Cyberbotics' Robot Curriculum

by Wikibooks contributors

Created on Wikibooks,
the open content textbooks collection.

Contents

Chapter 1

About this book

Learning about Intelligent Robots

This book is intended to students, teachers, hobbyists and researchers interested in intelligent robots. It will help you understanding what robots are, what they can do for you, and most interestingly how to program them. It includes two parts: a short theoretical part and a longer practical part. Practical part is decomposed in one chapter about the computer configuration and five chapters of exercises corresponding to five level of difficulty (see the next section). After reading this book, you should be able to design your own intelligent robots.

From Beginners to Robotics Experts

Even if you never wrote a computer program before, you will learn easily how to graphically program the behavior of a simple robot. From this first experience, you will be smoothly introduced to higher level computer programming and discover more possibilities of intelligent robots. This practical investigation is organized in projects for which a difficulty level is associated. You are free to stop at any level if the projects suddenly become too difficult to handle, but if you reach the latest levels successfully, you should consider yourself as a genuine robotics researcher! Here are the levels of difficulty:

- beginner: no prior knowledge needed, suitable for children from 8 years old and people without a scientific background (see Beginner programming Exercises)
- novice: scientific or technological interest needed, suitable for children from 8 years old (see Novice programming Exercises)
- intermediate: general computer science background needed, intended to student from 12 years old with some interest in computer science (see Intermediate programming Exercises)
- advanced: programming skills needed, intended to post-graduate students and researchers (see Advanced Programming Exercises)
- expert: research spirit needed, intended to post-graduate student and researchers (see Cognitive Benchmarks)

Important: The code to which we refer in the exercises is freely available at sourceforge.net. You can download it directly from the SVN at this address:

`http://robotcurriculum.svn.sourceforge.net/svnroot/robotcurriculum`

Easy-to-use robotics Tools

The practical part of this book relies on a couple of software and hardware tools that will allow you to practice intelligent robot programming for real. These tools are the e-puck robot and the Webots software. They are both widely used for education and research in Universities worldwide and are commercially available and well supported. These tools will be described in chapter E-puck and Webots.

Enjoy Robot Competitions

Several exercises are provided along this book. Starting from very simple introductory exercises in chapter Beginner programming Exercises, the reader will learn progressively how to create more and more advanced robotics controllers throughout the following chapters. Finally, the chapter Cognitive Benchmarks will introduce the reader into the realm of robot competitions through a cognitive benchmark: Rat's Life [1].

Further reading

- Cyberbotics Official Webpage
- E-puck website

[1] See their website, Rat's Life Programming Contest

Chapter 2

What is Artificial Intelligence?

Artificial Intelligence (AI) is an interdisciplinary field of study that includes computer science, engineering, philosophy and psychology. There is no widely accepted precise definition of Artificial Intelligence, because Intelligence is very difficult to define. John McCarthy defined Artificial Intelligence as "the science and engineering of making intelligent machine" [1] which does not explain what intelligent machines are. Hence, it does not help either to answer the question "Is a chess playing program an intelligent machine?".

GOFAI versus New AI

AI divides roughly into two schools of thought: GOFAI (Good Old Fashioned Artificial Intelligence) and New AI. GOFAI mostly involves methods now classified as machine learning, characterized by formalism and statistical analysis. This is also known as conventional AI, symbolic AI, logical AI or neat AI. Methods include:

- *Expert Systems* apply reasoning capabilities to reach a conclusion. An Expert System can process large amounts of known information and provide conclusions based on them.
- *Case Based Reasoning* stores a set of problems and answers in an organized data structure called cases. A Case Based Reasoning system upon being presented with a problem finds a case in its knowledge base that is most closely related to the new problem and presents its solutions as an output with suitable modifications.
- *Bayesian Networks* are probabilistic graphical models that represent a set of variables and their probabilistic dependencies.
- *Behavior Based AI* is a modular method building AI systems by hand.

