1 Introduction

The term robot means different things to different people. Science fiction books and movies have strongly influenced what many people expect a robot to be or what it can do. Sadly the practice of robotics is far behind this popular conception. One thing is certain though – robotics will be an important technology in this century. Products such as vacuum cleaning robots have already been with us for over a decade and self-driving cars are coming. These are the vanguard of a wave of smart machines that will appear in our homes and workplaces in the near to medium future.

In the eighteenth century the people of Europe were fascinated by automata such as Vaucanson's duck shown in Fig. 1.1a. These machines, complex by the standards of the day, demonstrated what then seemed *life-like* behavior. The duck used a cam mechanism to sequence its movements and Vaucanson went on to explore mechanization of silk weaving. Jacquard extended these ideas and developed a loom, shown in Fig. 1.1b, that was essentially a programmable weaving machine. The pattern to be woven was encoded as a series of holes on punched cards. This machine has many hallmarks of a modern robot: it performed a physical task and was reprogrammable.

The term robot first appeared in a 1920 Czech science fiction play "Rossum's Universal Robots" by Karel Čapek (pronounced Chapek). The term was coined by his brother Josef, and in the Czech language means serf labor but colloquially means hardwork or drudgery. The robots in the play were artificial people or androids and as in so many robot stories that follow this one, the robots rebel and it ends badly for humanity. Isaac Asimov's robot series, comprising many books and short stories written between 1950 and 1985, explored issues of human and robot interaction and morality. The robots in these stories are equipped with "positronic brains" in which the "Three laws of robotics" are encoded. These stories have influenced subsequent books and movies which in turn have shaped the public perception of what robots are. The mid twentieth century also saw the advent of the field of *cybernetics* – an uncommon term today but then an exciting science at the frontiers of understanding life and creating intelligent machines.

The first patent for what we would now consider a robot was filed in 1954 by George C. Devol and issued in 1961. The device comprised a mechanical arm with

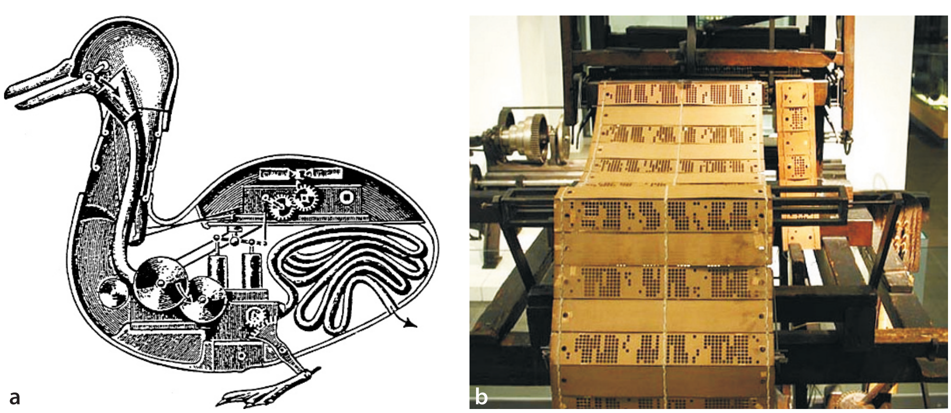


Fig. 1.1. Early programmable machines. **a** Vaucanson's duck (1739) was an automaton that could flap its wings, eat grain and defecate. It was driven by a clockwork mechanism and executed a single program; **b** The Jacquard loom (1801) was a reprogrammable machine and the program was held on punched cards (photograph by George P. Landow from www.victorianweb.org)

a gripper that was mounted on a track and the sequence of motions was encoded as magnetic patterns stored on a rotating drum. The first robotics company, Unimation, was founded by Devol and Joseph Engelberger in 1956 and their first industrial robot shown in Fig. 1.2 was installed in 1961. The original vision of Devol and Engelberger for robotic automation has become a reality and many millions of arm-type robots such as shown in Fig. 1.3 have been built and put to work at tasks such as welding, painting, machine loading and unloading, electronic assembly, packaging and palletizing. The use of robots has led to increased productivity and improved product quality. Today many products we buy have been assembled or handled by a robot.

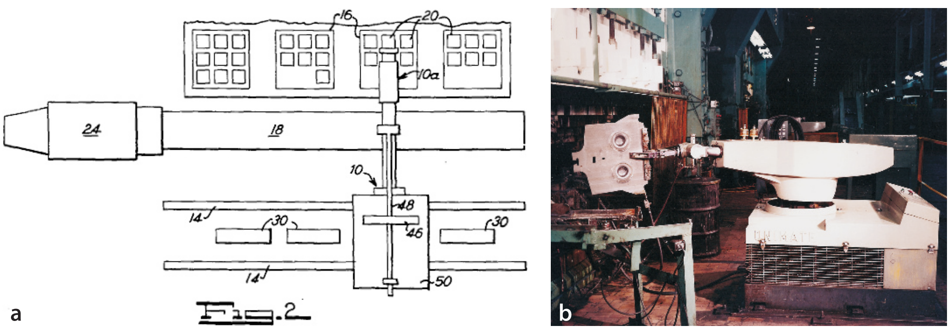


Fig. 1.2.
Universal automation. **a** A plan view of the machine from Devol's patent; **b** the first Unimation robot working at a General Motors factory (photo courtesy of George C. Devol)

Unimation Inc. (1956–1982). Devol sought financing to develop his unimation technology and at a cocktail party in 1954 he met Joseph Engelberger who was then an engineer with Manning, Maxwell and Moore. In 1956 they jointly established Unimation, the first robotics company, in Danbury Connecticut. The company was acquired by Consolidated Diesel Corp. (Condec) and became Unimate Inc. a division of Condec. Their first robot went to work in 1961 at a General Motors die-casting plant in New Jersey. In 1968 they licensed technology to Kawasaki Heavy Industries which produced the first Japanese industrial robot. Engelberger served as chief executive until it was acquired by Westinghouse in 1982. People and technologies from this company have gone on to be very influential on the whole field of robotics.

