

ROBOTICS AUTOMATION COMPUTATION AND INTELLIGENCE IN ENGINEERING

¹ R. NIVEDHA, B.Tech, M.Tech

¹Department of Computer Science and Engineering,

¹Golden Valley Integrated Campus (Affiliated to JNTU Anantapur).

Andhra Pradesh

ABSTRACT: *Most introductory robotics textbooks have been written from the mechanical engineering perspective. These texts spend hundreds of pages studying gears, motors, sensors, and other related topics. While a computer scientist needs to know something about these topics, he or she certainly is not concerned with them to the degree that the texts cover. Meanwhile, the texts leave out many of the topics that are the most interesting to computer scientists. This paper presents some ideas, resources, and references for those who may want to teach a Junior/Senior level Introduction to Robotics course from a computer science perspective without a textbook.*

INTRODUCTION

The field of robotics is still relatively new, and the robotics community continues to struggle over its scope. In much the same way that computer science grew out of mathematics and electronic engineering, robotics is emerging from computer science, applied mathematics, and mechanical engineering. Different disciplines view robots from different perspectives. For example, consider a robot arm programmed to use a video camera to view parts randomly dropped onto a conveyer belt, identify those parts that are misshapen, and quickly move the arm into position to pick up and discard the irregular parts. The interesting part of this system to the computer scientist is how to recognize the misshapen part in the first place, and quickly identify its location. Applied mathematicians have developed equations that compute the location of the robot manipulator (“the hand on the arm”) given the relative angles of its joints (forward kinematics) as well as the joint angles necessary to move the manipulator to a particular location in space (inverse kinematics). The mathematician must compute where the part will be (as the conveyer belt moves) by the time the robot can arrive at the part, and the kinematic equations necessary to place the arm at that location. Meanwhile, the mechanical engineer is concerned with ensuring that the motors and the gears are robust and quick enough to accomplish the task.

Historically, most robotics textbooks have been written from the mechanical engineering perspective. These texts spend hundreds of pages studying gears, motors, sensors, and other related topics. While a computer scientist needs to know something about these topics, he or she certainly is not concerned with them to the degree texts cover. When I began searching for a textbook for to use in a Robotics class, I was disappointed at the lack of books written for computer science students on the market. This paper is the result of my experience teaching this course without a textbook to cover the core material, and presents some ideas, resources, and references for others who may want to teach such a course.

Some large universities have both the demand and the resources to teach courses like Robotics, AI, and Computer Vision every semester, or at least every year. Under these circumstances, requiring AI as a prerequisite for a Robotics class may be appropriate. But often in smaller departments courses are not offered with enough frequency to achieve this, and it is certainly possible to teach Robotics without assuming prior AI experience.

The Robotics class at Rowan university is designed for Junior and Senior computer science majors, and expects these students to have a solid mathematical background (including some calculus and linear algebra), as well as a basic background in programming, data structures, and algorithms. My goal is to provide students with an overview of the field of robotics, with particular emphasis on software for robotics.

The syllabus for this three-credit course (meeting twice a week for one hour and fifteen minutes) includes the following topics:

- § Mathematics for Robotics
- § Sensors & Actuators
- § Practical Robot
- § Laboratory Introduction
- § AI for Robotics
- § Robot Teleoperation
- § Case Studies

The remainder of this paper will review each of these topics and present ideas, resources, and selected references to aid the development of a Robotics class for computer science majors.

All of the recommended readings included in this paper are available on the web. In order to fully cover all the suggested topics, the instructor will have to supplement these readings with class notes or short excerpts from textbooks, and selected other papers.

While not all of the recommended readings are downloadable for free, I have limited myself to two electronic databases that charge a fee: ACM’s Digital Library, and IEEE’s Xplore. Many college and university libraries subscribe to these databases. Furthermore, the majority of the papers that are not freely available have copyright notices that allow copying for educational purposes (and thus, only incur a one-time fee when downloaded by the instructor). I also encourage instructors to consider requiring students to join the ACM as student members (a worthwhile membership for any computer science student), giving students access to the entire digital library and a host of other benefits, for much less than the cost of a single textbook.