New AI involves iterative development or learning. It is often bio-inspired and provides models of biological intelligence, like the Artificial Neural Networks. Learning is based on empirical data and is associated with non-symbolic AI. Methods mainly include:

[1] See John McCarthy, What is Artificial Intelligence?

- *Artificial Neural Networks* are bio-inspired systems with very strong pattern recognition capabilities.
- *Fuzzy Systems* are techniques for reasoning under uncertainty; they have been widely used in modern industrial and consumer product control systems.
- *Evolutionary computation* applies biologically inspired concepts such as populations, mutation and survival of the fittest to generate increasingly better solutions to a problem. These methods most notably divide into *Evolutionary Algorithms* (including *Genetic Algorithms*) and *Swarm Intelligence* (including *Ant Algorithms*).

Hybrid Intelligent Systems attempt to combine these two groups. Expert Inference Rules can be generated through Artificial Neural Network or Production Rules from Statistical Learning.

History

Early in the 17th century, René Descartes envisioned the bodies of animals as complex but reducible machines, thus formulating the mechanistic theory, also known as the "clockwork paradigm". Wilhelm Schickard created the first mechanical digital calculating machine in 1623, followed by machines of Blaise Pascal (1643) and Gottfried Wilhelm von Leibniz (1671), who also invented the binary system. In the 19th century, Charles Babbage and Ada Lovelace worked on programmable mechanical calculating machines.

Bertrand Russell and Alfred North Whitehead published Principia Mathematica in 1910-1913, which revolutionized formal logic. In 1931 Kurt Gödel showed that sufficiently powerful consistent formal systems contain true theorems unprovable by any theorem-proving AI that is systematically deriving all possible theorems from the axioms. In 1941 Konrad Zuse built the first working mechanical program-controlled computers. Warren McCulloch and Walter Pitts published A Logical Calculus of the Ideas Immanent in Nervous Activity (1943), laying the foundations for neural networks. Norbert Wiener's Cybernetics or Control and Communication in the Animal and the Machine (MIT Press, 1948) popularized the term "cybernetics".

Game theory which would prove invaluable in the progress of AI was introduced with the paper, Theory of Games and Economic Behavior by mathematician John von Neumann and economist Oskar Morgenstern [2].

1950's

The 1950s were a period of active efforts in AI. In 1950, Alan Turing introduced the "Turing test" as a way of creating a test of intelligent behavior. The first working AI programs were written in 1951 to run on the Ferranti Mark I machine of the University of Manchester: a checkers-playing program written by Christopher Strachey and a chess-playing program written by Dietrich Prinz. John McCarthy coined the term "artificial intelligence" at the first conference devoted to the subject, in 1956. He also invented the Lisp programming language. Joseph Weizenbaum built ELIZA, a chatter-bot implementing Rogerian psychotherapy. The birth date of AI is generally considered to be July 1956 at the Dartmouth Conference, where many of these people met and exchanged ideas.

[2] Von Neumann, J.; Morgenstern, O. (1953), "Theory of Games and Economic Behavior", New York

1960s-1970s

During the 1960s and 1970s, Joel Moses demonstrated the power of symbolic reasoning for integration problems in the Macsyma program, the first successful knowledge-based program in mathematics. Leonard Uhr and Charles Vossler published "A Pattern Recognition Program That Generates, Evaluates, and Adjusts Its Own Operators" in 1963, which described one of the first machine learning programs that could adaptively acquire and modify features and thereby overcome the limitations of simple perceptrons of Rosenblatt. Marvin Minsky and Seymour Papert published Perceptrons, which demonstrated the limits of simple Artificial Neural Networks. Alain Colmerauer developed the Prolog computer language. Ted Shortliffe demonstrated the power of rule-based systems for knowledge representation and inference in medical diagnosis and therapy in what is sometimes called the first expert system. Hans Moravec developed the first computer-controlled vehicle to autonomously negotiate cluttered obstacle courses.

1980s

In the 1980s, Artificial Neural Networks became widely used due to the back-propagation algorithm, first described by Paul Werbos in 1974. The team of Ernst Dickmanns built the first robot cars, driving up to 55 mph on empty streets.