George C. Devol, Jr. (1912–2011) was a prolific American inventor. He was born in Louisville, Kentucky, and in 1932 founded United Cinephone Corp. which manufactured phonograph arms and amplifiers, registration controls for printing presses and packaging machines. In 1954, he applied for US patent 2,988,237 for Programmed Article Transfer which introduced the concept of Universal Automation or "Unimation". Specifically it described a track-mounted polar-coordinate arm mechanism with a gripper and a programmable controller – the precursor of all modern robots.

In 2011 he was inducted into the National Inventors Hall of Fame. (Photo on the right: courtesy of George C. Devol)

Joseph F. Engelberger (1925–2015) was an American engineer and entrepreneur who is often referred to as the "Father of Robotics". He received his B.S. and M.S. degrees in physics from Columbia University, in 1946 and 1949, respectively. Engelberger has been a tireless promoter of robotics. In 1966, he appeared on *The Tonight Show Starring Johnny Carson* with a Unimate robot which poured a beer, putted a golf ball, and directed the band. He promoted robotics heavily in Japan, which led to strong investment and development of robotic technology in that country.

Engelberger served as chief executive of Unimation until 1982, and in 1984 founded Transitions Research Corporation which became HelpMate Robotics Inc., an early entrant in the hospital service robot sector. He was elected to the National Academy of Engineering, received the Beckman Award and the Japan Prize, and has written two books: Robotics in Practice (1980) and Robotics in Service (1989). Each year the Robotics Industries Association presents an award in his honor to "persons who have contributed outstandingly to the furtherance of the science and practice of robotics."

These first generation robots are fixed in place and cannot move about the factory – they are not mobile. By contrast mobile robots as shown in Figs. 1.4 and 1.5 can move through the world using various forms of mobility. They can locomote over the ground using wheels or legs, fly through the air using fixed wings or multiple rotors, move through the water or sail over it. An alternative taxonomy is based on the function that the robot performs. *Manufacturing* robots operate in factories and are the technological descendents of the first generation robots. *Service robots* supply services to people such as cleaning, personal care, medical rehabilitation or fetching and carrying as shown in Fig. 1.5b. *Field robots*, such as those shown in Fig. 1.4, work outdoors on tasks such as environmental monitoring, agriculture, mining, construction and forestry. *Humanoid robots* such as shown in Fig. 1.6 have the physical form of a human being – they are both mobile robots and service robots.◂

In practice the categorization of robots is not very consistently applied.

Fig. 1.3. Manufacturing robots, technological descendants of the Unimate shown in Fig. 1.2. **a** A modern six-axis robot designed for high accuracy and throughput (image courtesy ABB robotics); **b** Baxter two-armed robot with built in vision capability and programmable by demonstration, designed for moderate throughput piece work (image courtesy Rethink Robotics)

Rossum's Universal Robots (RUR). In the introductory scene Helena Glory is visiting Harry Domin the director general of Rossum's Universal Robots and his robotic secretary Sulla.

Domin Sulla, let Miss Glory have a look at you.
Helena (stands and offers her hand) Pleased to meet you. It must be very hard for you out here, cut off from the rest of the world [the factory is on an island]
Sulla I do not know the rest of the world Miss Glory. Please sit down.
Helena (sits) Where are you from?
Sulla From here, the factory
Helena Oh, you were born here.
Sulla Yes I was made here.
Helena (startled) What?
Domin (laughing) Sulla isn't a person, Miss Glory, she's a robot.
Helena Oh, please forgive me …

The full play can be found at http://ebooks.adelaide.edu.au/c/capek/karel/rur. (Image on the left: Library of Congress item 96524672)

A manufacturing robot is typically an arm-type manipulator on a fixed base such as Fig. 1.3a that performs repetitive tasks within a local work cell. Parts are presented to the robot in an orderly fashion which maximizes the advantage of the robot's high speed and precision. High-speed robots are hazardous and safety is achieved by excluding people from robotic work places, typically placing the robot inside a cage. In contrast the Baxter robot shown in Fig. 1.3b is human safe, it operates at low speed and stops moving if it encounters an obstruction.

Field and service robots face specific and significant challenges. The first challenge is that the robot must operate and move in a complex, cluttered and changing environment. A delivery robot in a hospital must operate despite crowds of people and a time-varying configuration of parked carts and trolleys. A Mars rover as shown in Fig. 1.5a must navigate rocks and small craters despite not having an accurate local map in advance of its travel. Robotic, or self-driving cars, such as shown in Fig. 1.5c, must follow roads, avoid obstacles and obey traffic signals and the rules of the road. The second challenge for these types of robots is that they must operate safely in the presence of people. The hospital delivery robot operates among people, the robotic car contains people and a robotic surgical device operates *inside* people.

Fig. 1.4. Non-land-based mobile robots. **a** Small autonomous underwater vehicle (Todd Walsh © 2013 MBARI); **b** Global Hawk unmanned aerial vehicle (UAV) (photo courtesy of NASA)

Cybernetics, artificial intelligence and robotics. Cybernetics flourished as a research field from the 1930s until the 1960s and was fueled by a heady mix of new ideas and results from neurology, control theory and information theory. Research in neurology had shown that the brain was an electrical network of neurons. Harold Black, Henrik Bode and Harry Nyquist at Bell Labs were researching negative feedback and the stability of electrical networks, Claude Shannon's information theory described digital signals, and Alan Turing was exploring the fundamentals of computation. Walter Pitts and Warren McCulloch proposed an artificial neuron in 1943 and showed how it might perform simple logical functions. In 1951 Marvin Minsky built SNARC (from a B24 autopilot and comprising 3 000 vacuum tubes) which was perhaps the first neural-network-based learning machine as his graduate project. William Grey Walter's robotic tortoises showed life-like behavior. Maybe an electronic brain could be built!

