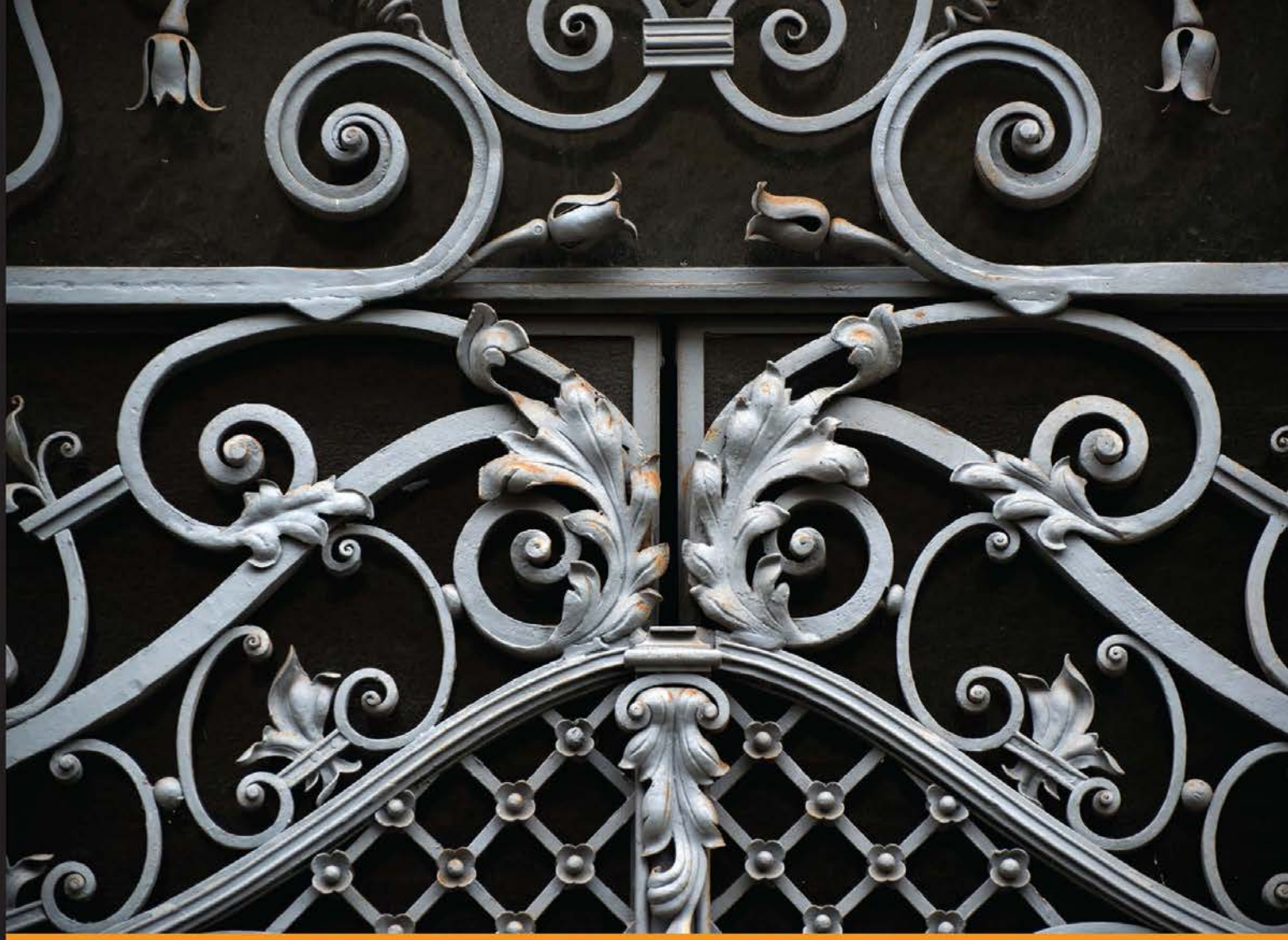


Community Experience Distilled

# Learning Robotics Using Python

Design, simulate, program, and prototype an interactive autonomous mobile robot from scratch with the help of Python, ROS, and Open-CV!

