

INTRODUCTION TO AUTONOMOUS ROBOTS



Nikolaus Correll
University of Colorado at Boulder

University of Colorado at Boulder
Introduction to Autonomous Robots

Nikolaus Correll

This text is disseminated via the Open Education Resource (OER) LibreTexts Project (<https://LibreTexts.org>) and like the hundreds of other texts available within this powerful platform, it is freely available for reading, printing and "consuming." Most, but not all, pages in the library have licenses that may allow individuals to make changes, save, and print this book. Carefully consult the applicable license(s) before pursuing such effects.

Instructors can adopt existing LibreTexts texts or Remix them to quickly build course-specific resources to meet the needs of their students. Unlike traditional textbooks, LibreTexts' web based origins allow powerful integration of advanced features and new technologies to support learning.



The LibreTexts mission is to unite students, faculty and scholars in a cooperative effort to develop an easy-to-use online platform for the construction, customization, and dissemination of OER content to reduce the burdens of unreasonable textbook costs to our students and society. The LibreTexts project is a multi-institutional collaborative venture to develop the next generation of open-access texts to improve postsecondary education at all levels of higher learning by developing an Open Access Resource environment. The project currently consists of 14 independently operating and interconnected libraries that are constantly being optimized by students, faculty, and outside experts to supplant conventional paper-based books. These free textbook alternatives are organized within a central environment that is both vertically (from advance to basic level) and horizontally (across different fields) integrated.

The LibreTexts libraries are Powered by [NICE CXOne](#) and are supported by the Department of Education Open Textbook Pilot Project, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

Have questions or comments? For information about adoptions or adaptations contact info@LibreTexts.org. More information on our activities can be found via Facebook (<https://facebook.com/Libretexts>), Twitter (<https://twitter.com/libretexts>), or our blog (<http://Blog.Libretexts.org>).

This text was compiled on 01/13/2024

TABLE OF CONTENTS

Licensing

1: Introduction

- 1.1: Intelligence and Embodiment
- 1.2: A Roboticians' Problem
- 1.3: Ratslife
- 1.4: Challenges of Mobile Autonomous Robots
- 1.5: Challenges of Autonomous Manipulation
- 1.6: Exercises

2: Locomotion and Manipulation

- 2.1: Locomotion and Manipulation Examples
- 2.2: Static and Dynamic Stability
- 2.3: Degrees-of-Freedom
- 2.4: Exercises

3: Forward and Inverse Kinematics

- 3.1: Coordinate Systems and Frames of Reference
- 3.2: Forward kinematics of selected Mechanisms
- 3.3: Forward Kinematics using the Denavit-Hartenberg scheme
- 3.4: Inverse Kinematics of Selected Mechanisms
- 3.5: Inverse Kinematics using Feedback-Control
- 3.6: Exercises

4: Path Planning

- 4.1: Map Representations
- 4.2: Path-Planning Algorithms
- 4.3: Sampling-based Path Planning
- 4.4: Path Smoothing
- 4.5: Planning at different length-scales
- 4.6: Other Path-Planning Applications
- 4.7: Exercises

5: Sensors

- 5.1: Robotic Sensors
- 5.2: Proprioception of Robot Kinematics and Internal forces
- 5.3: Sensors Using Light
- 5.4: Sensors using Sound
- 5.5: Inertia-based Sensors
- 5.6: Beacon-based Sensors
- 5.7: Terminology
- 5.8: Exercises

6: Vision

- 6.1: Images as Two-Dimensional Signals
- 6.2: From Signals to Information
- 6.3: Basic Image Operations
- 6.4: Exercises

7: Feature Extraction

- 7.1: Feature Detection as an Information-Reduction Problem
- 7.2: Features
- 7.3: Line Recognition
- 7.4: Scale-Invariant Feature Transforms
- 7.5: Exercises

8: Uncertainty and Error Propagation

- 8.1: Uncertainty in Robotics as Random Variable
- 8.2: Error Propagation
- 8.3: Exercises

9: Localization

- 9.1: Motivating Example
- 9.2: Markov Localization
- 9.3: Particle Filter
- 9.4: The Kalman Filter
- 9.5: Extended Kalman Filter
- 9.6: Exercises

10: Grasping

- 10.1: The Theory of Grasping
- 10.2: Simple Grasping Mechanisms
- 10.3: How to Find Good Grasps?
- 10.4: Manipulation
- 10.5: Exercises

