the core of any successful physical interaction with the environment.
To keep things simple, let us consider a mobile robot that moves slowly in a flat 2D world. Let us also assume the robot is given an exact map of the environment. The pose of such a mobile robot is defined by its two Cartesian coordinates with values x and y and its heading with value θ, as illustrated in Figure 25.8(a). If we arrange those three values in a vector, then any particular state is given by $\mathbf{X}t = (xt, yt, \theta\, t)^T$.

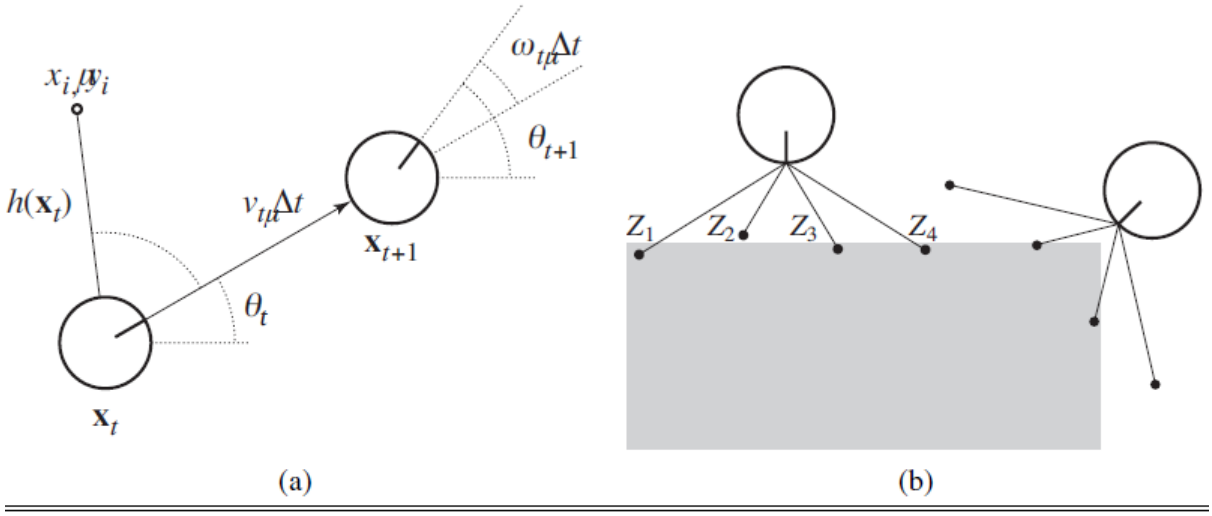


Figure 25.8 (a) A simplified kinematic model of a mobile robot. The robot is shown as a circle with an interior line marking the forward direction. The state $\mathbf{x}_t$ consists of the (x_t, y_t) position (shown implicitly) and the orientation θ_t. The new state $\mathbf{x}_{t+1}$ is obtained by an update in position of $v_t\Delta_t$ and in orientation of $\omega_t\Delta_t$. Also shown is a landmark at (x_i, y_i) observed at time t. (b) The range-scan sensor model. Two possible robot poses are shown for a given range scan (z_1, z_2, z_3, z_4). It is much more likely that the pose on the left generated the range scan than the pose on the right.

In the kinematic approximation, each action consists of the "instantaneous" specification of two velocities—a translational velocity v_t and a rotational velocity ω_t. For small time intervals Δt, a crude deterministic model of the motion of such robots is given by

$$\hat{\mathbf{X}}_{t+1} = f(\mathbf{X}_t, \underbrace{v_t, \omega_t}_{a_t}) = \mathbf{X}_t + \begin{pmatrix} v_t\Delta t \cos\theta_t \\ v_t\Delta t \sin\theta_t \\ \omega_t\Delta t \end{pmatrix} .$$

The notation $\hat{\mathbf{X}}$ refers to a deterministic state prediction. Of course, physical robots are somewhat unpredictable. This is commonly modeled by a Gaussian distribution with mean $f(\mathbf{X}_t, v_t, \omega_t)$ and covariance $\mathbf{\Sigma}_x$. (See Appendix A for a mathematical definition.)

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{X}_t, v_t, \omega_t) = N(\hat{\mathbf{X}}_{t+1}, \mathbf{\Sigma}_x) .$$

Next, we need a sensor model. We will consider two kinds of sensor model. The first assumes that the sensors detect *stable, recognizable* features of the environment called **landmarks**. The exact prediction of the observed range and bearing would be

$$\hat{\mathbf{z}}_t = h(\mathbf{x}_t) = \begin{pmatrix} \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2} \\ \arctan\frac{y_i - y_t}{x_i - x_t} - \theta_t \end{pmatrix}$$

Again, noise distorts our measurements. To keep things simple, one might assume Gaussian noise with covariance $\mathbf{\Sigma}_z$, giving us the sensor model

$$P(\mathbf{z}t \mid \mathbf{x}t) = N(\hat{}\mathbf{z}t, \mathbf{\Sigma}z) .$$

function MONTE-CARLO-LOCALIZATION(a, z, N, $P(X'|X,\ v,\ \omega)$, $P(z|z^*)$, m) **returns** a set of samples for the next time step

inputs: a, robot velocities v and ω
z, range scan $z_1, \ldots, z_M$
$P(X'|X,\ v,\ \omega)$, motion model
$P(z|z^*)$, range sensor noise model
m, 2D map of the environment

persistent: S, a vector of samples of size N

local variables: W, a vector of weights of size N
S', a temporary vector of particles of size N
W', a vector of weights of size N

if S is empty **then** /* initialization phase */
  for $i = 1$ to N **do**
    $S[i] \leftarrow$ sample from $P(X_0)$
  for $i = 1$ to N **do** /* update cycle */
    $S'[i] \leftarrow$ sample from $P(X'|X = S[i], v, \omega)$
    $W'[i] \leftarrow 1$
    for $j = 1$ to M **do**
      $z^* \leftarrow$ RAYCAST(j, $X = S'[i]$, m)
      $W'[i] \leftarrow W'[i] \cdot P(z_j | z^*)$
  $S \leftarrow$ WEIGHTED-SAMPLE-WITH-REPLACEMENT(N,S',W')
return S

Figure 25.9 A Monte Carlo localization algorithm using a range-scan sensor model with independent noise.