**Lentin Joseph**

[PACKT] PUBLISHING open source* community experience distilled

# Learning Robotics Using Python

Design, simulate, program, and prototype an interactive autonomous mobile robot from scratch with the help of Python, ROS, and Open-CV!

**Lentin Joseph**

[PACKT] PUBLISHING open source community experience distilled

BIRMINGHAM - MUMBAI

# Learning Robotics Using Python



First published: May 2015

Production reference: 1250515

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78328-753-6

www.packtpub.com

Cover image by Jarek Blaminsky (milak6@wp.pl)

# Credits


**Author**
Lentin Joseph

**Reviewers**
Avkash Chauhan
Vladimir Iakovlev
Blagoj Petrushev
Marek Suppa

**Commissioning Editor**
Rebecca Youé

**Acquisition Editor**
Rebecca Youé

**Content Development Editor**
Athira Laji

**Technical Editors**
Ankur Ghiye
Manali Gonsalves

**Copy Editors**
Pranjali Chury
Relin Hedly
Merilyn Pereira
Adithi Shetty

**Project Coordinator**
Harshal Ved

**Proofreaders**
Stephen Copestake
Safis Editing

**Indexer**
Priya Sane

**Graphics**
Sheetal Aute

**Production Coordinator**
Nitesh Thakur

**Cover Work**
Nitesh Thakur

# About the Author

**Lentin Joseph** is an electronics engineer, robotics enthusiast, machine vision expert, embedded programmer, and the founder and CEO of Qbotics Labs (`http://www.qboticslabs.com`) in India. He got his bachelor's degree in electronics and communication engineering at the Federal Institute of Science and Technology (FISAT), Kerala. In his final year engineering project, he created a social robot, which can interact with people. The project was a huge success and got mentioned in visual and print media. The main feature of this robot was that it could communicate with people and reply intelligently. It also has some image-processing capabilities, such as face, motion, and color detection. The entire project was implemented using the Python programming language. His interest in robotics, image processing, and Python began this project.

After graduation, he worked at a start-up company based on robotics and image processing for 3 years. In the meantime, he learned famous robotic software platforms—such as Robot Operating system (ROS), V-REP, and Actin (a robotic simulation tool)—and image processing libraries, such as OpenCV, OpenNI, and PCL. He also knows about robot 3D designing, embedded programming on Arduino, and Stellaris Launchpad.

After 3 years of work experience, he started a new company called Qbotics Labs, which is mainly focused on research to build great products in domains such as wearable technology, robotics, machine vision, green technology, and online education. He maintains a personal website (`http://www.lentinjoseph.com`) and a technology blog called technolabsz (`http://www.technolabsz.com`). He publishes his works on his tech blog. He was a speaker at PyCon2013 India, and he spoke on the topic of learning robotics using Python.

---

I would like to dedicate this book to my parents because they gave me the inspiration to write it. I would also like to convey my regards to my friends who helped and inspired me to write this book.

I would like to thank Marek Suppa for his valuable contribution in writing *Chapter 1*, *Introduction to Robotics*, in addition to reviewing this book.

---

# About the Reviewers

**Avkash Chauhan** is currently leading a team of engineers at a start-up based in San Francisco, where his team is building a big data monitoring platform using machine learning and new age methods to improve business continuity and gain maximum advantage from the platform itself. He is the founder and principal of Big Data Perspective, with a vision to make the Hadoop platform accessible to mainstream enterprises by simplifying its adoption, customization, management, and support. Before Big Data Perspective, he worked at Platfora Inc., building big data analytics software running natively on Hadoop. Previously, he worked for 8 years at Microsoft, building cloud and big data products and providing assistance to enterprise partners worldwide. Avkash has over 15 years of software development experience in cloud and big data disciplines. He is a programmer at heart in full-stack discipline and has the business acumen to work with enterprises, meeting their needs. He is passionate about technology and enjoys sharing his knowledge with others through various social media. He has also written a few books on big data discipline and is very active in the tech social space. He is an accomplished author, blogger, technical speaker, and he loves the outdoors.

