# 4

# INSIDE THE MACHINE

## *What is actually going on? How does the robot's "brain" work?*

Through the many examples encountered so far in this book, it is clear that all robots can be described in terms of sensors, actuators, overall body parts, and a control system. Control involves generation and coordination of behavior by somehow using incoming sensory information to produce appropriate actuator outputs. The exact details of what goes on between the sensors and the actuators—how the control architecture, the robot's "brain" works—can vary enormously. In Chapter 3 we saw examples ranging from the surprisingly effective behavior of Grey Walter's tortoises, with two richly interconnected electronic valves and a few simple sensors, to the grindingly slow, brittle outcomes of the gargantuan, reasoning-based control systems of traditional AI robotics. In the former, the sensor signals flow continuously through the electronic "nervous system," directly triggering motor outputs that in turn change sensor inputs as the robots move through the environment. In the latter, sensor inputs are processed and probed to build and modify a central world model, made of logic, which is used by a series of reasoning modules to try to figure out what to do next. No module can start its work until the preceding one in the pipeline has finished. If a plan can be made, its steps are then finally translated into actuator commands, which are turned into motor outputs. This and some of the other main types of control architectures are illustrated in Figure 4.1.

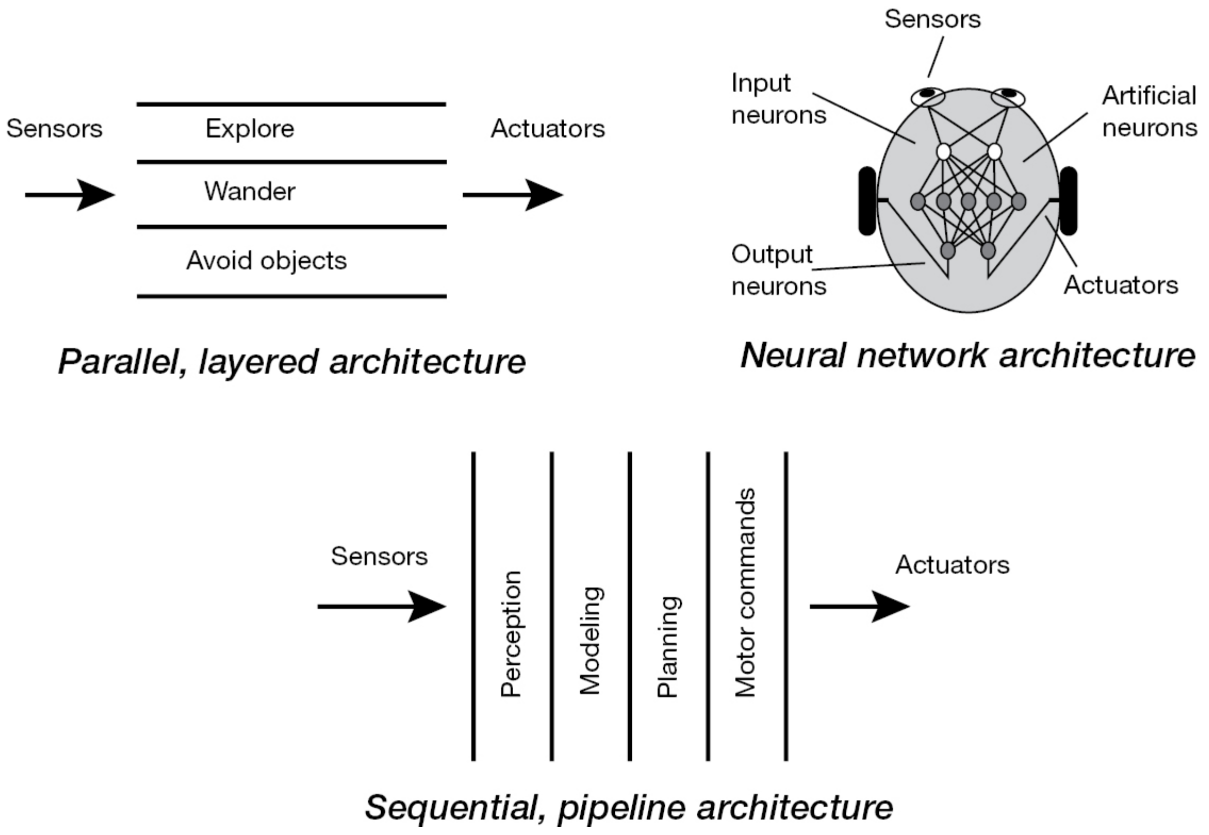

**Figure 4.1** Some standard control architecture types.

Walter’s tortoises used an early example of an artificial neural network (ANN) employed in a distributed parallel architecture. An ANN is a network of simple processing units (artificial neurons) that is very loosely analogous to a biological neural network. Later ANN-based architectures were rather different to Walter’s, not least because the networks were implemented in software on computer chips. They too involved sensors feeding into designated input units and actuator signals being taken from motor output units. In between the input and output units, an inscrutable tangle of interconnected artificial neurons processed signals flowing around the network, often in rather enigmatic ways (see examples in Figures 4.1 and 4.2).