1990s & Turn of the Millennium

The 1990s marked major achievements in many areas of AI and demonstrations of various applications. In 1995, one of Ernst Dickmanns' robot cars drove more than 1000 miles in traffic at up to 110 mph, tracking and passing other cars (simultaneously Dean Pomerleau of Carnegie Mellon tested a semi-autonomous car with human-controlled throttle and brakes). Deep Blue, a chess-playing computer, beat Garry Kasparov in a famous six-game match in 1997. Honda built the first prototypes of humanoid robots (see picture of the Asimo Robot).

During the 1990s and 2000s AI has become very influenced by probability theory and statistics. Bayesian networks are the focus of this movement, providing links to more rigorous topics in statistics and engineering such as Markov models and Kalman filters, and bridging the divide between GOFAI and New AI. This new school of AI is sometimes called 'machine learning'. The last few years have also seen a big interest in game theory applied to AI decision making.

The Turing test

Artificial Intelligence is implemented in machines (i.e., computers or robots), that are observed by "Natural Intelligence" beings (i.e., humans). These human beings are questioning whether or not these machines are intelligent. To give an answer to this question, they evidently compare the behavior of the machine to the behavior of another intelligent being they know. If both are similar, then, they can conclude that the machine appears to be intelligent.

Alan Turing developed a very interesting test that allows the observer to formally say whether or not a machine is intelligent. To understand this test, it is first necessary to understand that intelligence, just like beauty, is a concept relative to an observer. There is no absolute intelligence, like there is no absolute beauty. Hence it is not correct to say that a machine is more or less intelligent. Rather, we should say that a machine is more or less intelligent for a given observer.

Figure 2.1: Asimo: Honda's humanoid robot

Starting from this point of view, the Turing test makes it possible to evaluate whether or not a machine qualifies for artificial intelligence relatively to an observer.

The test consists in a simple setup where the observer is facing a machine. The machine could be a computer or a robot, it does not matter. The machine however, should have the possibility to be remote controlled by a human being (the remote controller) which is not visible by the observer. The remote controller may be in another room than the observer. He should be able to communicate with the observer through the machine, using the available inputs and outputs of the machine. In the case of a computer, the inputs and outputs may be a keyboard, a mouse and computer screen. In the case of a robot, it may be a camera, a speaker (with synthetic voice), a microphone, motors, etc. The observer doesn't know if the machine is remote controlled by someone else or if it behaves on its own. He has to guess it. Hence, he will interact with the machine, for example by chatting using the keyboard and the screen to try to understand whether or not there is a human intelligence behind this machine writing the answers to his questions. Hence he will want to ask very complicated questions and see what the machine answers and try to determine if the answers are generated by an AI program or if they come from a real human being. If the observer believes he is interacting with a human being while he is actually interacting with a computer program, then this means the machine is intelligent for him. He was bluffed by the machine. The table below summarizes all the possible results coming out of a Turing test.

The Turing test helps a lot to answer the question “can we build intelligent machines?”. It demonstrates that some machines are indeed already intelligent for some people. Although these people are currently a minority, including mostly children but also adults, this minority is growing as AI programs improve.

Although the original Turing test is often described as a computer chat session (see picture), the interaction between the observer and the machine may take very various forms, including a chess game, playing a virtual reality video game, interacting with a mobile robot, etc.