**Vladimir Iakovlev** is a software developer. Most of the time, he develops web applications using Python, Clojure, and JavaScript. He's the owner of a few semi-popular open source projects. He was a speaker at a few Python-related conferences.

In his free time, Vladimir likes to play with electronic devices, such as Arduino and PyBoard, and image-processing devices, such as Leap Motion. He has tried to build some robots. He has already built a robotic arm.

Currently, Vladimir works at Upwork, where he develops web applications, mostly with Python.

**Blagoj Petrushev** is a software engineer and consultant based in Skopje, Macedonia. His work revolves mainly around backends, datastores, and network applications. Among his interests are machine learning, NLP, data analysis, modeling and databases, and distributed programming.

**Marek Suppa** has been playing with (kind of) smart machines for the past few years, which are pretentiously called robots in some parts of the world. Right now, he leads a robotic football team, building tools to help others start with robots and setting off on a new venture to see how far the current technology will let us move toward the goal of creating a robot as it was first defined.

---

I would like to thank everyone who supported the creation of this book, whoever and wherever they might be.

---

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit `www.PacktPub.com`.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.


`https://www2.packtpub.com/books/subscription/packtlib`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

### Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

### Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

# Preface

Learning Robotics with Python contains twelve chapters that mainly aims at how to build an autonomous mobile robot from scratch and how to program it using Python. The robot mentioned in this book is a service robot, which can be used to serve food at home, hotels, and restaurants. From the beginning to end, this book discusses the step-by-step procedure on how to build this robot. The book starts with the basic concepts of robotics and then moves on to the 3D modeling and simulation of the robot. After the successful simulation of the robot, it discusses the hardware components required to build the robot prototype in order to complete the robot navigation.

The software part of this robot is mainly implemented using the Python programming language and software frameworks, such as Robot Operating System (ROS), Open-CV, and so on. You will understand the application of Python from the aspects of designing the robot to the robot's user interface. The Gazebo simulator is used to simulate the robot and machine vision libraries, such as Open-CV and OpenNI. PCL is used to process the 2D and 3D image data of the robot. Each chapter is presented with an adequate theory to understand the application aspect. The book is reviewed by experts in this field who are passionate about robotics.

## What this book covers

*Chapter 1*, *Introduction to Robotics*, contains basic concepts and terminologies of robotics. This chapter is a must for beginners who are just starting with robotics.

*Chapter 2*, *Mechanical Design of a Service Robot*, discusses the 2D and 3D CAD designing aspect of the robot using LibreCAD and Blender (free software). This chapter also demonstrates how to use Blender Python APIs in order to build the 3D model.

*Chapter 3*, *Working with Robot Simulation Using ROS and Gazebo*, takes you through the simulation of the service robot using Gazebo and ROS.

*Chapter 4*, *Designing ChefBot Hardware*, explains the hardware designing of the robot, including block diagram and hardware components required to build ChefBot.

*Chapter 5*, *Working with Robotic Actuators and Wheel Encoders*, covers interfacing of robotic actuators and wheel encoders using Tiva C LaunchPad. It also mentions high-end smart actuators like dynamixel.

*Chapter 6*, *Working with Robotic Sensors*, discusses interfacing of ultrasonic distance sensors, IR proximity sensors, and IMU using Tiva C LaunchPad.

*Chapter 7*, *Programming Vision Sensors Using Python and ROS*, talks about the introduction to Open-CV, OpenNI, and PCL libraries and interfacing these to ROS and programming using Python.

*Chapter 8*, *Working with Speech Recognition and Synthesis Using Python and ROS*, discusses speech recognition and synthesis using various libraries and interfacing it to ROS programming using Python.

*Chapter 9*, *Applying Artificial Intelligence to ChefBot Using Python*, covers tutorials to build a ChatterBot. This can be used to make the robot interactive.