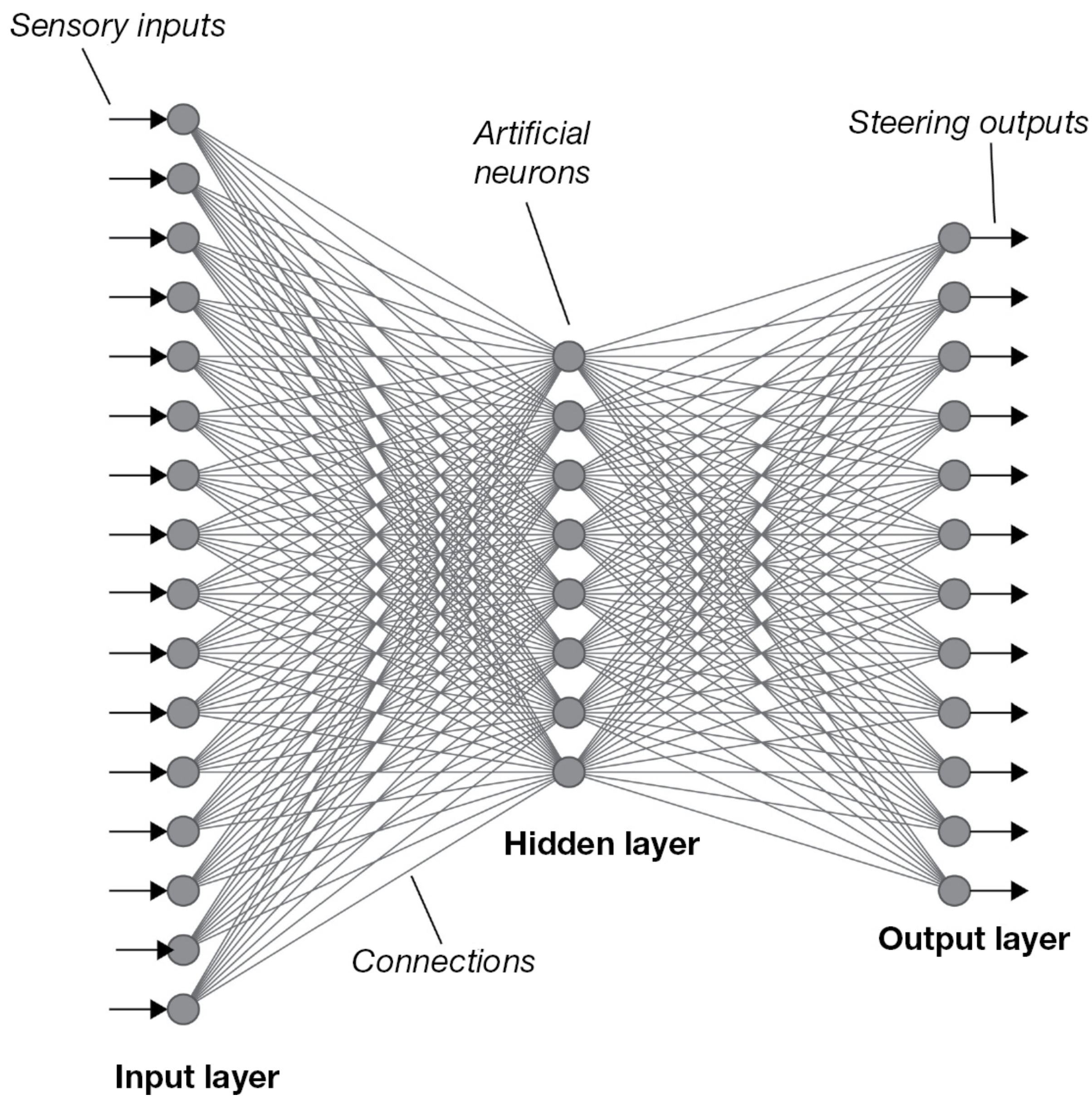

**Figure 4.2** An ANN of the kind used by an autonomous land vehicle in a neural network (ALVINN). Filled circles are artificial neurons which are joined together by connections, represented by lines. This is a three-layer feed-forward network. The input layer feeds into the hidden layer which feed into the output layer. Signals flow from sensory inputs through to motor outputs. Each of the neurons in the input layer connects to all the neurons in the hidden layer, and each neuron in the hidden layer connects to all the neurons in the output layer.

The behavior-based robotics control architectures of the 1990s and 2000s that we met in Chapter 3 were also distributed and parallel, favoring a layered design (Figure 4.1 top left), in direct contrast to the strict sequential pipeline of most traditional AI approaches (Figure 4.1 bottom). In the behavior-based approach each layer is usually made from a collection of simple processing units, or simple computer functions, that together generate a particular behavior. A layer can be thought of as a *level of competence*, with the simpler competences at the bottom of the vertical decomposition and the more complex ones at the top. The behavior layers each have access to sensor inputs and produce motor outputs—they can act as standalone control systems. They interact in such a way that, although the default position is often all behaviors running independently in parallel, this can switch to one layer overriding all others in certain circumstances, or higher-level behaviors taking control of lower-level competences.[1]

Today, as we'll see, although most practical architectures are distributed and parallel, they are often hybrid, using elements of all these approaches.

## *How many ways are there to control a robot?*

Countless. As many as the fevered imaginations of engineers and scientist can dream up. However, at any one time there will be just a few favored methods and techniques that most people use. These will change over time as new discoveries occur and certain methods prove themselves to be better in important practical cases. Most scientists and engineers follow the herd, but there are always a few mavericks and rebels who are crucial to pushing things forward.

Roboticists have a variety of motivations which will influence the kinds of methods they use. Some are heavily motivated by a particular theory and philosophy—they will want to demonstrate its worth and probe its limitations. Others, particularly those working in commercial environments, are much more motivated by getting stuff to work well in real applications. The latter case is generally much less purist—whatever does the job now is fine.

The overall approach taken will also be task dependent. The detailed requirements for control of a visually guided autonomous mobile robot are generally rather different from those of a bench-bound non-autonomous robot arm used to stack parts on a conveyor belt, as are those of an eight-legged walking robot moving materials around the complex terrain of a forest compared to a small flying drone intended to provide rapid active surveillance in a giant warehouse.