11: Simultaneous Localization and Mapping

- 11.1: Introduction
- 11.2: The Covariance Matrix
- 11.3: EKF SLAM
- 11.4: Graph-based SLAM

12: RGB-D SLAM

- 12.1: Converting Range Data into Point Cloud Data
- 12.2: The Iterative Closest Point (ICP) Algorithm
- 12.3: RGB-D Mapping

13: Trigonometry

- 13.1: Inverse trigonometry
- 13.2: Trigonometric Identities

14: Linear Algebra

- 14.1: Dot Product
- 14.2: Cross Product
- 14.3: Matrix Product
- 14.4: Matrix Inversion
- 14.5: Principal Component Analysis

15: Statistics

- 15.1: Random Variables and Probability Distributions
- 15.2: Conditional Probabilities and Bayes Rule
- 15.3: Sum of Two Random Processes
- 15.4: Linear Combinations of Independent Gaussian Random Variables
- 15.5: Testing Statistical Significance

16: How to Write a Research Paper

- 16.1: Original
- 16.2: Hypothesis- Or, What Do We Learn From this Work?
- 16.3: Survey and Tutorial
- 16.4: Writing it up!

17: Sample Curricula

- 17.1: An Introduction to Autonomous Mobile Robots
- 17.2: An Introduction to Autonomous Manipulation
- 17.3: Class Debates

[Index](#)

[Glossary](#)

[Detailed Licensing](#)

Licensing

A detailed breakdown of this resource's licensing can be found in [Back Matter/Detailed Licensing](#).

CHAPTER OVERVIEW

1: Introduction

Robotics celebrated its 50th birthday in 2011, dating back to the first commercial robot in 1961 (the Unimate). In a “Tonight Show” from the time, this robot did amazing things: it opens a bottle of beer, pours it, puts a golf ball into the hole, and even conducts an orchestra. This robot does all what we expect a good robot to do: it is dexterous, it is accurate, and even creative. Since this robot’s appearance on the Tonight show, more than 50 years have passed — so how incredible must be the capabilities of today’s robots and what must they be able to do?

Interestingly, we just recently learned doing all the things demonstrated by Unimate autonomously. Unimate indeed did what was shown on TV, but all motions have been preprogrammed and the environment has been carefully staged. Only the advent of cheap and powerful sensors and computation has recently enabled robots to detect an object by themselves, plan motions to it and grasp it. Yet, robotics is still far away from doing these tasks with human-like performance.

This book introduces you to the computational fundamentals of autonomous robots. Robots are autonomous when they make decisions in response to their environment vs. simply following a pre-programmed set of motions. They achieve this using techniques from signal processing, control theory, and artificial intelligence, among others. These techniques are coupled with the mechanics, the sensors, and the actuators of the robot. Designing a robot therefore requires a deep understanding of both algorithms and its interfaces to the physical world.

The goals of this introductory chapter are to introduce the kind of problems roboticists deal with and how they solve it.

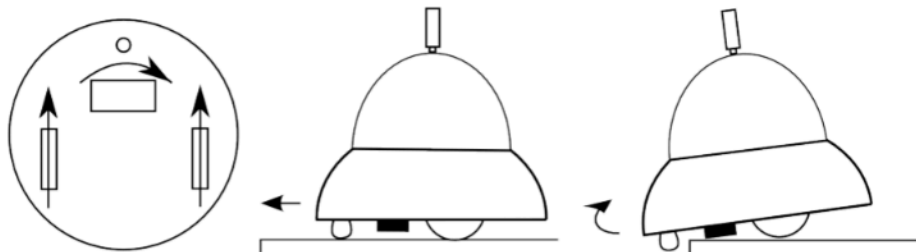


Figure 1.1: A wind-up toy that does not fall off the table using purely mechanical control. A fly-wheel that turns orthogonal to the robot’s motion induces a right turn as soon as it hits the ground once the front caster wheel goes off the edge.

[1.1: Intelligence and Embodiment](#)

[1.2: A Roboticians’ Problem](#)

[1.3: Ratslife](#)

[1.4: Challenges of Mobile Autonomous Robots](#)

[1.5: Challenges of Autonomous Manipulation](#)

[1.6: Exercises](#)

This page titled [1: Introduction](#) is shared under a [CC BY-NC 4.0](#) license and was authored, remixed, and/or curated by [Nikolaus Correll](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform; a detailed edit history is available upon request.