MATHEMATICS FOR ROBOTICS

We begin the semester by discussing homogeneous transformations, and coordinate frames. We move on to cover forward kinematics, but skip the more complex inverse kinematics. Alonzo Kelly published an excellent technical report [15] that gives a very careful description of coordinate frames and transformations, together with forward and inverse kinematics and more. I recommend that you only cover pages 1-16: coordinate frames and forward kinematics.

Later in the semester, we discuss calibration (including hand-eye calibration and calibration for mobile robots), triangulation ranging, and position estimation.

SENSORS & ACTUATORS / PRACTICAL ROBOTIC LABORATORY INTRODUCTION

While our course does not include a weekly laboratory period, students cannot have an understanding of robotics until they have tried to program their own robots. Our Robotics class includes laboratory exercises that the students do outside of the classroom in a small laboratory we have equipped. We use the Handy Board [8], a 6811-based microcontroller system that provides input ports for sensors and output ports for motors. Students build a robot base to support the Handy Board using Lego bricks, gears, motors and wheels, and write software in Interactive C [12]. Kumar and Meeden used a similar setup for an Artificial Intelligence course, which they describe in [18]. They also have an extensive on-line resource kit that provides more detail [28].

A less expensive option for a laboratory is the Lego Mindstorms [20] system. The Lego Mindstorms kit comes with its own visual programming environment, which is almost certainly too limited for use in this type of course. However several open source alternatives exist for programming the Mindstorms in a wide variety of languages including Ada, C, Scheme, and Java. O'Hara and Kay provide a survey of open source software for educational robotics in [24]. Patterson-McNeill and Binkerd discuss the Mindstorms environment in more detail in [25]. Many others have reported their experiences using Mindstorms in more introductory computer science courses. [2] [6] [7] [34]

More advanced robots such as Kheperas [13] and Pioneers [29] are certainly also an option, though the price is probably beyond that of most smaller departments. Harlan, Levine, and McClarigan have developed a class interface for use with Kheperas [9].

In the first few weeks of class I generally alternate lectures on mathematics for robots with an introduction to the hardware we will use in class. This naturally leads into a discussion of sensors and actuators, both those included in student kits as well as more sophisticated versions.

Both the Handy Board and Mindstorms systems require students to have some competency with Lego design. Fred Martin has written an excellent introduction to Lego design [21] that really helps the student (or instructor) who may not have had extensive experience with Lego as a child.

Fred Martin has also written a robotics textbook [22] that I used in one of my classes. While the text has some very helpful sections, most of the contents are geared more towards engineers. It is a very useful book to have on reserve in the library, but I have no plans to require students to purchase it for future classes.

AI FOR ROBOTICS

My goals in this section are to teach students something about learning, planning, and search. I also like to include a section on Neural Networks so that I can include a Neural Network-based mobile robot in my case studies section.

I tend to use a lot of class notes for this section, supplemented with small excerpts from a few AI textbooks. In addition, Ciesielski & McDonald have developed a tool that is available on the web [1]. This tool allows you to view a graphical demonstration of several search algorithms. The web site also includes a set of slides that you can use to help teach AI search techniques to your students. Ciesielski & McDonald have also written a paper describing their tool [4].

The D* algorithm is one that I enjoy teaching. Stentz's paper on D* [31] is not one that you can give to an undergraduate and expect them to follow the algorithm. However, I like the ideas behind the algorithm so much, that I do use this paper. I encourage the students to listen to my lecture before trying to follow the paper, and rather than require them to understand the algorithm completely, I am more concerned with them understanding why there is a need for such an algorithm, and the general way it operates.

Kevin Knight wrote an article for CACM [17] introducing Neural Networks that is written for a general computer science audience. I generally introduce this topic immediately before the Neural Network example in the case studies section below.