An important early book was Norbert Wiener's *Cybernetics or Control and Communication in the Animal and the Machine* (Wiener 1965). A characteristic of a cybernetic system is the use of feedback which is common in engineering and biological systems. The ideas were later applied to evolutionary biology, psychology and economics.

In 1956 a watershed conference was hosted by John McCarthy at Dartmouth College and attended by Minsky, Shannon, Herbert Simon, Allen Newell and others. This meeting defined the term artificial intelligence (AI) as we know it today with an emphasis on digital computers and symbolic manipulation and led to new research in robotics, vision, natural language, semantics and reasoning. McCarthy and Minsky formed the AI group at MIT, and McCarthy left in 1962 to form the Stanford AI Laboratory. Minsky focused on artificially simple "blocks world". Simon, and his student Newell, were influential in AI research at Carnegie-Mellon University from which the Robotics Institute was spawned in 1979. These AI groups were to be very influential in the development of robotics and computer vision in the USA. Societies and publications focusing on cybernetics are still active today.

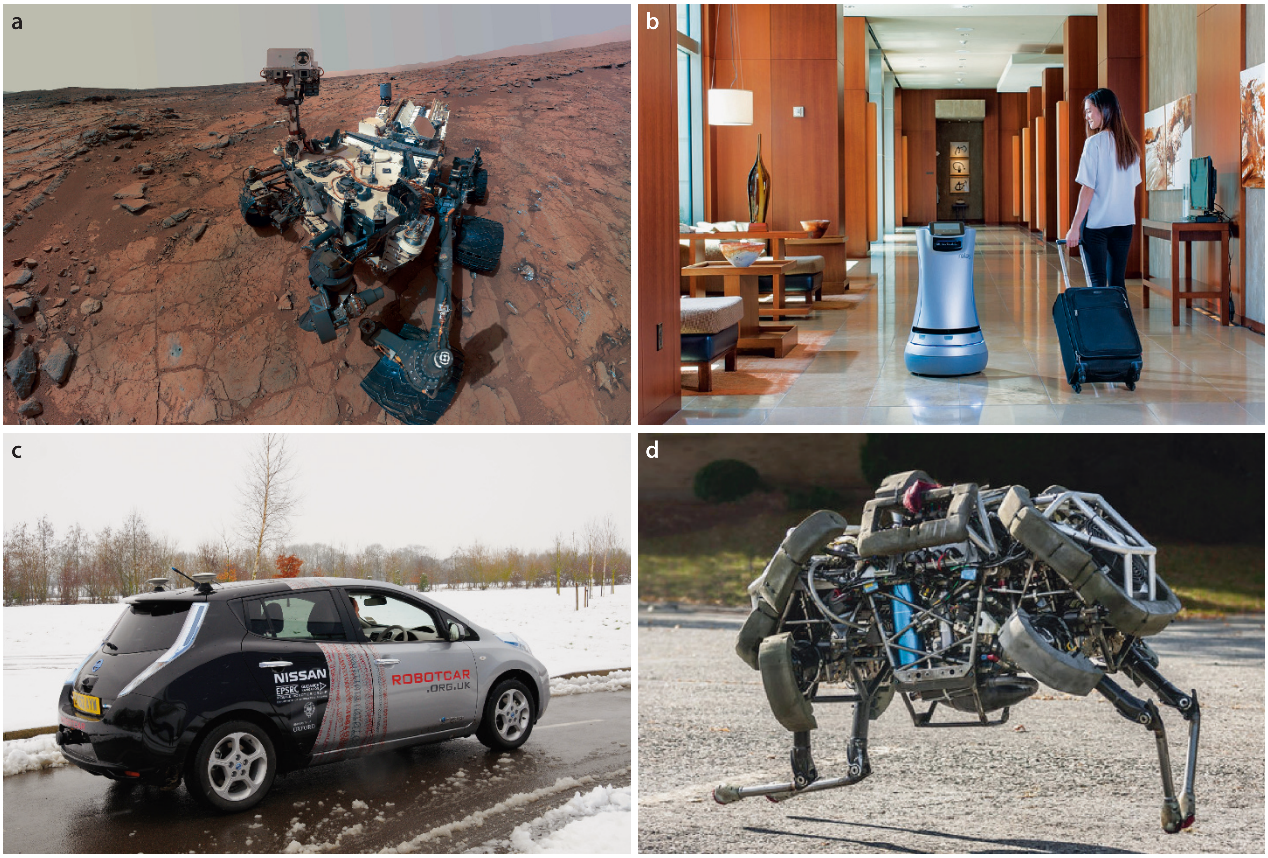


Fig. 1.5. Mobile robots. **a** Mars Science Lander, Curiosity, self portrait taken at "John Klein". The mast contains many cameras including two stereo camera pairs from which the robot can compute the 3-dimensional structure of its environment (image courtesy of NASA/JPL-Caltech/MSSS); **b** Savioke Relay delivery robot (image courtesy Savioke); **c** self driving car (image courtesy Dept. Information Engineering, Oxford Univ.); **d** Cheetah legged robot (image courtesy Boston Dynamics)

So what is a robot? There are many definitions and not all of them are particularly helpful. A definition that will serve us well in this book is

a goal oriented machine that can sense, plan and act.

A robot *senses* its environment and uses that information, together with a goal, to *plan* some *action*. The action might be to move the tool of an arm-robot to grasp an object or it might be to drive a mobile robot to some place.

Sensing is critical to robots. Proprioceptive sensors measure the state of the robot itself: the angle of the joints on a robot arm, the number of wheel revolutions on a mobile robot or the current drawn by an electric motor. Exteroceptive sensors measure the state of the world with respect to the robot. The sensor might be a simple bump sensor on a robot vacuum cleaner to detect collision. It might be a GPS receiver that measures distances to an orbiting satellite constellation, or a compass that measures the direction of the Earth's magnetic field vector relative to the robot. It might also be an active sensor that emits acoustic, optical or radio pulses in order to measure the distance to points in the world based on the time taken for a reflection to return to the sensor.