*Chapter 10*, *Integration of ChefBot Hardware and Interfacing it into ROS, Using Python*, explores tutorials to integrate the complete hardware and essential software section. It mainly discusses autonomous navigation of the service robot and how to program it using ROS and Python.

*Chapter 11*, *Designing a GUI for a Robot Using Qt and Python*, covers tutorials on how to build a GUI for the user who operates the robot in a typical restaurant. The GUI is built using Qt and the PyQt Python wrapper.

*Chapter 12*, *The Calibration and Testing of ChefBot*, explores tutorials on how to calibrate and test the robot for the final run.

# What you need for this book

The book is all about how to build a robot. To start with this book, you should have some hardware. The robot can be built from scratch, or you can buy a differential-drive configuration robot with an encoder feedback. You should buy a controller board, such as Texas Instruments Launchpad, for embedded processing. You should have at least a laptop/net book for the entire robot process. In this book, we will use Intel NUC for robot processing. It's very compact in size and delivers high performance. For the 3D vision, you should have 3D sensors, such as laser scanner, Kinect, and Asus Xtion Pro.

In the software section, you should have a good understanding on how to work with GNU/Linux commands. You should also have a good knowledge of Python. You should install Ubuntu 14.04.2 LTS to work with the examples. If you have knowledge about ROS, OpenCV, OpenNI, and PCL, it will be a great add-on. You have to install ROS Indigo to test these examples.

# Who this book is for

Learning Robotics with Python is a good companion for entrepreneurs who want to explore the service robotics domain, professionals who want to implement more features to their robots, researchers who want to explore more about robotics, and hobbyist or students who want to learn robotics. The book follows a step-by-step guide that can be easily understood by anyone.

# Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: " The first procedure is to create a world file and save it with the `.world` file extension."

A block of code is set as follows:

```
<xacro:include filename="$(find
  chefbot_description)/urdf/chefbot_gazebo.urdf.xacro"/>
<xacro:include filename="$(find
  chefbot_description)/urdf/chefbot_properties.urdf.xacro"/>
```