While I use AI as only a part of my course, others have taught entire AI courses using Robotics. In addition to Kumar and Meeden's work mentioned above [18], both Klassner [16] and Imberman [11] have used Robots in the context of their Artificial Intelligence courses. Computer scientists who wish to teach an entire course on AI for Robotics may also wish to consider Robin Murphy's textbook. [23]

ROBOT TELEOPERATION

With the popularity of battling robots on television today, it is worth taking some time to discuss with students the ideas behind robot teleoperation. Students are surprised that I am not a fan of these shows. While I find the robots physically interesting, they are so lacking in intelligence as to not be of much interest to me.

I am, however, tremendously interested in semi-autonomous teleoperation. Sharon Laubach presents the teleoperation (using 3D data) of the Sojourner rover from the NASA Mars Pathfinder mission in a paper from ACM Crossroads [19]. Kay and Thorpe present a 2D alternative in their paper describing the STRIPE system [14].

Methods

The diversity of robot research provides an enormous selection of technical papers. The difficulty, of course, is finding papers that are well written and suitable for undergraduates.

A relatively recent CACM special issue on Robotics has several worthwhile papers in it. Rodney Brooks considers issues in humanoid robots in his paper [3], primarily through two examples (Kismet and Cog) from MIT's AI lab. The paper includes some interesting discussion of human vision. Along with this article you might want to reference the Humanoid Robotics group at MIT [10], or other humanoid robot projects mentioned in the paper.

By building bigger robots out of smaller modules, Rus et al [30] describe how robots can reconfigure themselves for different tasks, and can cope with the failure of an individual module. Along with this article, you might want to reference the Dartmouth Robotics Lab web site.[5]

Sebastian Thrun's paper on Probabilistic Robotics [32] requires that students have a strong background in probability, but I think it is a good one nonetheless. Students in most classes will have experience in the impreciseness of their robots, and will be greatly relieved to find out that techniques for coping with a range of possible locations exist.

Manuela Veloso's paper on Entertainment Robotics [33] is a fun paper that your students will love. If you have any sort of a competition of your own, this will give students a glimpse at the next level. The focus is on Robot soccer and the Robocup [27] competitions.

I believe that Dean Pomerleau's use of Neural Networks in Robotics [26] is worth taking a look at, time permitting. While this paper is getting on in years, it is the most accessible reference that I've found that gives an overview of some of the possibilities of the field.

CONCLUSION

Robotics is an interdisciplinary branch of engineering and science that includes mechanical engineering, computer science and others. Robotics deals with the of robots, as well as computer systems for their control, sensory feedback, and information processing. These technologies are used to develop machines that can substitute for humans and replicate human actions. Robots can be used in any situation and for any purpose, but today many are used in dangerous environments (including bomb detection and de-activation), manufacturing processes, or where humans cannot survive. Robots can take on any form but some are made to resemble humans in appearance. This is said to help in the acceptance of a robot in certain replicative behaviors usually performed by people. Such robots attempt to replicate walking, lifting, speech, cognition, and basically anything a human can do. Many of today's robots are inspired by nature, contributing to the field of bio inspired robotica

REFERENCES

- [1] David J. Barnes, Teaching Introductory Java through LEGO MINDSTORMS Models, Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, 2002. Available <http://doi.acm.org/10.1145/563340.563397>
- [2] Rodney Brooks, Humanoid Robots, Communications of the ACM, Vol. 45, No. 3 (March 2002). Available at: <http://doi.acm.org/10.1145/504729.504751>
- [3] Vic Ciesielski and Peter McDonald, Using Animation of State Space Algorithms to Overcome Student Learning Difficulties, in Proceedings
- [4] Innovation and Technology in Computer Science Education, 2001. Available at: <http://doi.acm.org/10.1145/377435.377495>
- [5] Dartmouth College Robot Lab, <http://www.cs.dartmouth.edu/~robotlab/index.html> [accessed March 2003].