## *How do driverless cars work?*

A good way to start exploring the inner workings of modern mobile robots in more depth is to tackle one of the most important questions of the day: How do driverless cars work? Autonomous vehicles are a very good exemplar of current mobile robotics as they employ many of today's cutting-edge techniques, and indeed have been a major driver in their development. They may also have a significant impact on many lives in the near- to mid-term. So let's look at them in a bit more detail.

Driverless cars are mobile autonomous robots that use a variety of sensors to pull in information about the surroundings, and a series of actuators to control steering, acceleration, and braking. They have one main task: get from A to B without causing any danger or damage, while staying strictly within the rules of the road. In closed, quiet neighborhoods this is a relatively constrained problem. But, in general, in potentially busy, volatile traffic conditions where pedestrians and cyclists might be thrown into the mix, this is a non-trivial problem. To understand how things work today it is useful to first go back in time.

## *When did work on driverless cars start?*

Attempts to develop driverless vehicles go right back to the 1920s, only a few years after the first production automobiles started to appear. At that time, radio-controlled cars, which could automatically follow a leading vehicle transmitting instructions, were investigated. From the 1930s onward a great deal of effort went into systems involving cars that followed guiding circuits embedded in the road, with automatic control of traffic at junctions. While some of these approaches looked promising, it became increasingly clear that they would be difficult to implement in practice. Unless everyone was forced to use automatically guided vehicles it was hard to see how the system would work smoothly with a mix of guided and normal vehicles. And then there was the question of the cost of embedding the necessary systems in the roads. But perhaps most importantly, it was unclear how safe these vehicles would be. They were road followers, their sensors geared toward that task. What would happen if one of them broke down, or someone stumbled into their path, or something completely unexpected happened? From the 1980s onward attention switched to equipping standard vehicles with sensors and "intelligent" control systems that would allow them to operate safely in normal traffic.

A very significant step in this quest occurred in 1989 when Dean Pomerleau and a team at Carnegie Mellon University (CMU) demonstrated how an artificial neural network could be used to successfully steer a test vehicle around the CMU campus roads.[2] Pomerleau's system, ALVINN (autonomous land vehicle in a neural network), controlled the steering of CMU's NAVLAB, a converted Chevrolet van used for autonomous vehicle research.[3] NAVLAB had two main sensors, both mounted on the roof above the cab: a color video camera and a scanning laser rangefinder. It also had racks of computers and other equipment in the back which were connected to the sensors and actuators of the van (which was of course a large mobile robot). ALVINN used two "retinas" as sensory input. The first was derived from a 30×32 image continuously fed from the camera (very low resolution by today's standards) and the second was an 8×32 "image" from the laser rangefinder (which scanned over a volume 80 degrees wide by 30 degrees high). The values in this second retina were proportional to the distance to surrounding surfaces at the corresponding points in the scanned image. These surfaces could include the road, the verges, trees, or obstacles. The retinal values were fed directly to 1,216 input units (one for each retinal element) of an ANN (see Figure 4.2 for an illustration of the kind of ANN used). The output of the network determined the steering angle the wheels needed to be set at to accurately follow the road. This output was used by a servomechanism to automatically steer the vehicle. A human was in charge of other controls (accelerator, brake, etc.), but ALVINN was able to steer autonomously.

## *How do the artificial neural networks used in cars and robots work?*

The units (artificial neurons) in an ANN are joined together by connections through which "signals" flow (Figure 4.2). An ANN will have some dedicated input and output neurons which connect it to the task it is to perform (for a robot these will usually be directly connected to sensor inputs and motor outputs, as illustrated in Figures 4.1 and 4.2). The artificial neurons typically perform some simple mathematical transformation on the sum of all the signals from their incoming connections. This is used to produce an output value (which then becomes an input to any neurons connected downstream).[4] For instance, the neurons in the middle (hidden) layer of the three-layer network shown in Figure 4.2 each take inputs from all the neurons in the first (input) layer. They then pass their output simultaneously to each of the neurons in the third (output) layer.

Although ANNs can be built out of hardware, as Grey Walter did all those years ago, nowadays in most cases they are implemented in software; that is, they are simulated on a computer. The connections between units usually have "weights" (or strengths): these are simply numbers in some pre-defined range (for instance –2 to 2). The output from the neuron the connection originates *from* is simply multiplied by the connection's weight to produce the input at the neuron it is connected *to;* in other words, the signal flowing through the connection is multiplied by the weight. This means the signal on a connection with a higher weight will have more "influence" at the neuron into which it flows. In most cases an ANN "learns" by adjusting the values of these weights (increasing some and decreasing others). The input values feed through the mesh of weighted connections, combining in potentially complex ways, producing the output signals at the other end. Hence if the weights are changed, the network will process the values feeding through it differently and produce changed outputs. The idea, then, is that the weights are altered, according to the rules of a learning algorithm, until any set of inputs for the task at hand produces the desired outputs after they have flowed through the network. Learning is a matter of searching for the best set of connection weights in an efficient way.

ANNs started to become particularly popular in robotics and AI at about the time of Pomerleau's pioneering work at CMU because of a number of key properties that had potential to overcome the weaknesses of traditional AI methods. Apart from their ability to learn and adapt, and their use of parallel distributed processing, they could generalize to unseen data and deal with incomplete and/or noisy sensor signals.

There are generally three types of learning that can be applied to ANNs. *Supervised learning* involves the ANN being given many training examples where the desired output is known. After each example the weights (and sometimes other properties) of the network are adjusted according to how close the actual network output is to the correct output. As learning proceeds, the weights shift and jiggle around until they crystallize into a stable set of values. Successful learning produces a set of weights such that inputs to the network produce the desired output (in the case of ALVINN, the steering angle as in Figure 4.2).