A camera is a passive device that captures patterns of optical energy reflected from the scene. Our own experience is that eyes are a very effective sensor for recognition, navigation, obstacle avoidance and manipulation so vision has long been of interest to robotics researchers. An important limitation of a single camera, or a single eye, is that the 3-dimensional structure of the scene is lost in the resulting 2-dimensional image. Despite this, humans are particularly good at inferring the 3-dimensional nature of a scene using a number of visual cues. Robots are currently not as well developed. Figure 1.7 shows some very early work on reconstructing a 3-dimensional wireframe model from a single

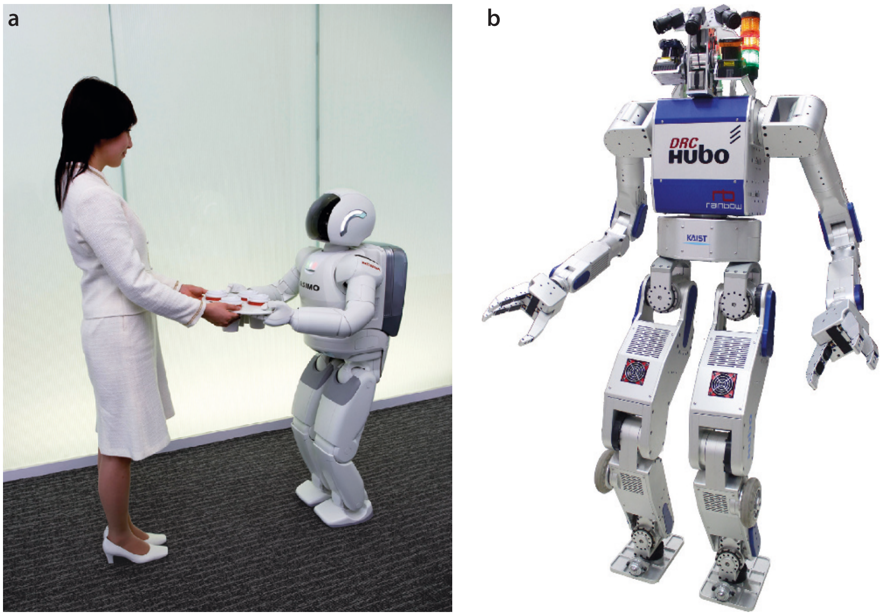


Fig. 1.6. Humanoid robots. **a** Honda's Asimo humanoid robot (image courtesy Honda Motor Co. Japan); **b** Hubo robot that won the DARPA Robotics Challenge in 2015 (image courtesy KAIST, Korea)

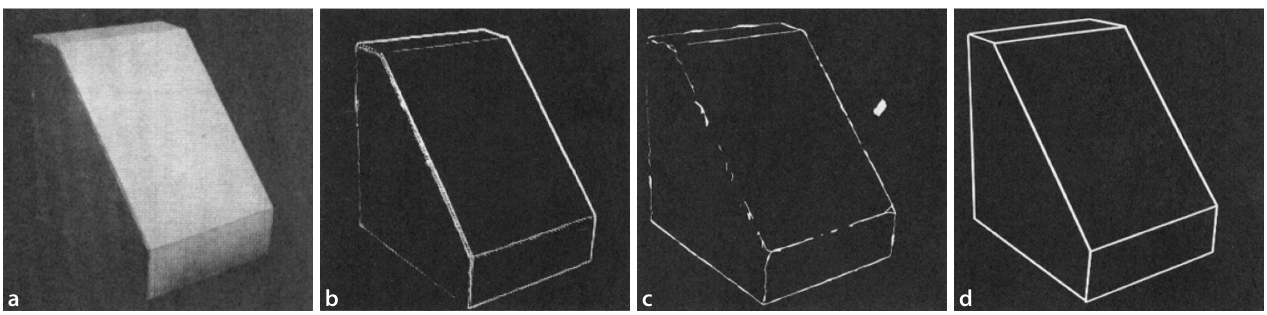


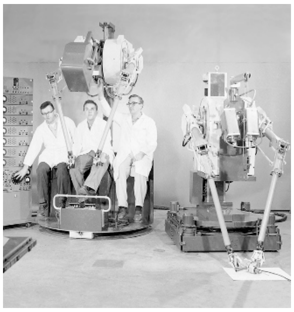
Fig. 1.7. Early results in computer vision for estimating the shape and pose of objects, from the Ph.D. work of L.G. Roberts at MIT Lincoln Lab in 1963 (Roberts 1963). **a** Original picture; **b** gradient image; **c** connected feature points; **d** reconstructed line drawing

2-dimensional image and gives some idea of the difficulties involved. Another approach is stereo vision where information from two cameras is combined to estimate the 3-dimensional structure of the scene – this is a technique used by humans and robots, for example, the Mars rover shown in Fig. 1.5a has a stereo camera on its mast.

In this book we focus on the use of cameras as sensors for robots. Machine vision, discussed in Part IV, is the use of computers to process images from one or more cameras and to extract numerical features. For example determining the coordinate of a round red object in the scene, or how far a robot has moved based on how the world appears to have moved relative to the robot.

If the robot's environment is unchanging it can make do with an accurate map and have little need to sense the state of the world, apart from determining where it is. Imagine driving a car with the front window covered over and just looking at the GPS navigation system. If you had the road to yourself you could probably drive from A to B quite successfully albeit slowly. However if there were other cars, pedestrians, traffic signals or roadworks then you would be in some difficulty. To deal with this you need to look outwards – to sense the world and plan your actions accordingly. For humans this is easy, done without conscious thought, but it is not yet easy to program a machine to do the same – this is the challenge of *robotic vision*.