This problem is important for many robot applications, and it has been studied extensively under the name **simultaneous localization and mapping**, abbreviated as **SLAM**.

SLAM problems are solved using many different probabilistic techniques, including the extended Kalman filter

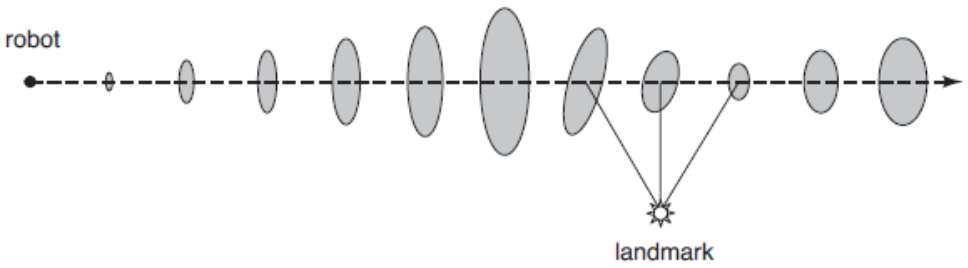


Figure 25.12 Example of localization using the extended Kalman filter. The robot moves on a straight line. As it progresses, its uncertainty increases gradually, as illustrated by the error ellipses. When it observes a landmark with known position, the uncertainty is reduced.

Expectation–maximization is also used for SLAM.

2.Other types of perception

Not all of robot perception is about localization or mapping. Robots also perceive the temperature, odors, acoustic signals, and so on. Many of these quantities can be estimated using variants of dynamic Bayes networks.

It is also possible to program a robot as a reactive agent, without explicitly reasoning about probability distributions over states.

3.Machine learning in robot perception

Machine learning plays an important role in robot perception. This is particularly the case when the best internal representation is not known. One common approach is to map high dimensional sensor streams into lower-dimensional spaces using unsupervised machine learning method. Such an approach is called **low-dimensional embedding**.

Methods that make robots collect their own training data are called **Self Supervised**.

In this instance, the robot uses machine learning to leverage a short-range sensor that works well for terrain classification into a sensor that can see much farther. That allows the robot to drive faster, slowing down only when the sensor model says there is a change in the terrain that needs to be examined more carefully by the short-range sensors.

PLANNING TO MOVE:

All of a robot's deliberations ultimately come down to deciding how to move effectors. The **point-to-point motion** problem is to deliver the robot or its end effector to a designated target location. A greater challenge is the **compliant motion** problem, in which a robot moves while being in physical contact with an obstacle.

There are two main approaches: **cell decomposition** and **skeletonization**. Each reduces the continuous path-planning problem to a discrete graph-search problem.

1 Configuration space

We will start with a simple representation for a simple robot motion problem. It has two joints that move independently. the robot's configuration can be described by a four dimensional coordinate: (x_e, y_e) for the location of the elbow relative to the environment and (x_g, y_g) for the location of the gripper. They constitute what is known as **workspace representation**.

The problem with the workspace representation is that not all workspace coordinates are actually attainable, even in the absence of obstacles. This is because of the **linkage constraints** on the space of attainable workspace coordinates.

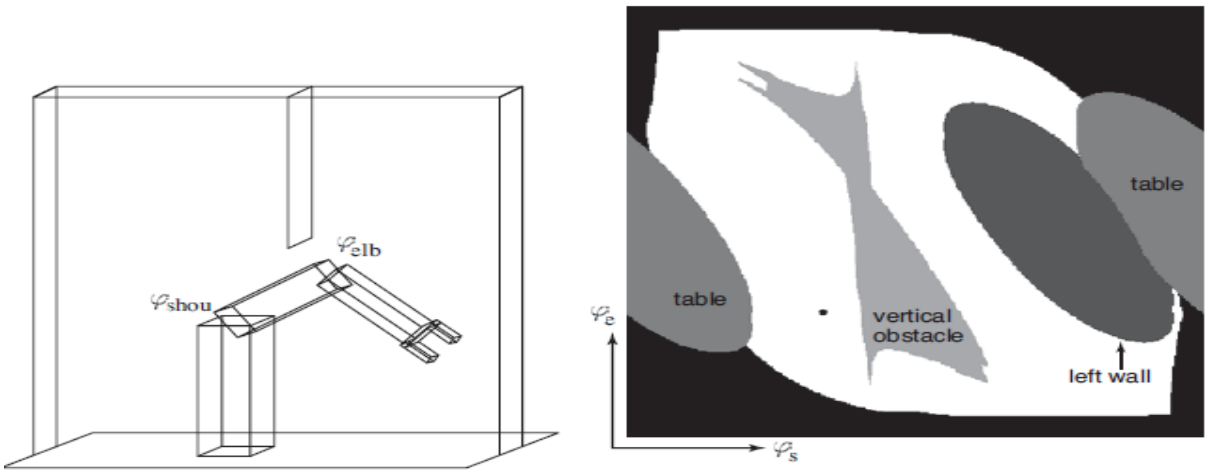


Figure 25.14 (a) Workspace representation of a robot arm with 2 DOFs. The workspace is a box with a flat obstacle hanging from the ceiling. (b) Configuration space of the same robot. Only white regions in the space are configurations that are free of collisions. The dot in this diagram corresponds to the configuration of the robot shown on the left.

Transforming configuration space coordinates into workspace coordinates is simple: it involves a series of straightforward coordinate transformations. These transformations are linear for prismatic joints and trigonometric for revolute joints. This chain of coordinate transformation is known as **kinematics**.

The inverse problem of calculating the configuration of a robot whose effector location is specified in workspace coordinates is known as **inverse kinematics.**

2 Cell decomposition methods

The first approach to path planning uses **cell decomposition**—that is, it decomposes the free space into a finite number of contiguous regions, called cells.

A decomposition has the advantage that it is extremely simple to implement, but it also suffers from three limitations. First, it is workable only for low-dimensional configuration spaces, Second, there is the problem of what to do with cells that are "mixed", And third, any path through a discretized state space will not be smooth.

Cell decomposition methods can be improved in a number of ways, to alleviate some of these problems. The first approach allows *further subdivision* of the mixed cells—perhaps using cells of half the original size. A second way to obtain a complete algorithm is to insist on an **exact cell decomposition** of the free space.