Any command-line input or output is written as follows:

```
$ roslaunch chefbot_gazebo chefbot_empty_world.launch
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: " we can command the robot to navigate to some position on the map using the **2D Nav Goal** button".

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

# Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

# Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

# Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from: `https://www.packtpub.com/sites/default/files/downloads/7536OS_ImageBundle.pdf`.

# Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

# Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

# Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

# 1
# Introduction to Robotics

If you read an introductory chapter in any technical book, you may have noticed that it pretty much always follows the same structure. It begins by describing how awesome the topic is, what a good decision it is to start reading the book, and how you should keep on reading because there are many exciting things awaiting you in its further chapters.

This chapter is no such chapter. It starts with the following quote:

> *Robotics is an art.*

Although, such a strong statement does probably deserve some explanation, we believe that after you finish reading this book (and building your own robots!), no further explanation will be needed.

So if robotics is an art, how does one learn it? To put it differently, what are the differences between learning to play a musical instrument, learning to paint, learning to write, and learning robotics? We believe that there are not too many of them. Just as musicians need to play on their instruments, painters need to produce paintings, and writers need to write their texts, roboticists (the term we use to describe people who build robotics) need to build their robots. Just as musicians, painters, and writers need to learn the jargon used in their trades, roboticists need to familiarize themselves with a few basic terms that they might run into while reading tutorials, researching scientific literature, and talking to other robotics enthusiasts. Also, just as any artist needs to know at least a little bit about the history of their respective art, so does any good roboticist need to know a thing or two about the history of robotics. That's why in this chapter, we will cover:

- What is a robot?
- Where do robots come from?
- What can we find in a robot?
- How do we build robots?

# What is a robot?

Rather than defining what a robot is right away, let's pause for a moment and discuss whether we need to answer a question like this after all. Everybody knows that a robot is some sort of a machine that can move around and depending on what movie you saw or which book you read, it can either help humans in their day-to-day life or mean the end of humanity.

It's clear that there is some controversy and lots of misunderstandings about robots and their role in the past, present, and the future. In order to better understand the situation, let's first examine closely the term "robot" itself. Then, we will try to define it a bit more formally to prevent any misunderstanding or controversy.

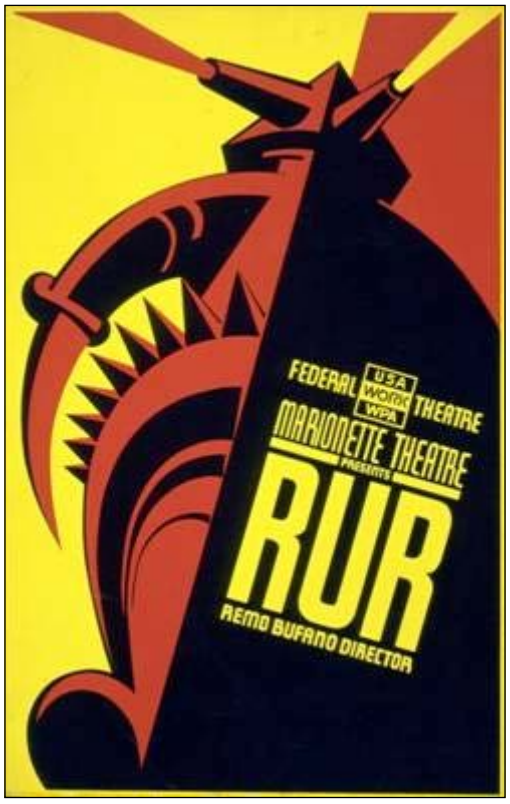
# History of the term robot

The term "robot" was used for the first time by Karel Čapek, a Czech writer in his play **Rossum's Universal Robots** (**R.U.R**) that he wrote in 1920, to denote an artificial human made out of synthetic organic matter. These robots (*roboti* in Czech) were made in factories and their purpose was to replace human workers. While they were very efficient and executed orders they were given perfectly, they lacked any emotion. It seemed that humans would not need to work at all because robots seemed to be happy to work for them. This changed after a while and a robot revolt resulted in extinction of the human race.

R.U.R is quite dark and disturbing, but it does not leave the future hopeless. It was considered quite a success back in the day and we certainly do recommend you to read it. As its copyright had already expired in many countries at the time of writing this book, it should not be a problem to find a version online, which is in the public domain.

> *"When he (Young Rossum) took a look at human anatomy he saw immediately that it was too complex and that a good engineer could simplify it. So he undertook to redesign anatomy, experimenting with what would lend itself to omission or simplification. Robots have a phenomenal memory. If you were to read them a twenty-volume encyclopedia they could repeat the contents in order, but they never think up anything original. They'd make fine university professors."*
>
> *– Karel Capek, R.U.R. (Rossum's Universal Robots), 1920*

While many attribute the term robot to Karel Čapek as he wrote the play in which it appeared for the first time, there are sources suggesting that it was actually Čapek's brother Josef who came up with the term (it seems that there was an article in Czech daily print written by Karel Čapek himself, in which he wants to set the record straight by telling this story). Karel wanted to use the term *laboři* (from Latin labor, work), but he did not like it. It seemed too artificial to him, so he asked his brother for advice. Josef suggested *roboti* and that was what Karel used in the end.