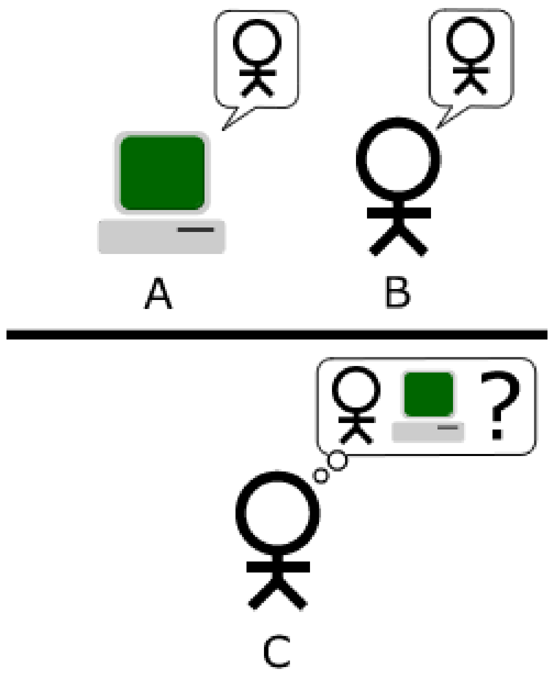

Figure 2.2: The Turing test

Similar experiments involve children observing two mobile robots performing a prey predator game and describing what is happening. Unlike adults who will generally say that the robots were programmed in some way to perform this behavior, possibly mentioning the sensors, actuators and micro-processor of the robot, the children will describe the behavior of the robots using the same words they would use to describe the behavior of a cat running after a mouse. They will grant feelings to the robots like ”he is afraid of”, ”he is angry”, ”he is excited”, ”he is quiet”, ”he wants to...”, etc. This leads us to think that for a child, there is little difference between the intelligence of such robots and animal intelligence.

	The machine is remote controlled by a human	**The machine runs an Artificial Intelligence program**
The observer believes he faces a human intelligence	**undetermined:** the observer is good at recognizing human intelligence	**successful**: the machine is intelligent for this observer
The observer believes he faces a computer program	**undetermined**: the observer has troubles recognizing human intelligence	**failed**: the machine is not intelligent for this observer

Table 2.1: All possible outcomes for a Turing test

Cognitive Benchmarks

Another way to measure whether or not a machine is intelligent is to establish cognitive (or intelligence) benchmarks. A benchmark is a problem definition associated with a performance metrics allowing evaluating the performance of a system. For example in the car industry, some benchmarks measure the time necessary for a car to accelerate from 0 km/h to 100 km/h. Cognitive benchmarks address problems where intelligence is necessary to achieve a good performance.

Again, since intelligence is relative to an observer, the cognitive aspect of a benchmark is also relative to an observer. For example if a benchmark consists in playing chess against the Deep Blue program, some observers may think that this requires some intelligence and hence it is a cognitive benchmark, whereas some other observers may object that it doesn't require intelligence and hence it is not a cognitive benchmark.

Some cognitive benchmarks have been established by people outside computer science and robotics. They include IQ tests developed by psychologists as well as animal intelligence tests developed by biologists to evaluate for example how well rats remember the path to a food source in a maze, or how do monkeys learn to press a lever to get food.

AI and robotics benchmarks have also been established mostly throughout programming or robotics competitions. The most famous examples are the AAAI Robot Competition, the FIRST Robot Competition, the DARPA Grand Challenge, the Eurobot Competition, the RoboCup competition (see picture), the Roboka Programming Contest. All these competitions define a precise scenario and a performance metrics based either on an absolute individual performance evaluation or a ranking between the different competitors. They are very well referenced on the Internet so that it should be easy to reach their official web site for more information.

The last chapter of this book will introduce you to a series of robotics cognitive benchmarks (especially the Rat's Life benchmark) for which you will be able to design your own intelligent systems and compare them to others.

Further reading

- Artificial Intelligence
- Embedded Control Systems Design/RoboCup

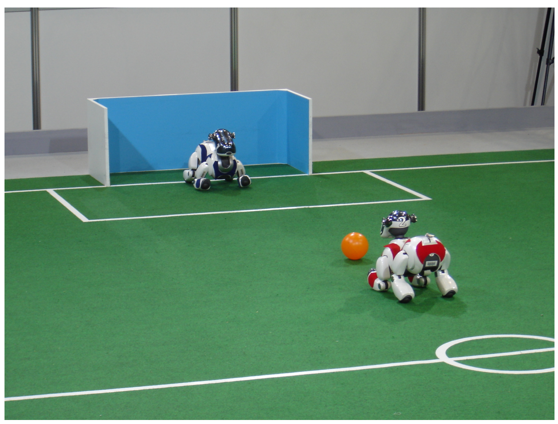

Figure 2.3: Aibo Robocup competition

Chapter 3

What are Robots?