3 Modified cost functions:

This problem can be solved by introducing a **potential field**. A potential field is a function defined over state space, whose value grows with the distance to the closest obstacle. The potential field can be used as an additional cost term in the shortest-path calculation. This induces an interesting trade off. On the one hand, the robot seeks to minimize path length to the goal. On the other hand, it

tries to stay away from obstacles by virtue of minimizing the potential function. Clearly, the resulting path is longer, but it is also safer.

There exist many other ways to modify the cost function. However, it is often easy to smooth the resulting trajectory after planning, using conjugate gradient methods. Such post-planning smoothing is essential in many real world applications.

4 Skeletonization methods

The second major family of path-planning algorithms is based on the idea of **skeletonization**.

These algorithms reduce the robot's free space to a one-dimensional representation, for which the planning problem is easier. This lower-dimensional representation is called a **skeleton** of the configuration space.

Voronoi graph of the free space—the set of all points that are equidistant to two or more obstacles. To do path planning with a Voronoi graph, the robot first changes its present configuration to a point on the Voronoi graph. It is easy to show that this can always be achieved by a straight-line motion in configuration space. Second, the robot follows the Voronoi graph until it reaches the point nearest to the target configuration. Finally, the robot leaves the Voronoi graph and moves to the target. Again, this final step involves straight-line motion in configuration space.

An alternative to the Voronoi graphs is the **probabilistic roadmap**, a skeletonization approach that offers more possible routes, and thus deals better with wide-open spaces. With these improvements, probabilistic roadmap planning tends to scale better to high-dimensional configuration spaces than most alternative path-planning techniques.

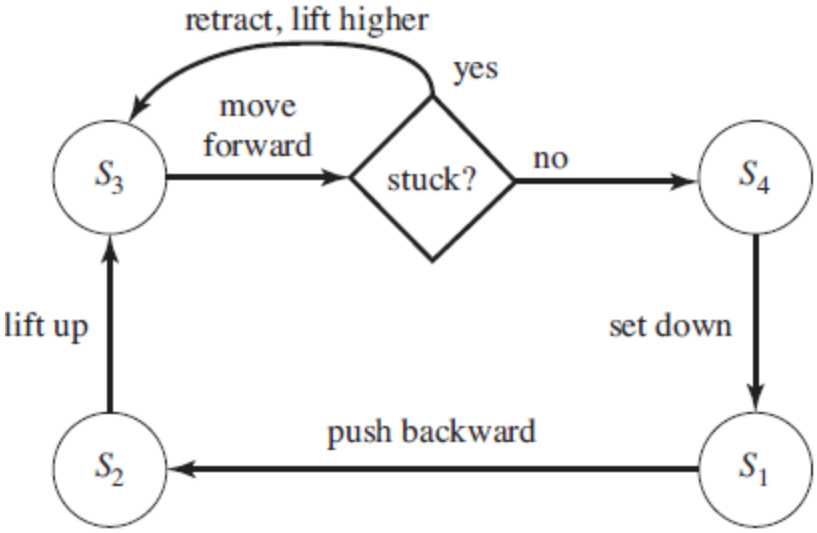
ROBOTIC SOFTWARE ARCHITECTURE:

A methodology for structuring algorithms is called a **software architecture**. An architecture includes languages and tools for writing programs, as well as an overall philosophy for how programs can be brought together. Architectures that combine reactive and deliberate techniques are called **hybrid architectures**.

1 Subsumption architecture

The **subsumption architecture** is a framework for assembling reactive controllers out of finite state machines. Nodes in these machines may contain tests for certain sensor variables, in which case the execution trace of a finite state machine is conditioned on the outcome of such a test. The resulting machines are refereed to as **augmented finite state machines**, or AFSMs, where the augmentation refers to the use of clocks.

An example of a simple AFSM is the four-state machine shown in BELOW Figure, which generates cyclic leg motion for a hexapod walker.



(b)

In our example, we might begin with AFSMs for individual legs, followed by an AFSM for coordinating multiple legs. On top of this, we might implement higher-level behaviors such as collision avoidance, which might involve backing up and turning.

Unfortunately, the subsumption architecture has its own problems. First, the AFSMs are driven by raw sensor input, an arrangement that works if the sensor data is reliable and contains all necessary information for decision making, but fails if sensor data has to be integrated in nontrivial ways over time. A subsumption style robot usually does just one task, and it has no notion of how to modify its controls to accommodate different goals. Finally, subsumption style controllers tend to be difficult to understand.

However, it has had an influence on other architectures, and on individual components of some architectures.

2 Three-layer architecture

Hybrid architectures combine reaction with deliberation. The most popular hybrid architecture is the **three-layer architecture**, which consists of a reactive layer, an executive layer, and a deliberative layer.

The **reactive layer** provides low-level control to the robot. It is characterized by a tight sensor–action loop. Its decision cycle is often on the order of milliseconds.

The **executive layer** (or sequencing layer) serves as the glue between the reactive layer and the deliberative layer. It accepts directives by the deliberative layer, and sequences them for the reactive layer.