Now that we know when the term *robot* was used for the first time and who actually created it, let's find out where does it come from. The explanation that many use is that it comes from the Czech words *robota* and *robotník*, which literally means "work" and "worker" respectively. However, the word *robota* also means "work" or "serf labor" in Slovak. Also, we should take into account that some sources suggest that by the time Karel was writing R.U.R, he and his brother often visited his father in a small Slovak spa town called Trenčianske Teplice. Therefore, it might very well be that the term robot was inspired by the usage of the word "robota" in Slovak language, which is coincidentally, the native language of one of the authors of this book.

Whether the term robot comes from Czech or Slovak, the word robota might be a matter of national pride, but it does not concern us too much. In both cases, the literal meaning is "work", "labor", or "hard work" and it was the purpose of the Čapek's robots. However, robots have evolved dramatically over the past hundred years. To say that they are all about doing hard work would probably be an understatement.

So, let's try to define the notion of a robot as we perceive it today.

# Modern definition of a robot

When we try to find a precise definition of some term, our first stop is usually some sort of encyclopedia or a dictionary. Let's try to do this for the term robot.

Our first stop will be *Encyclopedia Britannica*. Its definition of a robot is as follows:

> "*Any automatically operated machine that replaces human effort, though it might not resemble human beings in appearance or preform functions in a humanlike manner.*"

This is quite a nice definition, but there are quite a few problems with it.

First of all, it's a bit too broad. By this definition, a washing machine should also be considered a robot. It does operate automatically (well, most of them do), it does replace human effort (although not by changing the same tasks a human would do), and it certainly does not resemble a human.

Secondly, it's quite difficult to imagine what a robot actually is after reading this definition. With such a broad definition, there are way too many things that can be considered a robot and this definition does not provide us with any specific features.

It turns out that while Encyclopedia Britannica's definition of a robot does not fit our needs well enough, it's actually one of the best ones that one can find. For example, The Free Dictionary defines a robot as "*A mechanical device that sometimes resembles a human and is capable of performing a variety of often complex human tasks on command or by being programmed in advance*." This is even worse than what we had and it seems that a washing machine should still be considered a robot.

The inherent problem with these definitions is that they try to capture vast amount of machines that we call robots these days. The result is that it's very difficult, if not impossible, to come up with a definition that will be comprehensive enough and not include a washing machine at the same time. John Engelberger, founder of the world's first robotics company and industrial robotics (as we know it today) once famously said, "*I can't define a robot, but I know one when I see one*."

So, is it even possible to define a robot? Maybe not in general. However, if we limit ourselves just to the scope of this book, there may be a definition that will suit our needs well enough. In her very nice introductory book on the subject of robotics called *The Robotics Primer* (which we also highly recommend), *Maja J. Mataric* uses the following definition:

> "*A robot is an autonomous system which exists in the physical world, can sense its environment, and can act on it to achieve some goals.*"

At first sight, it might not seem like a vast improvement over what we have so far, but let's dissect it part by part to see whether it meets our needs.

The first part says, "A robot is an **autonomous** system". By autonomous, we mean that a robot makes decisions on its own—it's not controlled by a human. This already seems to be an improvement as it weeds out any machine that's controlled by someone (such as our famous washing machine). Robots that we will talk about throughout this book may sometimes have some sort of a remote function, which allows a human to control it remotely, but this functionality is usually built-in as sort of a safety measure so that if something goes wrong and the robot's autonomous systems fails to behave as we would expect them to, it's still possible to get the robot to safety and diagnose its problems afterwards. However, the main goal still stays the same, that is, to build robots that can take some direction from humans and are able to act and function on their own.

However, just being an autonomous system will certainly not be enough for a robot in this book. For instance, we can find many computer programs that we can call autonomous systems (they are not controlled by an individual and make decisions on their own) and yet we do not consider them to be robots.

To get around this obstacle, we need the other part of the sentence that says, "which exists in the **physical** world".

Given the recent advances in the fields of artificial intelligence and machine learning, there is no shortage of computer systems that act on their own and perform some work for us, which is what robots should be for. As a quite notorious example, let's consider **spam** filters. These are computer programs that read every e-mail that reaches your e-mail address and decides whether you may want to read it (and that the e-mail is indeed legitimate) or whether it's yet another example of an unwanted e-mail.