As the name suggests, *unsupervised learning* does not make use of comparisons with the correct answers; it is not given explicit guidance. Again, the network is given many training examples, but learning now involves the ANN self-organizing around hidden patterns it detects in the data. These patterns often involve complex interactions between information from different inputs (sensors in the case of robots) which are difficult/impossible for humans to spot. This information can then be very helpful in classifying inputs into different groups. These might refer to different types of objects picked out by a robot's visual system, or the classification of different types of environmental conditions which can then be fed into a robot vehicle's control system. In unsupervised learning the ANN weights are often adjusted using a remarkably simple but effective learning rule first proposed by Donald Hebb back in 1949.[5] Using the Hebb rule, a connection between two artificial neurons that are simultaneously strongly active is strengthened; otherwise the connection weight is reduced.

*Reinforcement learning* involves using feedback from the environment to refine a behavior. It makes use of reward signals that indicated the appropriateness of a system's action in relation to its current state. This can be particularly suited to some problems in robotics. Robot actions in the environment can generate a reward related to its overall behavior (e.g., how far it moved without crashing, how many times it moved in the correct direction according to an external visual cue, and so on). The idea is to maximize reward. The reward signals help to guide a trial-and-error learning process toward good behaviors—plenty of reward means the robot is doing the right thing; little reward means it is not and needs to try something else. Popular methods at the moment include deep reinforcement learning where a deep (many-layered) neural network processes sensor input and produced motor outputs. As it proceeds, the network's weights and other parameters are adjusted by algorithms that take into account the amount of reward received.[6] Gradually the system settles down into a state where it is doing the right thing more or less all the time.

There are numerous variations on the exact details, but these kinds of processes are at the heart of nearly all modern machine learning, which now underpins much of robotics and AI. Many robot control systems use combinations of all three types of learning, distributed across various sub-systems.

## *What was the structure of ALVINN’s neural network?*

ALVINN’s ANN was a three-layer feed-forward network. Although somewhat larger, the basic structure was the same as the network shown in Figure 4.2. An input layer fed into a middle layer which fed into an output layer. The sensor values fed directly into 1,216 input units. Each of the input units was connected to every one of 29 units in the middle (so-called hidden) layer, making a total of 1,217 × 29 = 35,293 connections (and weights) between the first two layers. Each of the middle layer units was connected to each of the 46 units in the output layer, making another 29 × 46 = 1,334 connections. Forty-five of the output units represented steering angles (the unit with the highest output determined the angle to be used by NAVLAB).[7]

ALVINN’s ANN was trained with recorded and simulated input data using a supervised learning method.[8] The trained network was able to successfully steer the vehicle along a 400-m course through a wooded area of the CMU campus under sunny conditions. The whole setup pushed the limits of available technology at the time (which was much slower than current computers and sensors), and so the vehicle was only able to travel at 0.5 m/s (a little over 1 mile/hour), but that was state-of-the-art in 1989.

From these beginnings—a van crawling at snail’s pace along a short stretch of road on a quiet campus—the technology developed (aided by rapid increases in computer power) such that within a few years ANNs could help control vehicles traveling at normal traffic speeds on open roads. Larger networks with more layers and inputs (so-called deep learning neural networks), using extended versions of the learning algorithms employed by ALVINN, are now routinely at work on many autonomous vehicles, particularly for visual processing. They are used for recognizing pedestrians or cyclists, or interpreting road signs, as well as for keeping in lane and road following. Another useful property of neural networks, as exemplified by ALVINN, is the seamless way data from different kinds of sensors can be integrated. The camera and laser rangefinder inputs are fed into the network in exactly the same way. They mingle and combine in potentially complex ways as they feed through the network. This merging of information from different sensor types is known as *sensor fusion* and can be very useful in coping with sensor noise and uncertainty—two sources of information is usually better than one.

## *How does an autonomous vehicle "know" where it is?*

ALVINN set the ball rolling on one aspect of driverless vehicles—steering—but it did not produce a fully autonomous system. There were still many other parts to tackle, not least being able to plan and follow a route in order to get from A to B. This usually requires a solution to the mapping problem: building some sort of representation of the robot's environment that can be used for accurate navigation. A related problem is that of localization—the ability of a robot to determine where it is, relative to a map, from its sensor readings (commercially available GPS SATNAV is not accurate enough to do this safely; it can have errors of up to several meters, which could be disastrous for a self-driving vehicle, but it can be a useful extra input).

Since the early 1990s, following pioneering research at Oxford University, most work in this area has concentrated on the simultaneous localization and mapping (SLAM) problem.[9] This requires a mobile robot, when placed at an unknown spot in an unknown environment, to incrementally construct a consistent map of the environment, at the same time as determining its location on the map. A great deal of progress has been made on this problem, and for certain types of environments very good solutions have been found. Nearly all these solutions rely on probabilistic inference. Indeed, real progress on the problem was made only once it was cast in probabilistic terms—that is, in terms of probabilities (or likelihoods) that certain things are true (the current location is X, the current sensor readings imply that the location is not Y, and so on). Once these values have been estimated with sufficient accuracy, the location can be assumed to be the place with highest probability. Today's solutions use probabilistic models of the robot and its environment, employing probabilistic inference in building maps from the robot's sensor readings. More and more accurate estimates of the pertinent probabilities are built up as the robot gathers evidence about the world as it moves through it. The success of the probabilistic approach stems from the fact that the mapping problem is inherently uncertain and robot sensors are noisy, as is robot movement. Sensor readings drift and fluctuate due to inherent noise in the electronics or changes in environmental conditions. Motor actions are often not quite perfect due to inherent noise and inaccuracy in the physical systems making up the actuators, or through tricky environmental conditions (e.g., a robot foot or tire might not grip quite as well on soft ground as on firm surfaces). The probabilistic approaches embrace these characteristics of the problem rather than ignoring them or trying to hide them.