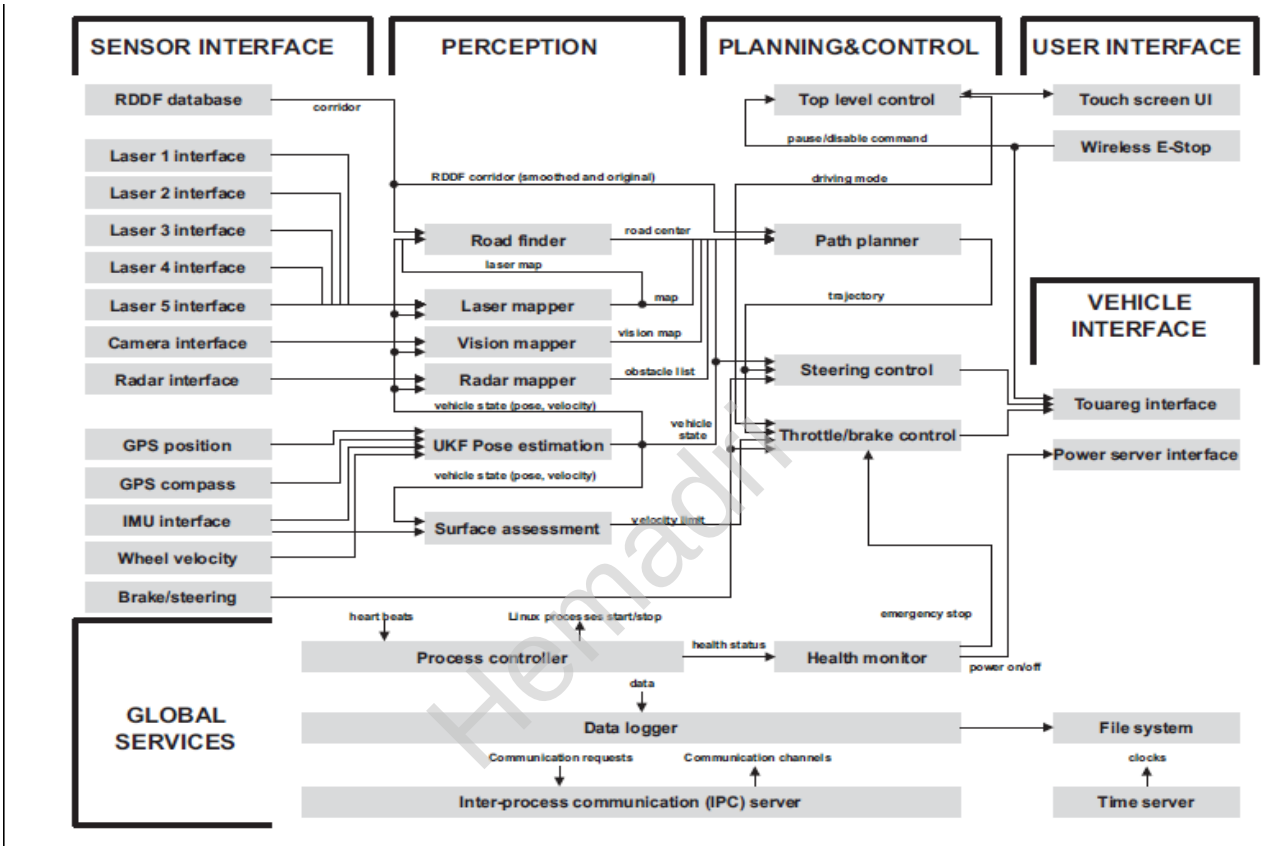
The **deliberative layer** generates global solutions to complex tasks using planning.

Because of the computational complexity involved in generating such solutions, its decision cycle is often in the order of minutes. The deliberative layer (or planning layer) uses models for decision making.

3 Pipeline architecture

Another architecture for robots is known as the **pipeline architecture**. Just like the subsumption architecture, the pipeline architecture executes multiple process in parallel.

Data enters this pipeline at the **sensor interface layer**. The **perception layer** then updates the robot's internal models of the environment based on this data. Next, these models are handed to the **planning and control layer**. Those are then communicated back to the vehicle through the **vehicle interface layer**.



The key to the pipeline architecture is that this all happens in parallel. While the perception layer processes the most recent sensor data, the control layer bases its choices on slightly older data. In this way, the pipeline architecture is similar to the human brain. We don't switch off our motion controllers when we digest new sensor data. Instead, we perceive, plan, and act all at the same time. Processes in the pipeline architecture run asynchronously, and all computation is data-driven. The resulting system is robust, and it is fast.

APPLICTION DOMAINS:

Industry and Agriculture. Traditionally, robots have been fielded in areas that require difficult human labour, yet are structured enough to be amenable to robotic automation. The best example is the assembly line, where manipulators routinely perform tasks such as assembly, part placement, material handling, welding, and painting. In many of these tasks, robots have become more cost-effective than human workers.

Transportation. Robotic transportation has many facets: from autonomous helicopters that deliver payloads to hard-to-reach locations, to automatic wheelchairs that transport people who are unable to control wheelchairs by themselves, to autonomous straddle carriers that outperform skilled human drivers when transporting containers from ships to trucks on loading docks.

Robotic cars. Most of use cars every day. Many of us make cell phone calls while
driving. Some of us even text. The sad result: more than a million people die every year in traffic accidents. Robotic cars like BOSS and STANLEY offer hope: Not only will they make driving much safer, but they will also free us from the need to pay attention to the road during our daily commute.

Health care. Robots are increasingly used to assist surgeons with instrument placement when operating on organs as intricate as brains, eyes, and hearts. Robots have become indispensable tools in a range of surgical procedures, such as hip replacements, thanks to their high precision. In pilot studies, robotic devices have been found to reduce the danger of lesions when performing colonoscopy.

Hazardous environments. Robots have assisted people in cleaning up nuclear waste, most notably in Chernobyl and Three Mile Island. Robots were present after the collapse of the World Trade Center, where they entered structures deemed too dangerous for human search and rescue crews.

Exploration. Robots have gone where no one has gone before, including the surface of Mars. Robotic arms assist astronauts in deploying and retrieving satellites and in building the International Space Station. Robots also help explore under the sea. They are routinely used to acquire maps of sunken ships.

Personal Services. Service is an up-and-coming application domain of robotics. Service robots assist individuals in performing daily tasks. Commercially available domestic service robots include autonomous vacuum cleaners, lawn mowers, and golf caddies. example for robot vaccum cleaner is ROOMBA.

Entertainment. Robots have begun to conquer the entertainment and toy industry. we see **robotic soccer**, a competitive game very much like human soccer, but played with autonomous mobile robots. Robot soccer provides great opportunities for research in AI, since it raises a range of problems relevant to many other, more serious robot applications.

Human augmentation. A final application domain of robotic technology is that of human augmentation. Researchers have developed legged walking machines that can carry people around, very much like a wheelchair. Several research efforts presently focus on the development of devices that make it easier for people to walk or move their arms by providing additional forces through extra skeletal attachments.

WEAK AI: CAN MACHINES ACT INTELLIGENTLY?

AI is impossible depends on how it is defined. we defined AI as the quest for the best agent program on a given architecture. With this formulation, AI is by definition possible: for any digital architecture with k bits of program storage there are exactly 2k agent programs, and all we have to do to find the best one is enumerate and test them all. This might not be feasible for large *k*, but philosophers deal with the theoretical, not the practical.

