What is AI (and what is it not)?

What would be a definition of AI? In general, it means a machine that exhibits some characteristics of intelligence—thinking, reasoning, planning, learning, and adapting. It canalso mean a software program that can simulate thinking or reasoning. Let's try some examples: a robot that avoids obstacles by simple rules (if the obstacle is to the right, go left)is not an AI. A program that learns by example to recognize a cat in a video, is an AI. A mechanical arm that is operated by a joystick is not AI, but a robot arm that adapts to different objects in order to pick them up is AI.

There are two defining characteristics of artificial intelligence robots that you must be aware of. First of all, AI robots learn and adapt to their environments, which means thatthey change behaviors over time. The second characteristic is emergent behavior, where therobot exhibits developing actions that we did not program into it explicitly. We are giving the robot controlling software that is inherently non-linear and self-organizing. The robot may suddenly exhibit some bizarre or unusual reaction to an event or situation that seems tobe odd, or quirky, or even emotional. I worked with a self-driving car that we swore had delicate sensibilities and moved very daintily, earning it the nickname Ferdinand after the sensitive, flower loving bull from the cartoon, which was appropriate in a nine-ton truck that appeared to like plants. These behaviors are just caused by interactions of the various software components and control algorithms, and do not represent anything more than that.One concept you will hear around AI circles is the Turing test. The Turing test was proposedby Alan Turing in 1950, in a paper entitled Computing Machinery and Intelligence. He postulated that a human interrogator would question an hidden, unseen AI system, along with another human. If the human posing the questions was unable to tell which person wasthe computer and which the human was, then that AI computer would pass the test. This testsupposes that the AI would be fully capable of listening to a conversation, understanding thecontent, and giving the same sort of answers a person will. I don't believe that AI has progressed to this point yet, but chat bots and automated answering services have done a good job of making you believe that you are talking to a human and not a robot.

Our objective in this book is not to pass the Turing test, but rather to take some novelapproaches to solving problems using techniques in machine learning, planning, goal seeking, pattern recognition, grouping, and clustering. Many of these problems would be very difficult to solve any other way. A software AI that could pass the Turing test would bean example of a general artificial intelligence, or a full, working intelligent artificial brain, and just like you, a general AI does not need to be specifically trained to solve any particular

problem. To date, a general AI has not been created, but what we do have is narrow AI, or software that simulates thinking in a very narrow application, such as recognizing objects, or picking good stocks to buy. What we are not building in this book is a general AI, and weare not going to be worried about our creations developing a mind of their own or getting out of control. That comes from the realm of science fiction and bad movies, rather than thereality of computers today. I am firmly of the mind that anyone preaching about the evils ofAI or predicting that robots will take over the world has not worked or practiced in this area,and has not seen the dismal state of AI research in respect of solving general problems or creating anything resembling an actual intelligence.

There is nothing new under the sun

Most of AI as practiced today is not new. Most of these techniques were developedin the 1960s and 1970s and fell out of favor because the computing machinery of the day was insufficient for the complexity of the software or number of calculations required, and only waited for computers to get bigger, and for another very significant event – the invention of the internet. In previous decades, if you needed 10,000 digitized pictures of catsto compile a database to train a neural network, the task would be almost impossible—you could take a lot of cat pictures, or scan images from books. Today, a Google search for cat pictures returns 126,000,000 results in 0.44 seconds. Finding cat pictures, or anything else,is just a search away, and you have your training set for your neural network—unless you need to

train on a very specific set of objects that don't happen to be on the internet, as we will see in this book, in which case we will once again be taking a lot of pictures with anothermodern aid not found in the 1960s, a digital camera. The happy combination of very fast computers; cheap, plentiful storage; and access to almost unlimited data of every sort has produced a renaissance in AI. Another modern development has occurred on the other end of the computer spectrum. While anyone can now have a supercomputer on their desk at home, the development of the smartphone has driven a whole series of innovations that arejust being felt in technology. Your wonder of a smartphone has accelerometers and gyroscopes made of tiny silicon chips called microelectromechanical systems (MEMS). It also has a high resolution but very small digital camera, and a multi-core computer processorthat takes very little power to run. It also contains (probably) three radios: a WiFi wireless network, a cellular phone, and a Bluetooth transceiver. As good as these parts are at makingyour iPhone™ fun to use, they have also found their way into parts available for robots. Thatis fun for us because what used to be only available for research labs and universities are now for sale to individual users. If you happen to have a university or research lab, or workfor a technology company with multi-millions dollar development budgets, you will also