Typically a map describes the locations of and relationships between key features picked out by the robot's perceptual system, along with various aspects of the geometrical layout of the environment. The process of refining estimates of probabilities through the gathering of evidence, that is, the values of the probabilities associated with the robot position and the map, is made possible by what is known as Bayesian inference. It is named after Thomas Bayes, an eighteenth-century English mathematician and Presbyterian minister who first developed some of the underlying mathematics (known as Bayes' theorem). Well-defined, efficient mathematical procedures have been developed to make the necessary updates and refinements, ensuring that the estimates of probabilities become better and better as the robot gathers more information about the world.[10] After a while the estimates become extremely good, allowing locations to be determined very accurately. Bayesian approaches to the robotics SLAM problem emerged at about the same time as behavior-based and biologically inspired approaches. Although its methodology is mainly complementary to that of those areas, it certainly has overlapping concerns. Just as biologically inspired robotics can be traced back to the work of Grey Walter, many of the methods at the core of probabilistic robotics also have their roots in cybernetics.[11]

One of the first really notable successes using these Bayesian methods was achieved by a team from Stanford University led by Sebastian Thrun, who in 2005 won the DARPA Grand Challenge for an autonomous vehicle to navigate across an unrehearsed 142-mile course in the Mojave Desert.[12] A few years later Thrun was brought in by Google to start a research program on driverless cars, which eventually led to the emergence of Waymo, a subsidiary company which is currently one of the global leaders in self-driving vehicle technology.

## *What kind of sensors do current autonomous vehicles use?*

Current leading-edge autonomous vehicles use a variety of sensors (see Figure 4.3). The main varieties are: vision (cameras), light detection and ranging (LIDAR), radar, ultrasonics, and GPS information. Quite often, all of these are used together.

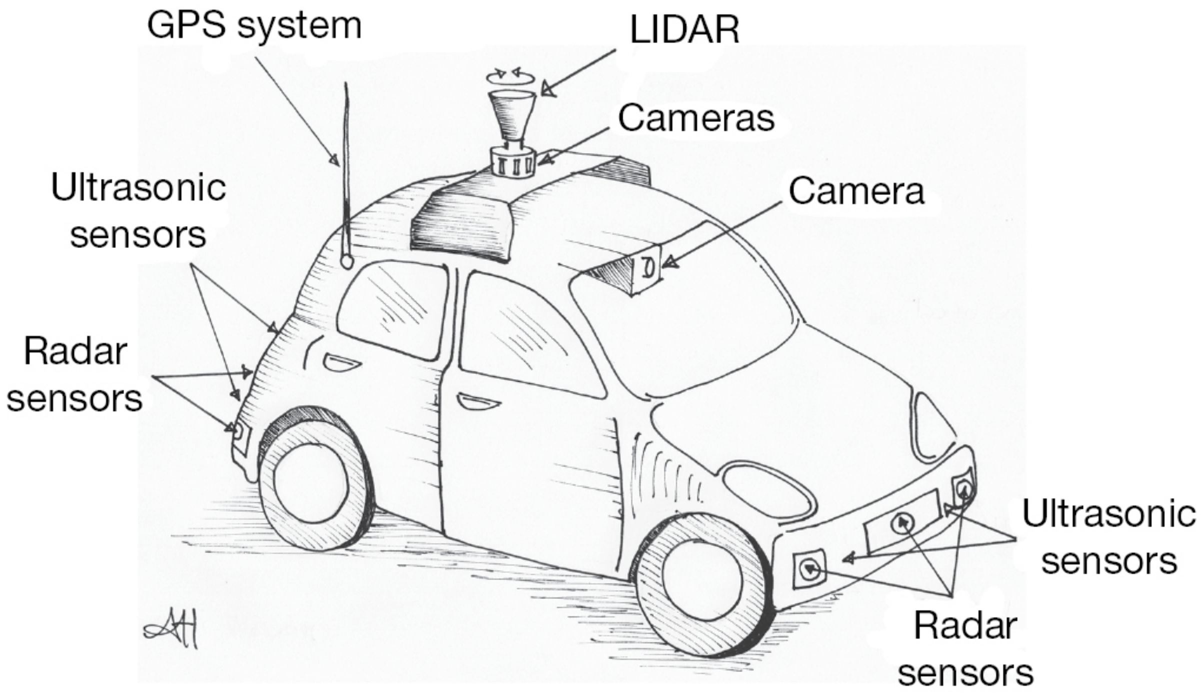

**Figure 4.3** The most common sensors used by driverless cars.

LIDAR[13] sensing involves a rapidly rotating array of laser rangefinders mounted on the roof, giving a real-time, 360-degree panoramic image of the surroundings up to a range of about 200 m. It provides very accurate information on distances to surrounding surfaces, which can be used to create a kind of 3-D model of the environment. LIDAR has been shown to be effective in building up maps of urban areas using the probabilistic SLAM approaches outlined earlier. To date, Google/Waymo have focused on an approach to autonomous vehicles that involves building very good maps of certain areas with LIDAR. Vehicles always have access to these maps such that they only need to concentrate on the localization part of the SLAM problem, again using LIDAR data. This means they can usually navigate well on networks of pre-mapped roads, which could be effective for urban driverless taxi services and the like. These maps are consulted by the control system to check on location, make decisions about when to change speed, decide if it’s safe to make a turn, and so on.

Forward and rear bumper-mounted radars (which bounce radio waves off objects in their path) are able to detect the approximate speed and distance of the vehicles ahead and behind, and can locate other objects in the vehicle’s path. Ultrasonic sensors (as used in automatic parking systems) detect very close objects to help avoid collisions or bumping into the curb.