Our definition of AI works well for the engineering problem of finding a good agent, given an

architecture. Therefore, we're tempted to end this section right now, answering the title question in the affirmative. But philosophers are interested in the problem of compar- ing two architectures—human and machine. Furthermore, they have traditionally posed the question not in terms of maximizing expected utility but rather as, "**Can machines think**?"

Alan Turing, in his famous paper "Computing Machinery and Intelligence" (1950), sug- gested that instead of asking whether machines can think, we should ask whether machines can pass a behavioral intelligence test, which has come to be called the **Turing Test**. The test is for a program to have a conversation (via online typed messages) with an interrogator for five minutes. The interrogator then has to guess if the conversation is with a program or a person; the program passes the test if it fools the interrogator 30% of the time.

The argument from disability

The "argument from disability" makes the claim that "a machine can never do *X*." As exam- ples of *X*, Turing lists the following:

Be kind, resourceful, beautiful, friendly, have initiative, have a sense of humor, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream, make someone fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behavior as man, do something really new

It is clear that computers can do many things as well as or better than humans, including things that people believe require great human insight and understanding. This does not mean, of course, that computers use insight and understanding in performing these tasks those are not part of behavior, and we address such questions elsewhere but the point is that one's first guess about the mental processes required to produce a given behavior is often wrong. It is also true, of course, that there are many tasks at which computers do not yet excel (to put it mildly), including Turing's task of carrying on an open-ended conversation.

The mathematical objection

It is well known, through the work of Turing (1936) and Go¨del (1931), that certain math- ematical questions are in principle unanswerable by particular formal systems. Go¨del's in- completeness theorem is the most famous example of this. Briefly, for any formal axiomatic system F powerful enough to do arithmetic, it is possible to construct a so-called Go¨del sentence G(F) with the following properties:

- $G(F)$ is a sentence of F, but cannot be proved within F.
- If F is consistent, then $G(F)$ is true.

even if we grant that computers have limitations on what they can prove, there is no evidence that humans are immune from those limitations. It is all too easy to show rigorously that a formal system cannot do *X*, and then claim that humans *can* do *X* using their own informal method, without giving any evidence for this claim. Indeed, it is impossible to prove that humans are not subject to Go¨del's incompleteness theorem, because any rigorous proof would require a formalization of the claimed unformalizable human talent, and hence refute itself. So we are left with an appeal to intuition that humans can somehow perform superhuman feats of mathematical insight. This appeal is expressed with arguments such as "we must assume our own consistency, if thought is to be possible at all". But if anything, humans are known to be inconsistent. This is certainly true for everyday reasoning, but it is

also true for careful mathematical thought. A famous example is the four-color map problem.

The argument from informality

One of the most influential and persistent criticisms of AI as an enterprise was raised by Turing as the "argument from informality of behavior." Essentially, this is the claim that human behavior is far too complex to be captured by any simple set of rules and that because computers can do no more than follow a set of rules, they cannot generate behavior as intelligent as that of humans. The inability to capture everything in a set of logical rules is called the **qualification problem** in AI.

1. Good generalization from examples cannot be achieved without background knowledge. They claim no one has any idea how to incorporate background knowledge into the neural network learning process. In fact, that there are techniques for using prior knowledge in learning algorithms. Those techniques, however, rely on the availability of knowledge in explicit form, something that Dreyfus and Dreyfus strenuously deny. In our view, this is a good reason for a serious redesign of current models of neural processing so that they *can* take advantage of previously learned knowledge in the way that other learning algorithms do.
2. Neural network learning is a form of supervised learning, requiring the prior identification of relevant inputs and correct outputs. Therefore, they claim, it cannot operate autonomously without the help of a human trainer. In fact, learning without a teacher can be accomplished by **unsupervised learning** and **reinforcement learning** .
3. Learning algorithms do not perform well with many features, and if we pick a subset of features, "there is no known way of adding new features should the current set proveinadequate to account for the learned facts." In fact, new methods such as support vector machines handle large feature sets very well. With the introduction of large Web-based data sets, many applications in areas such as language processing (Sha and Pereira, 2003) and computer vision (Viola and Jones, 2002a) routinely handle millionsof features.
4. The brain is able to direct its sensors to seek relevant information and to process itto extract aspects relevant to the current situation. But, Dreyfus and Dreyfus claim, "Currently, no details of this mechanism are understood or even hypothesized in a waythat could guide AI research." In fact, the field of active vision, underpinned by the theory of information value , is concerned with exactly the problem of directing sensors, and already some robots have incorporated the theoretical results obtained

STRONG AI: CAN MACHINES REALLY THINK?

Many philosophers have claimed that a machine that passes the Turing Test would still not be *actually* thinking, but would be only a *simulation* of thinking. Again, the objection was foreseen by Turing. He cites a speech by Professor Geoffrey Jefferson (1949):

> Not until a machine could write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain—that is, not only write it but know that it had written it.

Turing calls this the argument from **consciousness**—the machine has to be aware of its ownmental states and actions. While consciousness is an important subject, Jefferson's key point actually relates to **phenomenology**, or the study of direct experience: the machine has to actually feel emotions. Others focus on **intentionality**—that is, the question of whether the machine's purported beliefs, desires, and other representations are actually "about" some- thing in the real world.

Turing argues that Jefferson would be willing to extend the polite convention to ma- chines if only he had experience with ones that act intelligently. He cites the following dialog, which has become such a part of AI's oral tradition that we simply have to include it:

HUMAN: In the first line of your sonnet which reads "shall I compare thee to a summer's day," would not a "spring day" do as well or better?
MACHINE: It wouldn't scan.
HUMAN: How about "a winter's day." That would scan all right.
MACHINE: Yes, but nobody wants to be compared to a winter's day.
HUMAN: Would you say Mr. Pickwick reminded you of Christmas?
MACHINE: In a way.
HUMAN: Yet Christmas is a winter's day, and I do not think Mr. Pickwick would mind the comparison.
MACHINE: I don't think you're serious. By a winter's day one means a typical winter's day, rather than a special one like Christmas.