Robots are electro-mechanical machines, interacting autonomously with their environment. They include sensors allowing them to perceive the environment. They also include actuators allowing them to modify their environment. Finally, they include a micro-processor allowing them to process the sensory information and control their actuators accordingly.

Robots in our every Day's Life

There exist few applications of robots in our every days' life. The most well known applications are probably toys and autonomous vacuum cleaners (see figure with toy robots), but there are also grass mower robots, mobile robots in factories, robots for space exploration, surveillance robots, etc. These devices are becoming increasingly complex in term of sensors, actuators and information processing.

Figure 3.1: Two Pleo robots

Figure 3.2: Roomba of first generation: a vacuum cleaner

Robots as Artificial Animals

Like animals, robots can move, perceive their environment and act. Like animals, they need energy to be able to operate. This is probably why several examples of animal robots were developed for toy applications. This includes the Sony Aibo dog robot (see figure), the Furby toy and later the Pleo dinosaur robot. From the mechanical and electronic points of view, these robots are very advanced. They are equipped with many sensors (distance sensors, cameras, touch sensors, position sensors, temperature sensors, battery level sensors, accelerometers, microphones, wireless communication, etc.) and actuators (motors, speakers, LEDs, etc.). They also include a significant processing power with powerful onboard micro-controllers or micro-processors. Moreover, the latest Aibo robots and several vacuum cleaner robots are able to search their recharging station, to dock on it, recharge their batteries and move on once the battery is charged. This makes them even more autonomous. However, their learning capabilities and ability to adapt to unknown situations is often still very limited. Hence, this affect to comparison with real animals in term of intelligence. When observing an Aibo robot and a real dog, there is no doubt for most observers that the dog is more intelligent than the robot. The same could probably apply if you compare the Pleo toy robot with a real reptile. However, since reptiles appear to be more primitive than dogs, the difference of intelligence in the Pleo / reptile case may not be as evident as in the Aibo / dog case.

The conclusion we can draw from the above paragraph is that the hardware technology for intelligent robots is currently available. However, we still need to invent a better software technology to drive these robots. In other words, we currently have the bodies of our intelligent robots, but we lack their minds. This is probably the reason why most of the toy and vacuum cleaner robots described here are still provided with a remote control...

Hence this book will not focus on robot hardware, but rather on robot software because robot software is the greatest research challenge to overcome to be able to design more and more intelligent

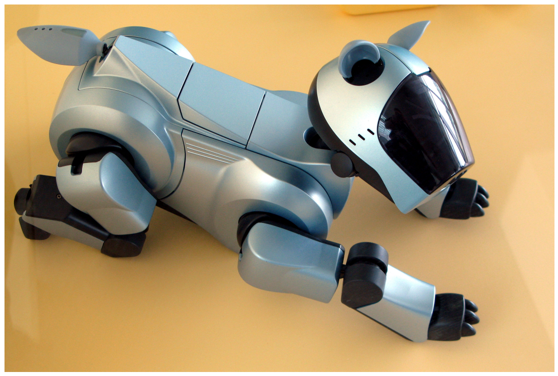

Figure 3.3: Asimo: Honda's humanoid robot

robots.

Chapter 4

E-puck and Webots

This chapter introduces you to a couple of useful robotics tools: e-puck, a mini mobile robot and Webots, a robotics CAD software. In the rest of this book, you will use both of them to practice hands-on robotics. Hopefully, this practical approach will make you understand what robots are and what you can do with them.