Telerobots are robot-like machines that are remotely controlled by a human operator. Perhaps the earliest was a radio controlled boat demonstrated by Nikola Tesla in 1898 and which he called a teleautomaton. According to the definition above these are not robots but they were an important precursor to robots and are still important today for many tasks where people cannot work but which are too complex for a machine to per-



The Manhattan Project in World War 2 (WW II) developed the first nuclear weapons and this required handling of radioactive material. Remotely controlled arms were developed by Ray Goertz at Argonne National Laboratory to exploit the manual dexterity of human operators while keeping them away from the hazards of the material they were handling. The operators viewed the work space through thick lead-glass windows or via a television link and manipulated the master arm (on the left). The slave arm (on the right) followed the motion, and forces felt by the slave arm were reflected back to the master arm, allowing the operator to feel weight and interference force. Telerobotics is still important today for many tasks where people cannot work but which are too complex for a machine to perform by itself, for instance the underwater robots that surveyed the wreck of the Titanic. (Photo on the left: Courtesy Argonne National Laboratory)

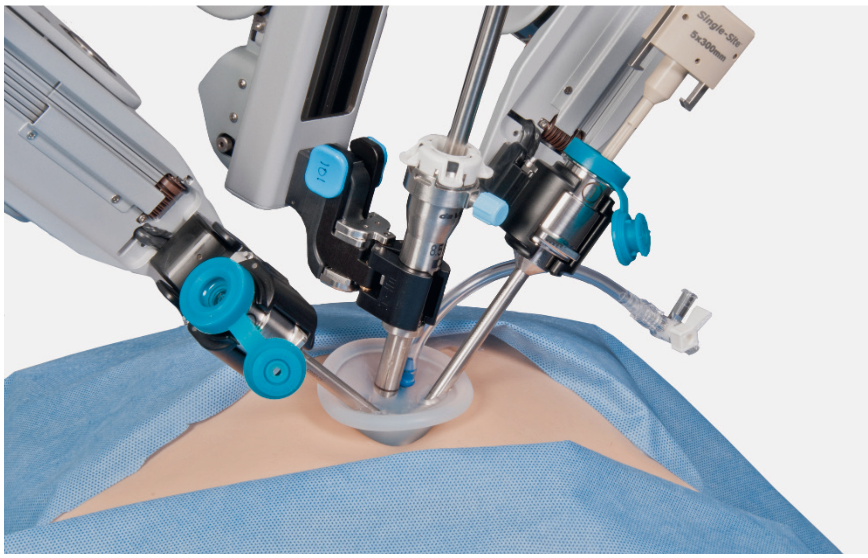


Fig. 1.8. The working end of a surgical robot, multiple tools working through a single small incision (image © 2015 Intuitive Surgical, Inc)

form by itself. For example the underwater robots that surveyed the wreck of the Titanic were technically remotely operated vehicles (ROVs). A modern surgical robot as shown in Fig. 1.8 is also teleoperated – the motion of the small tools are remotely controlled by the surgeon and this makes it possible to use much smaller incisions than the old-fashioned approach where the surgeon works inside the body with their hands.

The various Mars rovers autonomously navigate the surface of Mars but human operators provide the high-level goals. That is, the operators tell the robot where to go and the robot itself determines the details of the route. Local decision making on Mars is essential given that the communications delay is several minutes. Some robots are hybrids and the control task is shared or traded with a human operator. In traded control, the control function is passed back and forth between the human operator and the computer. For example an aircraft pilot can pass control to an autopilot and take control back. In shared control, the control function is performed by the human operator and the computer working together. For example an autonomous passenger car might have the computer keeping the car safely in the lane while the human driver just controls speed.

1.1 Robots, Jobs and Ethics

A number of ethical issues arise from the advent of robotics. Perhaps the greatest concern to the wider public is "robots taking jobs from people". This is a complex issue but we cannot shy away from the fact that many jobs now done by people will, in the future, be performed by robots. Clearly there are dangerous jobs which people should not do, for example handling hazardous substances or working in dangerous environments. There are many low-skilled jobs where human labor is increasingly hard to source, for instance in jobs like fruit picking. In many developed countries people no longer aspire to hard physical outdoor work in remote locations. What are the alternatives if people don't want to do the work? In areas like manufacturing, particularly car manufacturing, the adoption of robotic automation has been critical in raising productivity which has allowed that industry to be economically viable in high-wage countries like Europe, Japan and the USA. Without robots these industries could not exist; they would not employ any people, not pay any taxes, and not consume products and services from other parts of the economy. Automated industry might employ fewer people but it still makes an important contribution to society. Rather than taking jobs we could argue that robotics and automation has helped to keep manufacturing industries viable in high-labor cost countries. How do we balance the good of the society with the good of the individual?

There are other issues besides jobs. Consider self-driving cars. We are surprisingly accepting of manually driven cars even though they kill more than one million people every year, yet many are uncomfortable with the idea of self-driving cars even though they will dramatically reduce this loss of life. We worry about who to blame if a robotic car makes a mistake while the carnage caused by human drivers continues. Similar concerns are raised when talking about robotic healthcare and surgery – human surgeons are not perfect but robots are seemingly held to a much higher account. There is a lot of talk about using robots to look after elderly people, but does this detract from their quality of life by removing human contact, conversation and companionship? Should we use robots to look after our children, and even teach them? What do we think of armies of robots fighting and killing human beings?

Robotic cars, health care, elder care and child care might bring economic benefits to our society but is it the right thing to do? Is it a direction that we want our society to go? Once again how do we balance the good of the society with the good of the individual? These are deep ethical questions that cannot and should not be decided by roboticists alone. But neither should roboticists ignore them. This is a discussion for all of society and roboticists have a duty to be active participants in this debate.