There is no doubt that such a system is helpful (if you disagree, try to read some of the e-mails in your `Spam` folder—I am pretty sure it will be a boring read). It's estimated that over 60 percent of all e-mail traffic in 2014 can be attributed to spam e-mails. Being able to automatically filter them can save us a lot of reading time. Also, as there is a no human involved in the decision process (although, we can help it by marking an e-mail as spam), we can call such a system as autonomous. Still, we will not call it a true robot. Rather, we call them "software robots" or just "bots" (the fact that their name is shorter may come from the fact that they are short of the physical parts of true robots).

While software robots are definitely an interesting group on its own, it's the physical world in which robots operate that makes the process of creating them so exciting and difficult at the same time. When creating a software robot, you can count on the fact that the environment it will run in (usually the operating system) will be quite stable (as in, not too many things may change unexpectedly). However, when you are creating a real robot, you can never be sure.

This is why a real robot needs to know what is happening in the environment in which it operates. Also, this is why the next part of the definition says, "can *sense* its environment".

Sensing what is happening around a real robot is arguably its most important feature. To sense their surrounding environments, robots usually have sensors. These are devices that measure physical characteristics of the environment and provide this information back to the robot so that it can, for instance, react to sudden changes of temperature, humidity, or pressure. This is quite a big difference from software robots. While they just get the information they need in order to operate somewhat magically, real robots need to have a subsystem or subsystems that take care of obtaining this information. If we look at the differences between robots and humans, we will not find many (in our very high-level view, of course). We can think of sensoring subsystems as artificial replacements for human organs that provide this sort of information to the brain.

One important consequence of this definition is that anything that does not sense its environment cannot be called a robot. This includes any devices that just "drive blind" or move in a random fashion because they do not have any information from the environment to base their behavior on.

Any roboticist will tell you that robots are very exciting machines. Many will also argue that what makes them so exciting is actually their ability to interact with the outside world (which is to move or otherwise change the environment they are in). Without this, they are just another static machine that might be useful, but rather unexciting.

Our definition of a robot reflects this in its last part when it says, "can *act on it* to *achieve* some *goals*".

Acting on the environment might sound like a very complex task for a robot, but in this case, it just means changing the world in some (even very slight) way. We call these parts of robots that perform this as *effectors*. If we look at our robot vs human comparison, effectors are the artificial equivalents of hands, legs, and other body parts that allow it to move. Effectors make use of some lower-level systems such as motors or muscles that actually carry out the movement. We call them *actuators*. Although, the artificial ones may seem to function similar to the biological ones, a closer look will reveal that they are actually quite different.

You may have noticed that this part is not only about acting on the robot's environment, but also about achieving some goals. While many hobby roboticists build robots just for the fun of it, most robots are built in order to carry out (or, should we rather say, to help with) some tasks, such as moving heavy parts in a factory or locating victims in areas affected by natural disasters.

As we said before, a system or a machine that behaves randomly and does not use information from its environment cannot really be considered a robot. However, how can it use these information somehow? The easiest thing to do is to do something useful, which we can rephrase as trying to reach some goal that we consider useful, which in turn brings us back to our definition. A goal of a robot does not necessarily need to be something as complex and ambitious as "hard labor for human". It can easily be something simple, such as "do not bump into obstacles" or "turn the light switch on".

Now, as we have at least a slight idea of what a robot is, we can move on to briefly discuss where robots come from, in other words, the history of robotics.

# Where do robots come from?

As the title suggests, this part of the chapter should be about the history of robots. We already know a few quite important facts, such as the term robot was coined by a Czech author Karel Čapek in 1920. As it turns out, there are many more interesting events that happened over the years, other than this one. In order to keep things organized, let's start from the beginning.

It's quite difficult to pinpoint a precise date in history, which we can mark as the date of birth of the first robot. For one, we have established quite a restrictive definition of a robot previously; thus, we will have to wait until the 20th century to actually see a robot in the proper sense of the word. Until then, let's at least discuss the honorable mentions.