E-puck

Introduction

The e-puck robot was designed by Dr. Francesco Mondada and Michael Bonani in 2006 at EPFL, the Swiss Federal Institute of Technology in Lausanne (see Figure). It was intended to be a tool for university education, but is actually also used for research. To help the creation of a community inside and outside EPFL, the project is based on an open hardware concept, where all documents are distributed and submitted to a license allowing everyone to use and develop for it. Similarly, the e-puck software is fully open source, providing low level access to every electronic device and offering unlimited extension possibilities. The e-puck robots are now produced industrially by GCTronic S.à.r.l. (Switzerland) and Applied AI, Inc. (Japan) and are available for purchase from various distributors. You can order your own e-puck robot for about 950 Swiss Francs (CHF) from Cyberbotics Ltd. http://www.cyberbotics.com.

The e-puck robot was designed to meet a number of requirements:

- Neat Design: the simple mechanical structure, electronics design and software of e-puck is an example of a clean and modern system.
- Flexibility: e-puck covers a wide range of educational activities, offering many possibilities with its sensors, processing power and extensions.
- Simulation software: e-puck is integrated in the Webots simulation software for easy programming, simulation and remote control of real robot.
- User friendly: e-puck is small and easy to setup on a table top next to a computer. It doesn't need any cable (rely on Bluetooth) and provides optimal working comfort.

- Robustness and maintenance: e-puck resists to student use and is simple to repair.

- Affordable: the price tag of e-puck is friendly to university budgets.

The e-puck robot has already been used in a wide range of applications, including mobile robotics engineering, real-time programming, embedded systems, signal processing, image processing, sound and image feature extraction, human-machine interaction, inter-robot communication, collective systems, evolutionary robotics, bio-inspired robotics, etc.

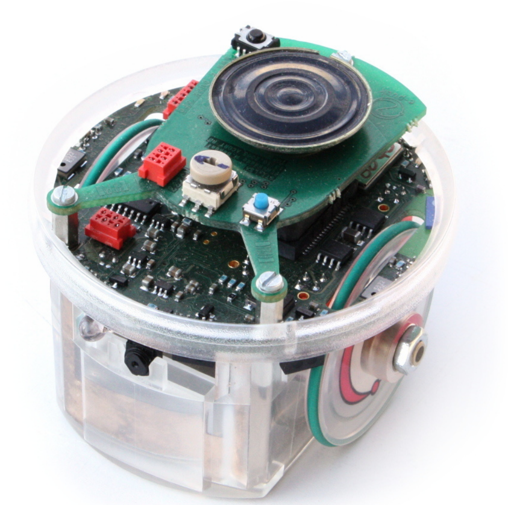

Figure 4.1: The e-puck mobile robot

Overview

The e-puck robot is powered by a dsPIC processor, i.e., a Digital Signal Programmable Integrated Circuit. It is a micro-controller processor produced by the Microchip company which is able to perform efficient signal processing. This feature is very useful in the case of a mobile robot, because extensive signal processing is often needed to extract useful information from the raw values measured by the sensors.

The e-puck robot also features a large number of sensors and actuators as depicted on the pictures with devices and the electronic layout and described in the table. Each of these sensors will be studied in detail during the practical investigations later in this book.

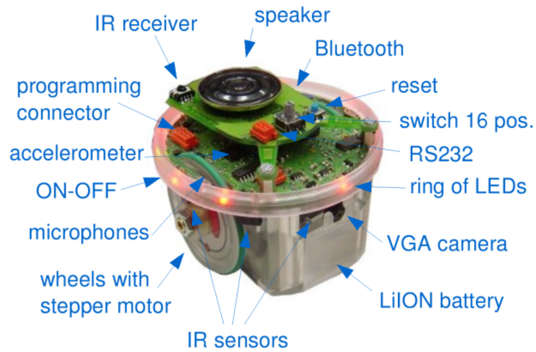

Figure 4.2: Sensors and actuators of the e-puck robot

Webots

Introduction

Webots is a software for fast prototyping and simulation of mobile robots. It has been developed since 1996 and was originally designed by Dr. Olivier Michel at EPFL, the Swiss Federal Institute of Technology in Lausanne, Switzerland, in the lab of Prof. Jean-Daniel Nicoud. Since 1998, Webots is a commercial product and is developed by Cyberbotics Ltd. User licenses of this software have been sold to over 400 universities and research centers world wide. It is mostly used for research and education in robotics. Besides universities, Webots is also used by research organizations and corporate research centers, including Toyota, Honda, Sony, Panasonic, Pioneer, NTT, Samsung, NASA, Stanford Research Institute, Tanner research, BAE systems, Vorverk, etc.