Vision can be used for a variety of things including road following and identifying road signs, pedestrians, other vehicles, and hazards. Most visual systems are powered by (deep learning) ANNs which have been trained to make appropriate classifications (e.g., pedestrian ahead) or output control signals (e.g., steer slightly to the right to stay in the center of the lane). Typically, several ANNs doing different jobs will be taking inputs from the visual system.

The various sensors all have their strengths and weaknesses but work well in combination. LIDAR is fairly high resolution and gives extremely accurate information on distances to surrounding surfaces and objects (depth information), gives 360-degree coverage, and works in the dark, but it is very expensive—an automotive grade LIDAR system costs about $80,000 (in 2019). However, the price of a LIDAR unit may plummet over the next few years as companies work on developing much cheaper systems.[14] In contrast, vision sensors (cameras) are cheap and provide very high resolution—but sometimes noisy—information. Roads are designed for humans, and our primary sense for driving is vision. While ANN-powered vision modules are no match for our visual system, they are good at recognizing key features, interpreting road signs, and, as we've seen, can successfully look after road following. However, they do not work in the dark, it is difficult to retrieve depth information from them, and their performance degrades badly in poor weather conditions (fog, snow and ice, heavy rain). Radar has none of these problems and is cheap, but it is low resolution; it is mainly used for fail-safe obstacle detection and for measuring the speed of surrounding traffic.

Typical autonomous vehicle control architectures are distributed and parallel, with multiple sensor systems feeding into various AI and control modules. The AI systems often include SLAM navigation, visual recognition, road following, obstacle avoidance, and overall planning and coordination elements. These modules are spread over multiple on-board computers and special-purpose chips which must communicate and coordinate in order to coherently control acceleration, braking, and steering in order to safely achieve the current goal, which might be to stay in lane, or change lane, or turn left at the junction when the traffic lights go green, or slow down to avoid the pedestrian crossing the road, and so on.

## *What are the main approaches to achieving the goal of widespread fully autonomous vehicles?*

At the moment there are basically two main approaches to autonomous vehicles. The first, as exemplified by Google/Waymo and Aurora (who bought Uber's autonomous vehicle division in December 2020), is based around fully autonomous vehicles operating in slow-speed, urban areas that have been mapped in great detail and present relatively constrained, benign driving conditions. Such an approach currently relies heavily on LIDAR-based localization and navigation. As progress is made on solving all the problems associated with perceiving and correctly interpreting unexpected events in the environment (which might be rare but could lead to a fatal accident if not handled properly), the longer-term plan is to move out into less-constrained scenarios. The direction of travel is toward more and more sophisticated autonomy, with the ultimate aim of being able to handle any driving conditions, not just the simplest. This is proving to be a challenging goal.

The second approach, as exemplified by Tesla and many of the mainstream automotive manufacturers, is to move from semi-autonomous vehicles toward fully autonomous vehicles. Semi-autonomous vehicles make use of increasingly sophisticated autopilot, cruise-control-type systems in certain driving conditions. But the human driver is nearly always in the loop. Currently, the semi-autonomous systems mainly use vision, in conjunction with radar, ultrasonics, and GPS SATNAV data. They work particularly well in another kind of relatively constrained, but very different, scenario: highway/motorway driving. The human driver can take over at any point and can switch between manual driving and autopilot mode. These kinds of systems enable the vehicle to automatically steer, accelerate, and brake within its lane, and, in some more advanced systems, even change lane and overtake other cars. Descendants of ALVINN, they make heavy use of neural net-based AI and are thus able to learn and improve over time as more and more training data become available as it is collected by the vehicle and others like it. Currently they require active driver supervision at all times, except in certain restricted conditions. In the longer term the plan is to move toward full autonomy. A big difference between truly autonomous and semi-autonomous vehicles is that in the latter the driver is ultimately responsible for safety; in the former AI has our lives in its hands.

Another area where there is a great deal of research, particularly in relation to semi-autonomous cars, is in using AI for monitoring the state of the driver. Cameras and other sensors are being used in conjunction with machine-learning systems (usually ANNs) to spot if the driver seems not to be paying proper attention to the road, or is agitated, or sleepy, or distracted.[15] The car might then be primed to switch to a more autonomous mode, or to change the internal environment (temperature, lighting, sound) to wake up or calm the driver. Clearly this could have a potentially major positive impact on safety. Car monitoring systems might also learn a particular driver's "style" so that switching between auto and manual modes is as smooth as possible: the car could mimic the driver's style as it comes out of auto mode, thus making the transition seamless and easier for the driver to take over. The car might also learn to adjust its own autonomous driving style, within safety limits, to best suit the car's passengers. Or it might detect sleepy or boisterous children in the back and adjust conditions/provide distraction in that area accordingly. Features like this could make the whole in-car experience more pleasurable and might be one of the things that accelerates acceptance of autonomous and semi-autonomous vehicles.

Other types of autonomous robots, such as the patrolling security bots we met in the opening chapters, or some robot vacuum cleaners, use very similar AI control methods to those outlined above for autonomous vehicles. Because they move slowly and in very constrained environments, their control does not need to be quite as complex; as long as they avoid obstacles and slow down as they approach people and have appropriate safety cut-offs, there is unlikely to be an issue. But autonomous vehicles are potentially lethal devices; safety is paramount.

Autonomous vehicles exist and have been successfully tested over many millions of miles of driving, albeit in restricted conditions and often with human safety drivers in place. For instance, at the end of 2018 Waymo started a very limited driverless taxi service in Phoenix, Arizona.[16] But there are still many challenges to be met before autonomous vehicles become widespread, and before they can operate over all driving conditions and on most roads. But more of that later.