The first one that comes close to a robot is a mechanical bird called "The Pigeon". This was postulated by a Greek mathematician Archytas of Tarentum in the 4th century BC and was supposed to be propelled by steam. It cannot not be considered a robot by our definition (not being able to sense its environment already disqualifies it), but it comes pretty close for its age. Over the following centuries, there were many attempts to create automatic machines, such as clocks measuring time using the flow of water, life-sized mechanical figures, or even first programmable humanoid robots (it was actually a boat with four automatic musicians on it). The problem with all these is that they are very disputable as there is very little (or none) historically trustworthy information available about these machines.

It would have stayed like this for quite some time if it was not for Leonardo Da Vinci's notebooks that were rediscovered in 1950s. They contain a complete drawing of a 1945 humanoid (a fancy word for a mechanical device that resemble humans), which looks like an armored knight. It seems that it was designed so that it could sit up, wave its arms, move its head, and most importantly, amuse royalty. In the 18th century, following the amusement line, Jacques de Vaucanson created three automata: a flute player that could play twelve songs, a tambourine player, and the most famous one, "The Digesting Duck". This duck was capable of moving, quacking, flapping wings, or even eating and digesting food (not in a way you will probably think—it just released matter stored in a hidden compartment). It was an example of "moving anatomy"—modeling human or animal anatomy using mechanics.

Our list will not be complete if we omitted these robot-like devices that came about in the following century. Many of them were radio-controlled, such as Nikola Tesla's boat, which he showcased at Madison Square Garden in New York. You could command it to go forward, stop, turn left or right, turn its lights on or off, and even submerge. All of this did not seem too impressive at that time because the press reports attributed it to "mind control".

At this point, we have once again reached the time when the term **robot** was used for the first time. As we said many times before, it was in 1920 when Karel Čapek used it in his play, R.U.R. Two decades later, another very important term was coined. Issac Asimov used the term **robotics** for the first time in his story "Runaround" in 1942. Asimov wrote many other stories about robots and is considered to be a prominent sci-fi author of his time.

However, in the world of robotics, he is known for his three laws of robotics:

- **First law**: A robot may not injure a human being or through inaction allow a human being to come to harm.
- **Second Law**: A robot must obey the orders given to it by human beings, except where such orders would conflict with the first law.
- **Third law**: A robot must protect its own existence, as long as such protection does not conflict with the first or second law.

After a while, he added a zeroth law:

- **Zeroth law**: A robot may not harm humanity or by inaction allow humanity to come to harm.

These laws somehow reflect the feelings people had about machines they called robots at that time. Seeing enslavement by some sort of intelligent machine as a real possibility, these laws were supposed to be some sort of guiding principles one should at least keep in mind, if not directly follow, when designing a new intelligent machine. Also, while many were afraid of the robot apocalypse, time has shown that it's still yet to come. In order for it to take place, machines will need to get some sort of intelligence, some ability to think, and act based on their thoughts. Also, while we can see that over the course of history, the mechanical side of robots went through some development, the intelligence simply was not there yet.

This was part of the reason why in the summer of 1956, a group of very wise gentlemen (which included Marvin Minsky, John McCarthy, Herbert Simon, and Allan Newell) were later called to be the founding fathers of the newly founded field of Artificial Intelligence. It was at this very event where they got together to discuss creating intelligence in machines (thus, the term artificial intelligence).

Although, their goals were very ambitious (some sources even mention that their idea was to build this whole machine intelligence during that summer), it took quite a while until some interesting results could be presented.

One such example is **Shakey**, a robot built by the **Stanford Research Institute** (**SRI**) in 1966. It was the first robot (in our modern sense of the word) capable to reason its own actions. The robots built before this usually had all the actions they could execute preprogrammed. On the other hand, Shakey was able to analyze a more complex command and split it into smaller problems on his own. The following image of Shakey is taken from `https://en.wikipedia.org/wiki/File:ShakeyLivesHere.jpg`:

Shakey, resting in the Computer History Museum in Mountain View, California

His hardware was quite advanced too. He had collision detectors, sonar range finders, and a television camera. He operated in a small closed environment of rooms, which were usually filled with obstacles of many kinds. In order to navigate around these obstacles, it was necessary to find a way around these obstacles while not bumping into something. Shakey did it in a very straightforward way.