The use of a fast prototyping and simulation software is really useful for the development of most advanced robotics project. It actually allows the designers to visualize rapidly their ideas, to check whether they meet the requirements of the application, to develop the intelligent control of the robots, and eventually, to transfer the simulation results into a real robot. Using such software tools saves a lot of time while developing new robotics projects and allows the designers to explore more possibilities than they would if they were limited to using only hardware. Hence both the development time and the quality of the results are improved by using a rapid prototyping and simulation software.

Overview

Webots allows you to perform 4 basic stages in the development of a robotic project as depicted on the figure.

The first stage is the modeling stage. It consists in designing the physical body of the robots, including their sensors and actuators and also the physical model of the environment of the robots. It is a bit like a virtual LEGO set where you can assemble building blocks and configure them by changing their properties (color, shape, technical properties of sensors and actuators, etc.). This way, any kind of robot can be created, including wheeled robots, four legged robots, humanoid robots, even swimming and flying robots! The environment of the robots is created the same

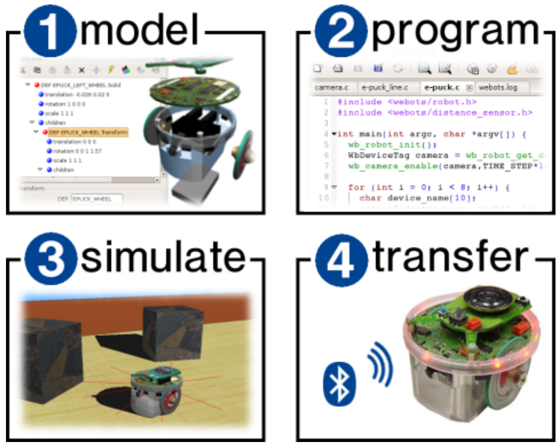

Figure 4.3: Webots development stages

way, by populating the space with objects like walls, doors, steps, balls, obstacles, etc. All the physical parameters of the object can be defined, like the mass distribution, the bounding objects, the friction, the bounce parameters, etc. so that the simulation engine in Webots can simulate their physics. The figure with the simulation illustrates the model of an e-puck robot exploring an environment populated with stones. Once the virtual robots and virtual environment are created, you can move on to the second stage.

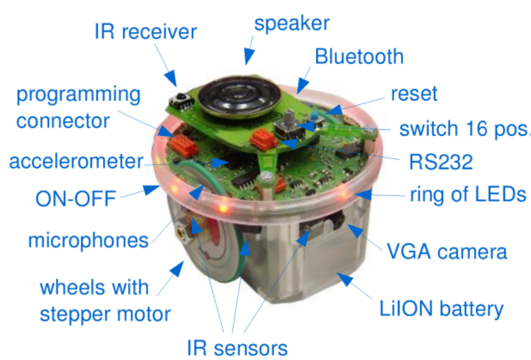

Figure 4.4: Model of an e-puck robot in Webots

The second stage is the programming stage. You will have to program the behavior of each robot. In order to achieve this, different programming tools are available. They include graphical programming tools which are easy to use for beginners and programming languages (like C, C++ or Java) which are more powerful and enable the development of more complex behaviors. The program controlling a robot is generally a endless loop which is divided into three parts: (1) read the values measured by the sensors of the robot, (2) compute what should be the next action(s) of the robot and (3) send actuators commands to performs these actions. The easiest parts are parts (1) and (3). The most difficult one is part (2) as this is here that lie all the Artificial Intelligence. Part (2) can be divided into sub-parts such as sensor data processing, learning, motor pattern generation, etc.