## *Is it a good idea to copy biology?*

In many cases, yes. A good example is locomotion in walking robots (which, as we’ve seen, can be very useful in rough terrain). In this instance there is a lot we can learn from biology. Hundreds of millions of years of evolution have provided us with many wonderful examples of how to walk efficiently and smoothly. This includes the biomechanics—superb designs for legs and joints and how to apply forces to them—as well as the neural control architectures and how these integrate with the subtleties of the biomechanics.

This kind of robot locomotion, which involves the coordination of multiple actuators (several for each leg), is not an easy matter. Traditional methods, often based around a central control system using rules governing the relative timing of individual leg movements,[17] usually result in rather unresponsive, “clunky” motion involving limited gait patterns. Insects, on the other hand, display a wide range of gaits and robust locomotion behaviors, including quickly recovering from damaged (or even lost) limbs—all this despite rather limited neural resources. From the mid-1980s onward, insects thus became an important source of inspiration for roboticists interested in walking machines. Rodney Brooks produced a distributed dynamical network controller for Ghengis, an early example of an insect-inspired walking robot.[18] Another influential strand of work originating in the late 1980s was the neural architecture introduced by Randy Beer and colleagues to control walking in a hexapod robot.[19] This wonderful example of a distributed architecture was based on studies of the neural circuits used in cockroach walking. Generalizations and extensions of it have been much used ever since.

The key lesson from biology that influenced these designs is the following. Rather than having one centralized controller that commands the legs in a master/slave fashion, animals distribute control across the physical characteristics of the legs themselves, neural circuitry local to each leg, and networks that interconnect these local leg controllers. In a distributed controller, stable, efficient gaits arise from cooperative interactions between the many different components making up the whole system.

Of course ANNs in general, including deep learning architectures—staples of many modern intelligent robot controllers—have their origins in biological inspiration.

Another interesting methodology to have come from biology is evolutionary robotics.[20] The idea was first proposed in 1950 by Alan Turing.[21] He suggested that worthwhile intelligent machines should be adaptive and should learn and develop, but conceded that designing, building, and programming such machines by hand will be difficult. He went on to sketch an alternative way of creating machines based on an artificial analog of biological evolution. Each machine would have hereditary material encoding its structure (artificial genes), mutated copies of which would form offspring machines. A selection mechanism would be used to favor better-adapted machines—in this case, those that learned to behave most intelligently. It was more than 40 years before Turing's long-forgotten suggestions became reality.

From the outset, the vast majority of work in this area, which started in the late 1980s, has involved populations of artificial genomes (sets of genes represented as lists of characters and numbers) encoding the structure and other properties of artificial neural networks that are used to control autonomous mobile robots required to carry out a particular task or to exhibit some set of behaviors. Other properties of the robot, such as sensor layout or body morphology, may also be under genetic control. The genomes are mutated and interbred, creating new generations of robots according to a Darwinian scheme in which the fittest individuals are most likely to produce offspring. Fitness is measured in terms of how well a robot behaves according to some evaluation criteria; this is usually automatically measured, but may, in the manner of eighteenth-century pig breeders, and in keeping with Turing's original proposal, be based on the experimenters' direct judgment. This method has been used to automatically design novel, often very concise, controllers for many different, but mainly quite simple, behaviors. It can also be used to fine tune and improve existing designs, which can be very fruitful. It has close parallels to other forms of machine learning such as reinforcement learning.

Hod Lipson and Jordan Pollack, working at Brandeis University, Massachusetts, pushed the idea further with ground-breaking work on fully evolvable robot hardware, where entire autonomous “creatures” (body and neural controllers) were evolved in simulation out of basic building blocks (neurons, bars, actuators). The fittest individuals were then automatically fabricated using 3-D printing, thus achieving autonomy of design and construction.[22]

As well as walking, insects have inspired other robot behaviors. In marked contrast to most current artificial systems, insects learn to visually navigate around complex environments in remarkably few trials, and use vision to perform many rapid and intricate maneuvers. Given their relatively tiny neural resources, they must make use of exquisitely clever innate behaviors crafted by evolution, along with ultra-efficient processing methods. Studies of insect behavior and their associated neural mechanisms have started to reveal some details of the cunning strategies involved, thus uncovering a potentially rich seam of inspiration for highly efficient, yet robust, robot algorithms. Such methods could be very useful for robot navigation in certain applications (e.g., agriculture) in areas where there is limited or no GPS coverage and expensive, detailed LIDAR mapping is infeasible, or in applications where computational processing/power/weight must be minimized (e.g., in miniature flying robots or space robotics).

An example is a radically different approach to navigation based on a novel model of ant navigation focusing on scene familiarity.[23] It comes from the insight that for an ant, movement and viewing direction are inextricably linked, due to constraints on head articulation relative to body, meaning that a familiar view specifies a familiar direction of movement. Since the views experienced along a habitual route will be more familiar, route navigation can be re-cast as a search for familiar views. This search can be performed with simple scanning, a behavior that ants have been observed to perform. A (learning) neural network model of this idea proved successful at explaining biological data and has now been applied to navigation in various robots, demonstrating that visually guided routes can be learned with incredibly parsimonious mechanisms that do not specify when or what to learn, nor separate routes into sequences of waypoints.

## *How reliable are current robots?*

Reliability is good, but not perfect. Robots developed in research labs are often a little unreliable—adjustments and repairs are occasionally needed—but that’s not the point; they are there to develop and test new theories. Commercial robots, however, need to be reliable. Exactly how reliable depends on the task they are designed to perform. The odd minor blip in a robot vacuum cleaner is unlikely to be much of an issue, but a major failure in an autonomous vehicle could lead to the loss of many lives.