Mental states and the brain in a vat

Physicalist philosophers have attempted to explicate what it means to say that a person—and, by extension, a computer—is in a particular mental state. They have focused in particular on **intentional states**. These are states, such as believing, knowing, desiring, fearing, and so on, that refer to some aspect of the external world. For example, the knowledge that one is eating a hamburger is a belief *about* the hamburger and what is happening to it.

If physicalism is correct, it must be the case that the proper description of a person's mental state is *determined* by that person's brain state. Thus, if I am currently focused on eating a hamburger in a mindful way, my instantaneous brain state is an instance of the class of mental states "knowing that one is eating a hamburger." Of course, the specific configurations of all the atoms of my brain are not essential: there are many configurations of my brain, or of other people's brain, that would belong to the same class of mental states. The key point is that the same brain state could not correspond to a fundamentally distinct mental state, such as the knowledge that one is eating a banana.

The "**wide content**" view interprets it from the point of view of an omniscient outside observer with access to the whole situation, who can distinguish differences in the world. Under this view, the content of mental states involves both the brain state and the environment history. **Narrow content**, on the other hand, considers only the brain state. The narrow content of the brain states of a real hamburger-eater and a brain-in-a-vat "hamburger"-"eater" is the same in both cases.

Functionalism and the brain replacement experiment

The theory of **functionalism** says that a mental state is any intermediate causal condition between input and output. Under functionalist theory, any two systems with isomorphic causal processes would have the same mental states. Therefore, a computer program could have the same mental states as a person. Of course, we have not yet said what "isomorphic" really means, but the assumption is that there is some level of abstraction below which the specific implementation does not matter.

And this explanation must also apply to the real brain, which has the same functional properties. There are three possible conclusions:

1. The causal mechanisms of consciousness that generate these kinds of outputs in normalbrains are still operating in the electronic version, which is therefore conscious.
2. The conscious mental events in the normal brain have no causal connection to behavior, and are missing from the electronic brain, which is therefore not conscious.
3. The experiment is impossible, and therefore speculation about it is meaningless.

Biological naturalism and the Chinese Room

A strong challenge to functionalism has been mounted by John Searle's (1980) biological naturalism, according to which mental states are high-level emergent features that are caused by low-level physical processes in the neurons, and it is the (unspecified) properties of the neurons that matter. Thus, mental states cannot be duplicated just on the basis of some pro- gram having the same functional structure with the same input–output behavior; we would require that the program be running on an architecture with the same causal power as neurons. To support his view, Searle describes a hypothetical system that is clearly running a program and passes the Turing Test, but that equally clearly (according to Searle) does not understand anything of its inputs and outputs. His conclusion is that running the appropriate program (i.e., having the right outputs) is not a sufficient condition for being a mind.

So far, so good. But from the outside, we see a system that is taking input in the form of Chinese sentences and generating answers in Chinese that are as "intelligent" as those in the conversation imagined by Turing.[4] Searle then argues: the person in the room doesnot understand Chinese (given). The rule book and the stacks of paper, being just pieces ofpaper, do not understand Chinese. Therefore, there is no understanding of Chinese. Hence, according to Searle, running the right program does not necessarily generate understanding.

The real claim made by Searle rests upon the following four axioms :

1. Computer programs are formal (syntactic).
2. Human minds have mental contents (semantics).
3. Syntax by itself is neither constitutive of nor sufficient for semantics.
4. Brains cause minds.

From the first three axioms Searle concludes that programs are not sufficient for minds. In other words, an agent running a program *might* be a mind, but it is not *necessarily* a mind justby virtue of running the program. From the fourth axiom he concludes "Any other system capable of causing minds would have to have causal powers (at least) equivalent to thoseof brains." From there he infers that any artificial brain would have to duplicate the causal powers of brains, not just run a particular program, and that human brains do not produce mental phenomena solely by virtue of running a program.

Consciousness, qualia, and the explanatory gap

Running through all the debates about strong AI—the elephant in the debating room, soto speak—is the issue of **consciousness**. Consciousness is often broken down into aspects such as understanding and self-awareness. The aspect we will focus on is that of *subjective experience*: why it is that it *feels* like something to have certain brain states (e.g., while eatinga hamburger), whereas it presumably does not feel like anything to have other physical states(e.g., while being a rock). The technical term for the intrinsic nature of experiences is **qualia**(from the Latin word meaning, roughly, "such things").

Qualia present a challenge for functionalist accounts of the mind because different qualia could be involved in what are otherwise isomorphic causal processes. Consider, for example, the **inverted spectrum**

thought experiment, which the subjective experience of per- son *X* when seeing red objects is the same experience that the rest of us experience when seeing green objects, and vice versa.

This **explanatory gap** has led some philosophers to conclude that humans are simply incapable of forming a proper understanding of their own consciousness. Others, notably Daniel Dennett (1991), avoid the gap by denying the existence of qualia,attributing them to a philosophical confusion.

THE ETHICS AND RISKS OF DEVELOPING ARTIFICIAL INTELLIGENCE

So far, we have concentrated on whether we can develop AI, but we must also consider whether we should. If the effects of AI technology are more likely to be negative than positive, then it would be the moral responsibility of workers in the field to redirect their research. Many new technologies have had unintended negative side effects: nuclear fission brought Chernobyl and the threat of global destruction; the internal combustion engine brought air pollution, global warming, and the paving-over of paradise. In a sense, automobiles are robots that have conquered the world by making themselves indispensable.

AI, however, seems to pose some fresh problems beyond that of, say, building bridges that don't fall down:

- People might lose their jobs to automation.
- People might have too much (or too little) leisure time.
- People might lose their sense of being unique.
- AI systems might be used toward undesirable ends.
- The use of AI systems might result in a loss of accountability.
- The success of AI might mean the end of the human race.

People might lose their jobs to automation. The modern industrial economy has be come dependent on computers in general, and select AI programs in particular. For example, much of the economy, especially in the United States, depends on the availability of con- sumer credit. Credit card applications, charge approvals, and fraud detection are now done by AI programs. One could say that thousands of workers have been displaced by these AI programs, but in fact if you took away the AI programs these jobs would not exist, because human labor would add an unacceptable cost to the transactions.