1.2 About the Book

This book is about robotics and computer vision – separately, and together as robotic vision. These are big topics and the combined coverage is necessarily broad. The intent is not to be shallow but rather to give the reader a flavor of what robotics and vision is about and what it can do – consider it a grand tasting menu.

The goals of the book are:

- to provide a broad and solid base of understanding through theory and examples;
- to make abstract concepts tangible
- to tackle more complex problems than other more specialized textbooks by virtue of the powerful numerical tools and software that underpins it;
- to provide instant gratification by solving complex problems with relatively little code;
- to complement the many excellent texts in robotics and computer vision;
- to encourage intuition through hands on numerical experimentation; and
- to limit the number of equations presented to those cases where (in my judgment) they add value or clarity.

The approach used is to present background, theory and examples in an integrated fashion. Code and examples are first-class citizens in this book and are not relegated to the end of the chapter or an associated web site. The examples are woven into the discussion like this

```
>> p = transl(Ts);
>> plot(t, p);
```

where the MATLAB® code illuminates the topic being discussed and generally results in a crisp numerical result or a graph in a figure that is then discussed. The examples illustrate how to use the associated Toolboxes and that knowledge can then be applied to other problems. Most of the figures in this book have been generated by the code examples provided and they are available from the book's website as described in Appendix A.

1.2.1 MATLAB Software and the Toolboxes

To do good work, one must first have good tools.
Chinese proverb

The computational foundation of this book is MATLAB®, a software package developed by The MathWorks Inc. MATLAB is an interactive mathematical software environment that makes linear algebra, data analysis and high-quality graphics a breeze. MATLAB is a popular package and one that is very likely to be familiar to engineering students as well as researchers. It also supports a programming language which allows the creation of complex algorithms.

A strength of MATLAB is its support for Toolboxes which are collections of functions targeted at particular topics. Toolboxes are available from MathWorks, third party companies and individuals. Some Toolboxes are products to be purchased while others are open-source and generally free to use. This book is based on two open-source Toolboxes written by the author: the Robotics Toolbox for MATLAB and the Machine Vision Toolbox for MATLAB. These Toolboxes, with MATLAB, turn a personal computer into a powerful and convenient environment for investigating complex problems in robotics, machine vision and vision-based control. The Toolboxes are free to use and distributed under the GNU Lesser General Public License (GNU LGPL).

The *Robotics Toolbox* (RTB) provides a diverse range of functions for simulating mobile and arm-type robots. The Toolbox supports a very general method of representing the structure of serial-link manipulators using MATLAB objects and provides functions for forward and inverse kinematics and dynamics. The Toolbox includes functions for manipulating and converting between datatypes such as vectors, homogeneous transformations, 3-angle representations, twists and unit-quaternions which are necessary to represent 3-dimensional position and orientation. The Toolbox also includes functionality for simulating mobile robots and includes models of wheeled vehicles and quadrotors and controllers for these vehicles. It also provides standard algorithms for robot path planning, localization, map making and SLAM.

The *Machine Vision Toolbox* (MVTB) provides a rich collection of functions for camera modeling, image processing, image feature extraction, multi-view geometry and vision-based control. The MVTB also contains functions for image acquisition and

The MATLAB software we use today has a long history. It starts with the LINPACK and EISPACK projects run by the Argonne National Laboratory in the 1970s to produce high quality, tested and portable mathematical software. LINPACK is a collection of routines for linear algebra and EISPACK is a library of numerical algorithms for computing eigenvalues and eigenvectors of matrices. These packages were written in Fortran which was then the language of choice for large-scale numerical problems.

Cleve Moler, then at the University of New Mexico, contributed to both projects and wrote the first version of MATLAB in the late 1970s. It allowed interactive use of LINPACK and EISPACK for problem solving without having to write and compile Fortran code. MATLAB quickly spread to other universities and found a strong audience within the applied mathematics and engineering community. In 1984 Cleve Moler and Jack Little founded The MathWorks Inc. which exploited the newly released IBM PC – the first widely available desktop computer.

Cleve Moler received his bachelor's degree from Caltech in 1961, and a Ph.D. from Stanford University. He was a professor of mathematics and computer science at universities including University of Michigan, Stanford University, and the University of New Mexico. He has served as president of the Society for Industrial and Applied Mathematics (SIAM) and was elected to the National Academy of Engineering in 1997.

See also http://www.mathworks.com/company/aboutus/founders/clevemoler.html which includes a video of Cleve Moler and also http://history.siam.org/pdfs2/Moler_final.pdf.

display; filtering; blob, point and line feature extraction; mathematical morphology; image warping; stereo vision; homography and fundamental matrix estimation; robust estimation; bundle adjustment; visual Jacobians; geometric camera models; camera calibration and color space operations. For modest image sizes on a modern computer the processing rate can be sufficiently "real-time" to allow for closed-loop control.

If you're starting out in robotics or vision then the Toolboxes are a significant initial base of code on which to build your project. The Toolboxes are provided in source code form. The bulk of the code is written in the MATLAB M-language but a few functions are written in C▸ or Java for increased computational efficiency. In general the Toolbox code is written in a straightforward manner to facilitate understanding, perhaps at the expense of computational efficiency. Appendix A provides details of how to obtain the Toolboxes and pointers to online resources including discussion groups.

These are implemented as MEX files, which are written in C in a very specific way that allows them to be invoked from MATLAB just like a function written in M-language.

This book provides examples of how to use many Toolbox functions in the context of solving specific problems but it is not a reference manual. Comprehensive documentation of all Toolbox functions is available through the MATLAB builtin help mechanism or the PDF format manual that is distributed with each Toolbox.

1.2.2 Notation, Conventions and Organization

The mathematical notation used in the book is summarized in the Nomenclature section on page xxv. Since the coverage of the book is broad there are just not enough good symbols to go around, so it is unavoidable that some symbols have different meanings in different parts of the book.