There are two main aspects to robot reliability. The first is reliability of the electronic and mechanical hardware, which includes the sensors and actuators, the control system electronics and computers. High-spec quality components should be used to guard against failures; this is especially important in robots that interact with, and could potentially harm, humans. The overall integrity of the hardware design must be thoroughly tested and verified. In general, electronics have become much more reliable over recent years, but they are still prone to occasional and sudden faults. Therefore safety-critical circuitry needs to include fail-safe methods such as built-in redundancy and self-checking electronics that can initiate a graceful glide into a secure mode. Such additions greatly increase the cost and complexity of the hardware and are very difficult to make completely watertight, especially for large circuits. The automotive sector already has strict regulations and international rules about reliability and safety (but still many components fail), and these will need to be tightened as autonomous vehicles arrive. But other, as yet almost unregulated, areas of robotics probably do not have the same level of awareness of the fundamental issue of electronics reliability. Many assume it is a solved problem—it is not.[24]

The second aspect is reliability and robustness of the control algorithms. Will the robot always do the right thing? Will there be circumstances in which it does something dangerous? Again, how crucial this aspect of reliability is will depend on the task. Thorough testing and analysis of the properties of the control system are major parts of the attack on this issue. This is an area where ANNs have a potential weakness. It is often difficult to work out exactly what is going on inside a trained neural network, and therefore it is hard to examine and interpret the minutiae of how certain decisions were reached. But when an autonomous car's controller does the wrong thing, we need to know why, so it can be fixed. Hence there is currently a big drive for explainable AI systems, particularly ANN-based systems. But in truth, the explainability and exhaustive testing problems are faced to some extent by all control methods. Training and adapting, which involve exposure to vast amounts of data from a huge variety of scenarios, including the difficult, unexpected, bizarre ones (so-called edge cases), are an integral part of machine-learning approaches. If there are enough data—and in the case of autonomous vehicles, Waymo, Tesla, and the rest have collected vast oceans of the stuff—this perceived weakness could possibly become a strength. These systems will probably be the most thoroughly tested—over millions and millions of hours of training and live operation—in the history of technology. Of course, the training/test data need to be of the right quality, and the verification process (testing out in live driving scenarios) must be extensive. Testing/verification cannot have hidden biases and so must be as varied and exhaustive as possible, with numerous edge cases covered. But none of this makes the explainability problem go away; that remains a major challenge for machine learning in safety-critical applications.

Commercial pressures are forcing robots to become reliable—who is going to buy something that regularly fails or doesn’t perform as claimed? But complete perfection is not possible with any technology. Electronic and mechanical failures will always occur, and one-in-a-billion freak circumstances that confuse the control system, no matter how well tested and designed, are probably inevitable.

## *How intelligent are they, really?*

Robots are undoubtedly getting smarter. Driving a car full of people around urban streets while obeying all traffic rules and not endangering anyone—inside or outside the vehicle—is impressive, and something that most of us would agree requires a degree of intelligence. However, even control of an autonomous car is a very specialized, constrained behavior. The cases that currently work are even more constrained (limited, very heavily mapped urban areas in benign traffic and good weather conditions). The general case—of being able to autonomously drive anywhere in any conditions, with or without a detailed map—is probably still a long way off. Being able to make safe, sensible decisions while approaching a complex crowded junction full of cars and pedestrians during a driving rain storm after the traffic lights have failed, and in the midst of a full-scale riot, requires a more general kind of intelligence that can extrapolate and improvise. How long it will be—if ever—before robots are capable of such behavior is an open question. There is no evidence that it is just around the corner, as some voices, more hopeful than informed, would claim.

Self-improving, adaptive, generalizing capabilities are an important part of the higher robot intelligence we are striving for. Relevant skills include transferring something learnt in one situation to another different situation, or being able to generate novel strategies for problem solving. AI techniques for such qualities are developing, but it is impossible to say when they will be mature. It could easily be many decades. There are still quite basic issues that are very challenging (e.g., perception in foul weather for outdoor robots; decision making in unfamiliar, unstructured situations, or learning from very few examples, for all robots).

At the moment the ways in which humans can communicate with most robots are still quite limited. But AI-based voice recognition and interpretation has come a long way in the past few years (mainly thanks to machine-learning methods), and so robot voice command and response is a potentially powerful route that is being explored with some vigor. Anyone who has experience of the various "home assistant" smart speaker devices knows that parsing and comprehension of spoken language is pretty good, if far from perfect. These devices are connected to the cloud, where huge computing resources are used to power the AI behind them (including large neural networks). A similar approach would work for robots that were permanently connected to the cloud. But in some safety-critical cases where a dropped wireless connection could not be risked, a probably more limited self-contained on-board system would have to be used, and, if lives are potentially at risk, it would have to be close to perfect. Of course there are numerous other ways humans communicate, including reading body language and facial expressions, and through gestures. Research into providing robots with similar capabilities has been ongoing for some time and is making progress.[25]

So, for the moment, robot intelligence remains very narrow and very task specific. It is limited but has got to a stage where it is genuinely useful. This chapter has mainly focused on driverless cars as an exemplar of autonomous robots; we'll explore the state-of-the-art in some other areas in later chapters, including unnerving possibilities being pursued by the military. Until then it is worth noting that many of the most impressive examples developed to date have involved huge teams of engineers and vast amounts of training data, only possible through enormous corporate financial resources—but exactly how deep those pockets are is unclear. Robot AI will continue to develop, but trying to predict how fast and how far is largely futile. Anyone who tells you otherwise should be treated with great skepticism. The methods that will enable less narrow, more flexible intelligence might be very far from what we have now, may not even have been imagined yet.

But what it might be like to interact with such robotic intelligence has been imagined—in films, literature, plays, and TV shows. Depictions of robots in popular culture can be a powerful way of illustrating and thinking about the potential opportunities and dangers of robot futures. Chapter 5 begins to explore this theme.