The third stage is the simulation stage. It allows you to test if your program behaves correctly. By running the simulation, you will see you robot executing your program. You will be able to play interactively with you robot, by moving obstacles using the mouse, moving the robot itself, etc. You will also be able to visualize the values measured by the sensors, the results of the processing of your program, etc. It is likely you will return several times to the second stage to fix or improve your program and test it again in the simulation stage.

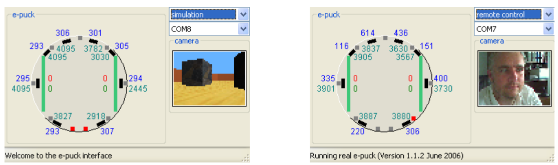

Figure 4.5: Transfer from the simulation to the real robot

Finally, the fourth stage is the transfer to a real robot. Your control program will be transferred into the real robot running in the real world. You could then see if your control program behaves the same as in simulation. If the simulation model of your robot was performed carefully and was calibrated against its real counterpart, the real robot should behave roughly the same as the simulated robot. If the real robot doesn't behave the same, then it is necessary to come back to the first stage and refine the model of the robot, so that the simulated robot will behave like the real one. In this case, you will have to go through the second and third stages again, but mostly for some little tuning, rather than redesigning your program. The figure with two windows shows the e-puck control window allowing the transfer from the simulation to the real robot. On the left hand side, you can see the point of view of the simulated camera of the e-puck robot. On the right hand side, you can see the point of view of the real camera of the robot.

Features	**Thecnical information**
Size, weight	70 mm diameter, 55 mm height, 150 g
Battery autonomy	5Wh LiION rechargeable and removable battery providing about 3 hours autonomy
Processor	dsPIC 30F6014A @ 60 Mhz (˜15 MIPS) 16 bit microcontroller with DSP core
Memory	RAM: 8 KB; FLASH: 144 KB
Motors	2 stepper motors with a 50:1 reduction gear, resolution: 0.13 mm
Speed	Max: 15 cm/s
Mechanical structure	Transparent plastic body supporting PCBs, battery and motors
IR sensors	8 infra-red sensors measuring ambient light and proximity of objects up to 6 cm
Camera	VGA color camera with resolution of 480x640 (typical use: 52x39 or 480x1)
Microphones	3 omni-directional microphones for sound localization
Accelerometer	3D accelerometer along the X, Y and Z axis
LEDs	8 independent red LEDs on the ring, green LEDs in the body, 1 strong red LED in front
Speaker	On-board speaker capable of WAV and tone sound playback
Switch	16 position rotating switch on the top of the robot
PC connection	Standard serial port up to 115 kbps
Wireless	Bluetooth for robot-computer and robot-robot wireless communication
Remote control	Infra-red receiver for standard remote control commands
Expansion bus	Large expansion bus designed to add new capabilities
Programming	C programming with free GNU GCC compiler. Graphical IDE (integrated development environment) provided in Webots
Simulation	Webots facilitates the use of the e-puck robot: powerful simulation, remote control, graphical and C programming systems

Table 4.1: Features of the e-puck robot

Appendix A

Document Information & History

History

This book was created on the Wikibooks project and developed on the project by the contributors listed in Appendix A, page 25. For convenience, this PDF was created for download from the project. The latest Wikibooks version may be found at `http://en.wikibooks.org/wiki/Cyberbotics'_Robot_Curriculum`.

PDF Information & History

This PDF was compiled from LaTeX on January 7, 2009, based on the 7 January 2009 Wikibooks textbook.The latest version of the PDF may be found at `http://en.wikibooks.org/wiki/Image:Cyberbotics'_Robot_Curriculum.pdf`.

Authors

Cyberbotics, Olivier Michel, Fabien Rohrer, Nicolas Heiniger, DavidCary, Trolli101, and anonymous contributors.

Appendix B

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.