People might lose their sense of being unique. In *Computer Power and Human Rea- son*, Weizenbaum (1976), the author of the ELIZA program, points out some of the potential threats that AI poses to society. One of Weizenbaum's principal arguments is that AI research makes possible the idea that humans are automata—an idea that results in a loss of autonomyor even of humanity.

AI systems might be used toward undesirable ends. Advanced technologies have often been used by the powerful to suppress their rivals. As the number theorist G. H. Hardy wrote (Hardy, 1940), "A science is said to be useful if its development tends to accentuate the existing inequalities in the distribution of wealth, or more directly promotes the destruction of human life." This holds for all sciences, AI being no exception. Autonomous AI systems are now commonplace on the battlefield; the U.S. military deployed over 5,000 autonomousaircraft and 12,000 autonomous ground vehicles in Iraq (Singer, 2009).

The use of AI systems might result in a loss of accountability. In the litigious atmo- sphere that prevails in the United States, legal liability becomes an important issue. When aphysician relies on the judgment of a medical expert system for a diagnosis, who is at fault ifthe diagnosis is wrong? Fortunately, due in part to the growing influence of decision-theoreticmethods in medicine, it is now accepted that negligence cannot

be shown if the physician performs medical procedures that have high *expected* utility, even if the *actual* result is catastrophic for the patient.

The success of AI might mean the end of the human race. Almost any technology has the potential to cause harm in the wrong hands, but with AI and robotics, we have the new problem that the wrong hands might belong to the technology itself. Countless science fiction stories have warned about robots or robot–human cyborgs running amok.

If ultra intelligent machines are a possibility, we humans would do well to make sure that we design their predecessors in such a way that they design themselves to treat us well. Science fiction writer Isaac Asimov (1942) was the first to address this issue, with his 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 orders given to it by human beings, except where such orders would conflict with the First Law.

A robot must protect its own existence as long as such protection does not conflict with the First or Second Law

AGENT COMPONENTS

Interaction with the environment through sensors and actuators: For much of the history of AI, this has been a glaring weak point. With a few honorable exceptions, AI sys- tems were built in such a way that humans had to supply the inputs and interpret the outputs,

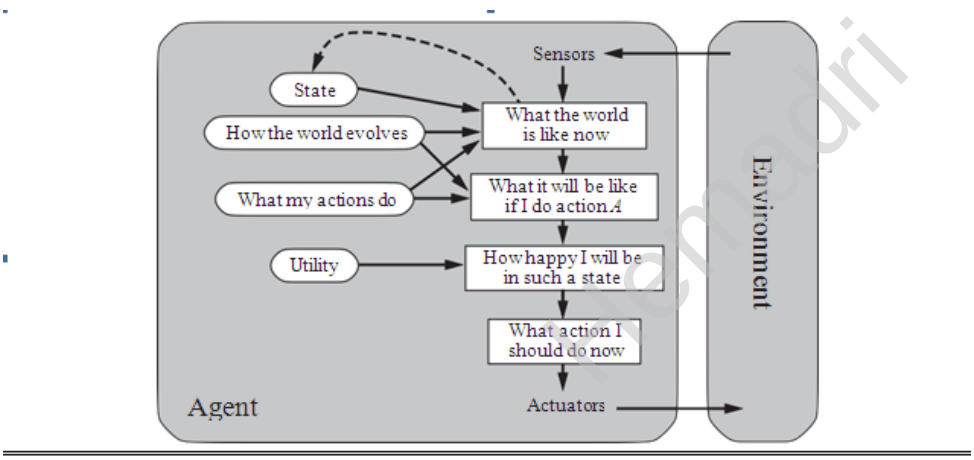


Figure A model-based, utility-based agent

while robotic systems focused on low-level tasks in which high-level reasoning and plan- ning were largely absent. This was due in part to the great expense and engineering effort required to get real robots to work at all. The situation has changed rapidly in recent years with the availability of ready-made programmable robots. These, in turn, have benefited from small, cheap, high-resolution CCD cameras and compact, reliable motor drives. MEMS (micro-electromechanical systems) technology has supplied miniaturized accelerometers, gy- roscopes, and actuators for an artificial flying insect (Floreano *et al.*, 2009). It may also be possible to combine millions of MEMS devices to produce powerful macroscopic actuators.

Keeping track of the state of the world: This is one of the core capabilities required for an intelligent agent. It requires both perception and updating of internal representations. showed how to keep track of atomic state representations, described how to do it for factored (propositional) state representations extended this to first-order logic; and Chapter 15 described **filtering** algorithms for probabilistic reasoning

in uncertain environments. Current filtering and perception algorithms can be combined to do a reasonable job of reporting low-level predicates such as "the cup is on the table." Detecting higher-level actions, such as "Dr. Russell is having a cup of tea with Dr. Norvig while dis- cussing plans for next week," is more difficult. Currently it can be done only with the help of annotated examples.

Projecting, evaluating, and selecting future courses of action: The basic knowledge- representation requirements here are the same as for keeping track of the world; the primary difficulty is coping with courses of action—such as having a conversation or a cup of tea—that consist eventually of thousands or millions of primitive steps for a real agent. It is only by imposing **hierarchical structure** on behavior that we humans cope at all.how to use hierarchical representations to handle problems of this scale; fur- ther more, work in **hierarchical reinforcement learning** has succeeded in combining some of these ideas with the techniques for decision making under uncertainty described in. As yet, algorithms for the partially observable case (POMDPs) are using the same atomic state representation we used for the search algorithms

It has proven very difficult to decompose preferences over complex states in the same way that Bayes nets decompose beliefs over complex states. One reason may be that preferences over states are really *compiled* from preferences over state histories, which are described by **reward functions**

Learning: Chapters 18 to 21 described how learning in an agent can be formulated as inductive learning (supervised, unsupervised, or reinforcement-based) of the functions that constitute the various components of the agent. Very powerful logical and statistical tech- niques have been developed that can cope with quite large problems, reaching or exceeding human capabilities in many tasks—as long as we are dealing with a predefined vocabulary of features and concepts.

AGENT ARCHITECTURES

It is natural to ask, "Which of the agent architectures should an agent use?" The answer is, "All of them!" We have seen that reflex responses are needed for situations in which time is of the essence, whereas knowledge-based deliberation allows the agent to plan ahead. A complete agent must be able to do both, using a **hybrid architecture**. One important property of hybrid architectures is that the boundaries between different decision components are not fixed. For example, **compilation** continually converts declarative in- formation at the deliberative level into more efficient representations, eventually reaching the reflex level.