There is a lot of MATLAB code in the book and this is indicated in blue fixed-width font such as

```
>> a = 2 + 2
a =
     4
```

The MATLAB command prompt is >> and what follows is the command issued to MATLAB by the user. Subsequent lines, without the prompt, are MATLAB's response. All functions, classes and methods mentioned in the text or in code segments are cross-referenced and have their own indexes at the end of the book allowing you to find different ways that particular functions can be used.

Colored boxes are used to indicate different types of material. Orange informational boxes highlight material that is particularly important while red and orange warning boxes highlight points that are often traps for those starting out. Blue boxes provide technical, historical or biographical information that augment the main text but they are not critical to its understanding.

As an author there is a tension between completeness, clarity and conciseness. For this reason a lot of detail has been pushed into notes▸ and blue boxes and on a first reading these can be skipped. Some chapters have an Advanced Topics section at the end that can also be skipped on a first reading. However if you are trying to understand a particular algorithm and apply it to your own problem then understanding the details and nuances can be important and the notes or advanced topics are for you.

They are placed as marginal notes near the corresponding marker.

Each chapter ends with a *Wrapping Up* section that summarizes the important lessons from the chapter, discusses some suggested further reading, and provides some exercises. For clarity, references are cited sparingly in the text of each chapter. The *Further Reading* subsection discusses prior work and references that provide more rigor or more complete description of the algorithms. *Resources* provides links to relevant online code and datasets. *MATLAB Notes* provides additional details about the author's toolboxes and those with similar functionality from MathWorks. *Exercises* extend the concepts discussed within the chapter and are generally related to specific code examples discussed in the chapter. The exercises vary in difficulty from straightforward extension of the code examples to more challenging problems.

1.2.3 Audience and Prerequisites

The book is intended primarily for third or fourth year engineering undergraduate students, Masters students and first year Ph.D. students. For undergraduates the book will serve as a companion text for a robotics or computer vision course or to support a major project in robotics or vision. Students should study Part I and the appendices for foundational concepts, and then the relevant part of the book: mobile robotics, arm robots, computer vision or vision-based control. The Toolboxes provide a solid set of tools for problem solving, and the exercises at the end of each chapter provide additional problems beyond the worked examples in the book.

For students commencing graduate study in robotics, and who have previously studied engineering or computer science, the book will help fill the gaps between what you learned as an undergraduate and what will be required to underpin your deeper study of robotics and computer vision. The book's working code base can help bootstrap your research, enabling you to get started quickly and working productively on your own problems and ideas. Since the source code is available you can reshape it to suit your need, and when the time comes (as it usually does) to code your algorithms in some other language then the Toolboxes can be used to cross-check your implementation.

For those who are no longer students, the researcher or industry practitioner, the book will serve as a useful companion for your own reference to a wide range of topics in robotics and computer vision, as well as a handbook and guide for the Toolboxes.

The book assumes undergraduate-level knowledge of linear algebra (matrices, vectors, eigenvalues), basic set theory, basic graph theory, probability, dynamics (forces, torques, inertia) and control theory. Some of these topics will likely be more familiar to engineering students than computer science students. Computer science students may struggle with some concepts in Chap. 4 and 9 such as the Laplace transform, transfer functions, linear control (proportional control, proportional-derivative control, proportional-integral control) and block diagram notation. This material could be skimmed over on a first reading and Albertos and Mareels (2010) may be a useful introduction to some of these topics. The book also assumes the reader is familiar with using and programming in MATLAB and also familiar with object-oriented programming techniques (perhaps C++, Java or Python). Familiarity with Simulink®, the graphical block-diagram modeling tool integrated with MATLAB will be helpful but not essential.

1.2.4 Learning with the Book

The best way to learn is by doing. Although the book shows the MATLAB commands and the response there is something special about doing it for yourself. Consider the book as an invitation to tinker. By running the commands yourself you can look at the results in ways that you prefer, plot the results in a different way, or try the algorithm on different data or with different parameters. The book is especially designed to stay open which enables you to type in commands as you read. You can also look at the online documentation for the Toolbox functions, discover additional features and options, and experiment with those, or read the code to see how it really works and perhaps modify it.

Most of the commands are quite short so typing them in to MATLAB is not too onerous. However the book's web site, see Appendix A, includes all the MATLAB commands shown in the book (more than 1 600 lines) and these can be cut and pasted into MATLAB or downloaded and used to create your own scripts.

In 2015 two open online courses (MOOCs) were released – based on the content and approach of this book. Introduction to Robotics covers most of Parts I and III, while Robotic Vision covers some of Parts IV and V. Each MOOC is six weeks long and comprises 12 hours of video lecture material plus quizzes, assignments and an optional project. They can be reached via http://petercorke.com/moocs.

1.2.5 Teaching with the Book

The book can be used in support of courses in robotics, mechatronics and computer vision. All courses should include the introduction to coordinate frames and their composition which is discussed in Chap. 2. For a mobile robotics or image processing course it is sufficient to teach only the 2-dimensional case. For robotic manipulators or multi-view geometry the 2- and 3-dimensional cases should be taught.

Most figures (MATLAB-generated and line drawings) in this book are available as PDF format files from the book's web site and you are free to use them with attribution in any course material that you prepare. All the code in this book can be downloaded from the web site and used as the basis for demonstrations in lectures or tutorials. See Appendix A for details.

The exercises at the end of each chapter can be used as the basis of assignments, or as examples to be worked in class or in tutorials. Most of the questions are rather open ended in order to encourage exploration and discovery of the effects of parameters and the limits of performance of algorithms. This exploration should be supported by discussion and debate about performance measures and what *best* means. True understanding of algorithms involves an appreciation of the effects of parameters, how algorithms fail and under what circumstances.

The teaching approach could also be inverted, by diving headfirst into a particular problem and then teaching the appropriate prerequisite material. Suitable problems could be chosen from the Application sections of Chap. 7, 14 or 16, or from any of the exercises. Particularly challenging exercises are so marked.