learn something from this book, and find tools and ideas that hopefully will inspire yourrobotics creations or power new products with exciting capabilities.

The example problem – clean up this room!

In the course of this book, we will be using a single problem set that I feel most people can relate to easily, while still representing a real challenge for the most seasoned roboticist. We will be using AI and robotics techniques to pick up toys in my upstairs gameroom after my grandchildren have visited. That sound you just heard was the gasp from theprofessional robotics engineers and researchers in the audience. Why is this a tough problem,and why is it ideal for this book? This problem is a close analog to the problem Amazon hasin picking items off of shelves and putting them in a box to send to you. For the last severalyears, Amazon has sponsored the Amazon Robotics Challenge where they invited teams totry and pick items off shelves and put them into a box for cash prizes. They thought the program difficult enough to invite teams from around the world. The contest was won in 2017 by a team from Australia. Let’s discuss the problem and break it down a bit. Later, inChapter 2, we will do a full task analysis, use cases, and storyboards to develop our approach, but we can start here with some general observations. Robotics designers first start with the environment – where does the robot work? We divide environments into two categories: structured and unstructured. A structured environment, such as the playing fieldfor a first robotics competition, an assembly line, or lab bench, has everything in an organized space. You have heard the saying A place for everything and everything in its place—that is a structured environment. Another way to think about it, is that we know in advance where everything is or is going to be. We know what color things are, where they are placed in space, and what shape they are. A name for this type of information is a prior knowledge – things we know in advance. Having advanced knowledge of the environment in robotics is sometimes absolutely essential. Assembly line robots are expecting parts to arrive in exactly the position and orientation to be grasped and placed into position. In otherwords, we have arranged the world to suit the robot. In the world of our game room, this issimply not an option. If I could get my grandchildren to put their toys in exactly the same spot each time, then we would not need a robot for this task. We have a set of objects that isfairly fixed – we only have so many toys for them to play with. We occasionally add thingsor lose toys, or something falls down the stairs, but the toys are a element of a set of fixed objects. What they are not is positioned or oriented in any particular manner – they are justwhere they were left when the kids finished playing with them and went home. We also havea fixed set of furniture, but some parts move – the footstool or chairs can be moved around.

This is an unstructured environment, where the robot and the software have to adapt, not thetoys or furniture.

The problem is to have the robot drive around the room, and pick up toys. Let's breakthis task down into a series of steps: 1. We want the user to interact with the robot by talkingto it. We want the robot to understand what we want it to do, which is to say, what our intentis for the commands we are giving it. 2. Once commanded to start, the robot will have to identify an object as being a toy, and not a wall, a piece of furniture, or a door. 3. The robotmust avoid hazards, the most important being the stairs going down to the first floor. Robotshave a particular problem with negative obstacles (dropoffs, curbs, cliffs, stairs, and so on),and that is exactly what we have here. 4. Once the robot finds a toy, it has to determine howto pick the toy up with its robot arm. Can it grasp the object directly, or must it scoop the item up, or push it along? We expect that the robot will try different ways to pick up toys and may need several trials and error attempts. 5. Once the toy is acquired by the robot arm,the robot needs to carry the toy to a toy box. The robot must recognize the toy box in the room, remember where it is for repeat trips, and then position itself to place the toy in the box. Again, more than one attempt may be required. 6. After the toy is dropped off, the robotreturns to patrolling the room looking for more toys. At some point, hopefully, all of the toysare retrieved. It may have to ask us, the human, if the room is acceptable, or if it needs to continue cleaning. What will we be learning from this problem? We will be using this backdrop to examine a variety of AI techniques and tools. The purpose of the book is to teach you how to develop AI solutions with robots. It is the process and the approach that isthe critical information here, not the problem and not the robot I developed so that we havesomething to take pictures of for the book. We will be demonstrating techniques for makinga moving machine that can learn and adapt to its environment. I would expect that you willpick and choose which chapters to read and in which order according to your interests and you need, and as such, each of the chapters will be standalone lessons. The first three chapters are foundation material that support all of the rest of the book by setting up the problem and providing a firm framework to attach all of the rest of the material.