1.1: Intelligence and Embodiment

Our notion of “intelligent behavior” is strongly biased by our understanding of the brain and how computers work: intelligence is located in our heads. In fact, however, a lot of behavior that looks intelligent can be achieved by very simple means. For example, mechanical wind-up toys can avoid falling off an edge simply by using a fly-wheel that rotates at a right angle to their direction of motion and a caster wheel. Once the caster wheel loses contact with the ground—that is the robot has reached the edge—the fly-wheel kicks in and pulls the robot to the right (Figure 1.1).

A robot vacuum cleaner might solve the same problem very differently: it employs infrared sensors that are pointed downwards to detect edges such as stairs and then issues a command to make an avoiding turn. Once electronics are on-board, this is a much more efficient, albeit much more complex, approach.

Whereas the above examples provide different approaches to implement intelligent behaviors, similar trade-offs exist for robotic planning. For example, ants can find the shortest path between their nest and a food source by simply choosing the trail that already has more pheromones, the chemicals ants communicate with, on it. As shorter paths have ants not only moving faster towards the food, but also returning faster, their pheromone trails build up quicker (Figure 1.2). But ants are not stuck to this solution. Every now and then, ants give the longer path another shot, eventually finding new food sources

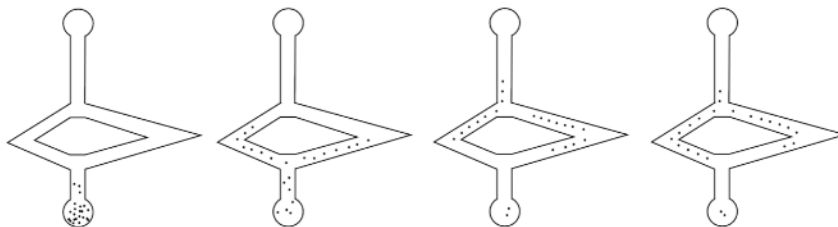


Figure 1.1.1: *Ants finding the shortest path from their nest (bottom) to a food source (top). From left to right: The ants initially have equal preference for the left and the right branch, both going back and forth. As ants return faster on the shorter branch there will be more pheromones present on the short branch once a new ant arrives from the nest.*

What looks like intelligent behavior at the swarm level, is essentially achieved by a pheromone sensor that occasionally fails. A modern industrial robot would solve the problem completely different: it would first acquire some representation of the environment in the form of a map populated with obstacles, and then plan a path using an algorithm. Which solution to achieve a certain desired behavior is best depends on the resources that are available to the designer. We will now study a more elaborate problem for which many, more or less efficient, solutions exist.

This page titled [1.1: Intelligence and Embodiment](#) is shared under a [CC BY-NC 4.0](#) license and was authored, remixed, and/or curated by [Nikolaus Correll](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform; a detailed edit history is available upon request.

1.2: A Roboticists' Problem

Imagine the following scenario. You are a robot in a maze-like environment such as a cluttered warehouse, hospital or office building. There is a chest full of gold coins hidden somewhere inside. Unfortunately, you don't have a map of the maze. In case you find the chest, you may only take a couple of coins at a time, and bring them to the exit door where your car is parked.

Query

Think about a strategy that will allow you to harvest as many coins in the shortest time as possible. Think about the cognitive and perception capabilities you would make use of. Now discuss alternative strategies, if you would not have these capabilities, i.e., what if you were blind, had no memory?

These are exactly the same problems a robot would have. A robot is a mobile machine that has sensors and computation, which allows it to reason about its environment. Current robots are far from the capabilities that humans have, therefore it makes a lot of sense to think about what strategies you would employ to solve a problem, if you were lacking important perception or computational capabilities.

Before we move forward to discuss potential strategies for robots with impeded sensory systems, let's quickly consider an optimal strategy. You will need to explore the maze without entering any branch twice. You can use a technique known as depth-first search to do this, but will need to be able to not only map the environment, but also localize in the environment, e.g., by recognizing places and dead-reckoning on the map. Once you have found the gold, you will need to plan the shortest path back to the exit, which you can then use to go back and forth until all the gold is harvested.

This page titled [1.2: A Roboticists' Problem](#) is shared under a [CC BY-NC 4.0](#) license and was authored, remixed, and/or curated by [Nikolaus Correll](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform; a detailed edit history is available upon request.