If you wanted to consider a flipped learning approach then the two MOOCs mentioned on page 11 could be used in conjunction with your class. Students would watch the videos and undertake some formative assessment out of the classroom, and you could use classroom time to work through problem sets.

For graduate level teaching the papers and textbooks mentioned in the *Further Reading* could form the basis of a student's reading list. They could also serve as candidate papers for a reading group or journal club.

1.2.6 Outline

I promised a book with instant gratification but before we can get started in robotics there are some fundamental concepts that we absolutely need to understand, and understand well. Part I introduces the concepts of pose and coordinate frames – how we represent the position and orientation of a robot, a camera or the objects that the robot needs to work with. We discuss how motion between two poses can be *decomposed* into a sequence of elementary translations and rotations, and how elementary motions can be *composed* into more complex motions. Chapter 2 discusses how pose can be represented in a computer, and Chap. 3 discusses the relationship between velocity and the derivative of pose, estimating motion from sensors and generating a sequence of poses that smoothly follow some path in space and time.

With these formalities out of the way we move on to the first main event – robots. There are two important classes of robot: mobile robots and manipulator arms and these are covered in Parts II and III respectively.▸

Although robot arms came first chronologically, mobile robotics is mostly a 2-dimensional problem and easier to understand than the 3-dimensional arm-robot case.

Part II begins, in Chap. 4, with motion models for several types of wheeled vehicles and a multi-rotor flying vehicle. Various control laws are discussed for wheeled vehicles such as moving to a point, following a path and moving to a specific pose. Chapter 5 is concerned with navigation, that is, how a robot finds a path between points A and B in the world. Two important cases, with and without a map, are discussed. Most navigation techniques require knowledge of the robot's position and Chap. 6 discusses various approaches to this problem based on dead-reckoning, or landmark observation and a map. We also show how a robot can make a map, and even determine its location while simultaneously mapping an unknown region.

Part III is concerned with arm-type robots, or more precisely serial-link manipulators. Manipulator arms are used for tasks such as assembly, welding, material handling and even surgery. Chapter 7 introduces the topic of kinematics which relates the angles of the robot's joints to the 3-dimensional pose of the robot's tool. Techniques to generate smooth paths for the tool are discussed and two examples show how an arm-robot can draw a letter on a surface and how multiple arms (acting as legs) can be used to create a model for a simple walking robot. Chapter 8 discusses the relationships between the rates of change of joint angles and tool pose. It introduces the Jacobian matrix and concepts such as singularities, manipulability, null-space motion, and resolved-rate motion control. It also discusses under- and over-actuated robots and the general numerical solution to inverse kinematics. Chapter 9 introduces the design of joint control systems, the dynamic equations of motion for a serial-link manipulator, and the relationship between joint forces and joint motion. It discusses important topics such as variation in inertia, the effect of payload, flexible transmissions and independent joint versus nonlinear control strategies.

Computer vision is a large field concerned with processing images in order to enhance them for human benefit, interpret the contents of the scene or create a 3D model corresponding to the scene. Part IV is concerned with machine vision, a subset of computer vision, and defined here as the extraction of numerical features from images to provide input for control of a robot. The discussion starts in Chap. 10 with the fundamentals of light, illumination and color. Chapter 11 describes the geometric model of perspective image creation using lenses and discusses topics such as camera calibration and pose estimation. We introduce nonperspective imaging using wide-angle lenses and mirror systems, camera arrays and light-field cameras. Chapter 12 discusses *image processing* which is a domain of 2-dimensional signal processing that transforms one image into another image. The discussion starts with acquiring real-world images and then covers various arithmetic and logical operations that can be performed on images. We then introduce spatial operators such as convolution, segmentation, morphological filtering and finally image shape and size changing. These operations underpin the discussion in Chap. 13 which describe how numerical features are extracted from images. The features describe homogeneous regions (blobs), lines or distinct points in the scene and are the basis for vision-based robot control. Chapter 14 is concerned with estimating the underlying three-dimensional geometry of a scene using classical methods such as structured lighting and also combining features found in different views of the same scene to provide information about the geometry and the spatial relationship between the camera views which is encoded in fundamental, essential and homography matrices. This leads to the topic of bundle adjustment and structure from motion and applications including perspective correction, mosaicing, image retrieval and visual odometry.

Part V discusses how visual features extracted from the camera's view can be used to control arm-type and mobile robots – an approach known as vision-based control or visual servoing. This part pulls together concepts introduced in the earlier parts of the book. Chapter 15 introduces the classical approaches to visual servoing known as position-based and image-based visual servoing and discusses their respective limitations. Chapter 16 discusses more recent approaches that address these limitations and also covers the use of nonperspective cameras, under-actuated robots and mobile robots.

This is a big book but any one of the parts can be read standalone, with more or less frequent visits to the required earlier material. Chapter 2 is the only mandatory material. Parts II, III or IV could be used respectively for an introduction to mobile robots, arm robots or computer vision class. An alternative approach, following the instant gratification theme, is to jump straight into any chapter and start exploring – visiting the earlier material as required.

Further Reading

The Handbook of Robotics (Siciliano and Khatib 2016) provides encyclopedic coverage of the field of robotics today, covering theory, technology and the different types of robot such as telerobots, service robots, field robots, flying robots, underwater robots and so on. The classic work by Sheridan (2003) discusses the spectrum of autonomy from remote control, through shared and traded control to full autonomy.

A comprehensive coverage of computer vision is the book by Szeliski (2011), and a solid introduction to artificial intelligence is the text by Russell and Norvig (2009).

A number of recent books discuss the future impacts of robotics and artificial intelligence on society, for example Ford (2015), Brynjolfsson and McAfee (2014), Bostrom (2016) and Neilson (2011). The YouTube video Grey (2014) makes some powerful points about the future of work and is always a great discussion starter.