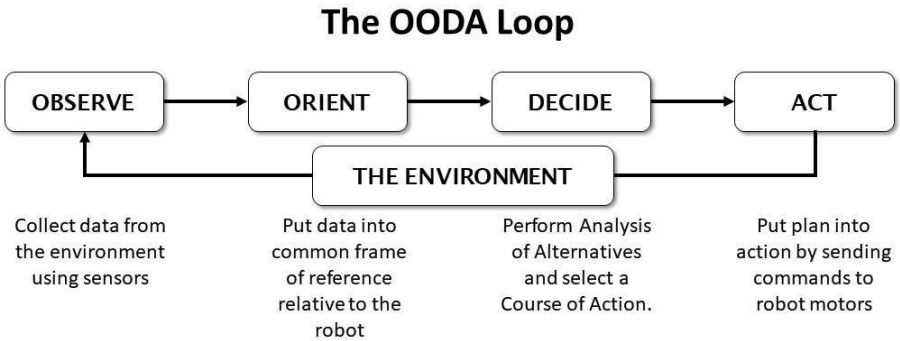
What you will learn

Not all of the chapters or topic in this book are considered *classical* AI approaches,but they do represent different ways of approaching machine learning and decision-makingproblems.

Building a firm foundation for robot control by understanding control theory and timing. We will be using a soft real-time control scheme with what I call a frame-based

control loop. This technique has a fancy technical name – rate monotonic scheduling—but I think you will find the concept fairly intuitive and easy to understand.

At the most basic level, AI is a way for the robot to make decisions about its actions. We will introduce a model for decision making that comes from the US Air Force,called the OODA (Observe-Orient-Decide- Act) loop. Our robot will have two of these loops: an inner loop or introspective loop, and an outward looking environment sensor loop. The lower, inner loop takes priority over the slower, outer loop, just as the autonomicparts of your body (heartbeat, breathing, eating) take precedence over your task functions(going to work, paying bills, mowing the lawn). This makes our system a type of subsumption architecture in Chapter 2, *Setting Up Your Robot*, a biologically inspired control paradigm named by Rodney Brooks of MIT, one of the founders of iRobot and designer of a robot named Baxter.



The **OODA loop** was invented by Col. John Boyd, a man also called *The Father of the F-16*. Col. Boyd's ideas are still widely quoted today, and his OODA loop is used to describe robot artificial intelligence, military planning, or marketing strategies with equal utility. The OODA provides a model for how a thinking machine that interacts withits environment might work.

Our robot works not by simply doing commands or following instructions step bystep, but by setting goals and then working to achieve these goals. The robot is free to setits own path or determine how to get to its goal. We will tell the robot to *pick up that toy* and the robot will decide which toy, how to get in range, and how to pick up the toy. If we, the human robot owner, instead tried to treat the robot as a teleoperated hand, we wouldhave to give the robot many individual instructions, such as move forward, move right, extend arm, open hand, each individually and without giving the robot any idea of why wewere making those motions.

Before designing the specifics of our robot and its software, we have to match its capabilities to the environment and the problem it must solve. The book will introduce some tools for designing the robot and managing the development of the software. We willuse two tools from the discipline of systems engineering to accomplish this – use cases andstoryboards. I will make this process as streamlined as possible. More advanced types of systems engineering are used by NASA and aerospace companies to design rockets and aircraft – this gives you a taste of those types of structured processes.