For example, a taxi-driving agent that sees an accident ahead must decide in a split second either to brake or to take evasive action. It should also spend that split second thinking about the most important questions, such as whether the lanes to the left and right are clear and whether there is a large truck close behind, rather than worrying about wear and tear on the tires or where to pick up the next passenger. These issues are usually studied under the heading of **real-time AI**

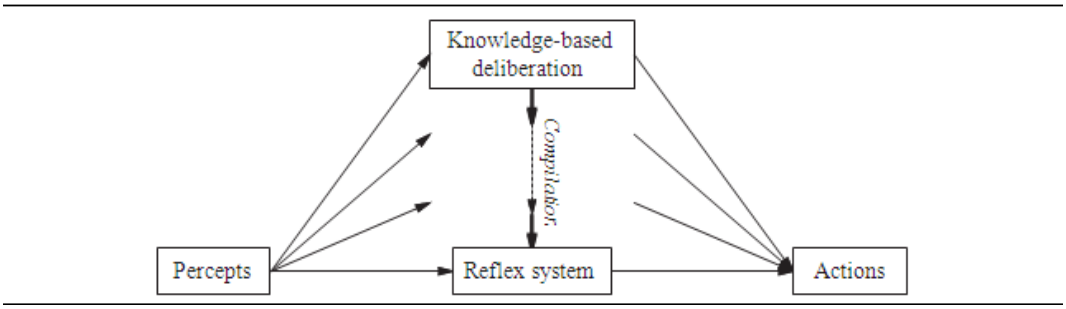


Fig: Compilation serves to convert deliberative decision making into more efficient, reflexive mechanisms. Clearly, there is a pressing need for *general* methods of controlling deliberation, ratherthan specific recipes for what to think about in each situation. The first useful idea is to employ **anytime algorithms**

The second technique for controlling deliberation is decision-theoretic meta reasoning (Russell and Wefald, 1989, 1991; Horvitz, 1989; Horvitz and Breese, 1996). This method applies the theory of information value to the selection of individual computations. The value of a computation depends on both its cost (in terms of delaying action) andits benefits (in terms of improved decision quality). Meta reasoning techniques can be used to design better search algorithms and to guarantee that the algorithms have the anytime property. Meta reasoning is expensive, of course, and compilation methods can be applied so that the overhead is small compared to the costs of the computations being controlled. Meta level reinforcement learning may provide another way to acquire effective policies for controllingdeliberation

Meta reasoning is one specific example of a **reflective architecture**—that is, an architecture that enables deliberation about the computational entities and actions occurring withinthe architecture itself. A theoretical foundation for reflective architectures can be built by defining a joint state space composed from the environment state and the computational stateof the agent itself.

ARE WE GOING IN THE RIGHT DIRECTION?

The preceding section listed many advances and many opportunities for further progress. Butwhere is this all leading? Dreyfus (1992) gives the analogy of trying to get to the moon by climbing a tree; one can report steady progress, all the way to the top of the tree. In this section, we consider whether AI's current path is more like a tree climb or a rocket trip.

Perfect rationality. A perfectly rational agent acts at every instant in such a way as to maximize its expected utility, given the information it has acquired from the environment. We have seen that the calculations necessary to achieve perfect rationality in most environments are too time consuming, so perfect rationality is not a realistic goal.

Calculative rationality. This is the notion of rationality that we have used implicitly in de- signing logical and decision-theoretic agents, and most of theoretical AI research has focusedon this property. A calculatively rational agent *eventually* returns what *would have been* therational choice at the beginning of its deliberation. This is an interesting property for a systemto exhibit, but in most environments, the right answer at the wrong time is of no value. In practice, AI system designers are forced to compromise on decision quality to obtain reason- able overall performance; unfortunately, the theoretical basis of calculative rationality does not provide a well-founded way to make such compromises.

Bounded rationality. Herbert Simon (1957) rejected the notion of perfect (or even approximately perfect) rationality and replaced it with bounded rationality, a descriptive theory of decision making by real agents.
Bounded optimality (BO). A bounded optimal agent behaves as well as possible, *given its computational resources*. That is, the expected utility of the agent program for a bounded optimal agent is at least as high as the expected utility of any other agent program running onthe same machine.

WHAT IF AI DOES SUCCEED?

In David Lodge's *Small World* (1984), a novel about the academic world of literary criticism, the protagonist causes consternation by asking a panel of eminent but contradictory literary theorists the following question: "What if you were right?" None of the theorists seems to have considered this question before, perhaps because debating unfalsifiable theories is an end in itself. Similar confusion can be evoked by asking AI researchers, "*What if you succeed?*"

We can expect that medium-level successes in AI would affect all kinds of people in their daily lives. So far, computerized communication networks, such as cell phones and the Internet, have had this kind of pervasive effect on society, but AI has not. AI has been at workbehind the scenes—for example, in automatically approving or denying credit card transactions for every purchase made on the Web—but has not been visible to the average consumer.We can imagine that truly useful personal assistants for the office or the home would have a large positive impact on people's lives, although they might cause some economic dislocation in the short term. Automated assistants for driving could prevent accidents, saving tens of thousands of lives per year. A technological capability at this level might also be applied to the development of autonomous weapons, which many view as undesirable. Some of the biggest societal problems we face today—such as the harnessing of genomic information fortreating disease, the efficient management of energy resources, and the verification of treaties concerning nuclear weapons—are being addressed with the help of AI technologies.

Finally, it seems likely that a large-scale success in AI—the creation of human-level in-telligence and beyond—would change the lives of a majority of humankind. The very natureof our work and play would be altered, as would our view of intelligence, consciousness, andthe future destiny of the human race. AI systems at this level of capability could threaten hu-man autonomy, freedom, and even survival. For these reasons, we cannot divorce AI researchfrom its ethical consequences

In conclusion, we see that AI has made great progress in its short history, but the final sentence of Alan Turing's (1950) essay on *Computing Machinery and Intelligence* is still valid today:

We can see only a short distance ahead, but we can see that much remains to be done.