Teaching Statement

Ross A. Knepper

1 Teaching

I enjoy teaching a wide variety of computer science and engineering subjects such as artificial intelligence and feedback control. At Cornell, my teaching has focused on robotics. Each course I have offered at Cornell was newly developed to expand the breadth and depth of Cornell's robotics curriculum. I worked in collaboration with robotics faculty across departments to introduce an undergraduate robotics minor at Cornell in 2019. I also worked to introduce an interdisciplinary robotics Ph.D. program at Cornell, which is anticipated to matriculate its first class by 2022.

1.1 Philosophy

Education is an active process that requires continual engagement from students. I believe in creating an environment where all students take away something of value, regardless of background or identity. Such an environment requires an organized curriculum with clear expectations and incentives. An engaged, curious, open mind is best prepared to learn new ideas. Students possess many different preferred learning styles, so I include a variety of learning and assessment styles without weighting one above the others. These include reading assignments, reading mini-quizzes, problem sets, open-ended projects, and exams.

I believe in the importance of project work to take students beyond memorization of facts and into the practicing of transferable skills. Appropriately open-ended problems, where more than one solution is possible, engage and excite students by giving them a sense of ownership in their work. This type of work has special significance because it helps prepare students for the open-endedness of work in the real world after graduation. Students learn from both successes and failures. Often, the failures are the most educational. Thus, I encourage students to correct mistakes they make on exams for partial credit. I invite feedback from the students about my own failures and opportunities for improvement. I allow students to see that I am both imperfect and responsive to their feedback.

1.2 Approach

All of my undergraduate courses involve short, preparatory, pre-lecture readings. I give a short reading comprehension quiz before the start of class to incentivize doing the reading assignments. I lecture using the blackboard in order to pace myself (so that note-takers can keep up) and facilitate opportunistic asides. My lectures involve frequent class discussions, and I encourage an atmosphere in which all students are comfortable participating (both asking and answering questions) by rewarding thoughtful effort rather than right answers. I also frequently use live demonstrations and props. I use activities during lecture to get the students engaged by working together on problems in small groups before discussing them as a whole class. In the event that an explanation does not make sense to a student, I pride myself on being able to explain a concept in several different ways to find one that is clear. This is time well spent in lecture because for each student who speaks up when they don't understand, many others do not. I want to ensure that every student walks away having learned something, so I focus on both high-level concepts and technical details.

1.3 Intellectual Development

An important perspective on teaching methodology comes from understanding the process of intellectual development. Several models have been proposed to explain students' treatment of information as they mature intellectually. Perry [4], Belenky et al. [2], and Baxter Magolda [1] each propose a similar model of progression in intellectual development described in terms of how the student treats knowledge. Baxter Magolda's version proposes intellectual phases deemed Absolute Knowledge (naive), Independent Knowledge, and Contextual Knowledge (nuanced). This final phase is most conducive to critical thinking. The undergraduate student cohort includes individuals at all three phases. Not all students progress through the phases as they mature. To effectively deliver material to college students, the teacher should understand both the students' background knowledge and current phase of development. But more importantly, the teacher can help to advance students through the stages of intellectual development, thus having impacts well beyond one subject area. Teachers achieve such advancement by challenging students to understand where ideas come from and to develop and articulate their own arguments and ideas. Such skills are essential to success in any engineering job in industry or academia.

1.4 Cornell Teaching

Since coming to Cornell, I have developed or co-developed four new courses. At the undergraduate/M.Eng. level, "CS 4752/5752: Robotic Manipulation" has been offered twice, "CS 4750/5750: Foundations of Robotics" has been offered four times, and I co-taught "CS 4754/INFO 4410: Human-Robot Interaction" with Prof. Malte Jung once. At the graduate level, "CS 6751: Introduction to Robotic Mobile Manipulation" has been offered four times.

1.4.1 CS 4752/5752: Robotic Manipulation

This course encourages students to think broadly about what it means to manipulate an object and how a robot can accomplish it. The course is driven by projects implemented on a two-armed $30k Baxter robot from Rethink Robotics. The students learn to use ROS (the Robot Operating System) and MoveIt! to control the robot and perform tasks such as stacking blocks, writing on a whiteboard, and playing ball games. The culmination of the course is a soccer-like tournament played on a table-top game board (Figure 1), in which Baxter's two arms are controlled by the two teams, respectively. The students learn kinematics, dynamics, control, and a little computer vision. To help students relate to the concept of manipulation, each student is asked to showcase a special manipulation skill they have, such as juggling, twirling a pencil, or pouring a bowl of cereal. In every case, we discuss what special manipulation strategies are employed beyond pick-and-place.

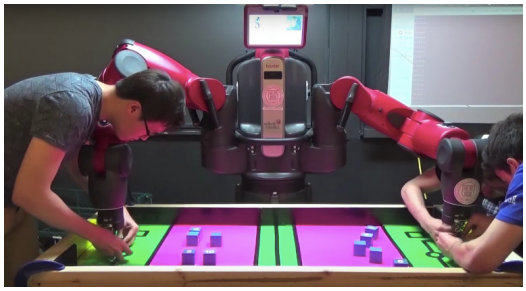

Figure 1: In CS 4752/5752, the students learn robotic manipulation hands-on with Baxter. Instead of a final exam, the students compete in a final competition that shows off the cumulative capabilities they have implemented throughout the semester. The game is similar to soccer, where each team controls one of Baxter's arms to defend their own goal and score on the opponent.

1.4.2 CS 4750/5750: Foundations of Robotics

This course offers a broad, mathematical/algorithmic introduction to robotics. I created it when I recognized that the undergraduate robotics electives at Cornell had to teach a lot of the same background math. It was designed to scale to a large class size, so I made the difficult decision to

forgo real robot hardware in favor of simulation. The course includes units on robot mechanisms, simulation and numerical methods, kinematics, uncertainty, optimization, and control. Since none of the existing robotics textbooks cover all of these topics at an introductory level, I and my course staff effectively wrote our own textbook, which we refer to as "course notes". The pre-lecture assigned readings come from this text.

The course has a significant, open-ended, project-based component. Projects are implemented in Python using ROS in the V-REP simulator (Figure 2). The students learn to work with numerical and optimization tools such as NumPy and CVXOPT. Due to its open-endedness, the project component is the most critical component of the course in terms of both student learning and student satisfaction.

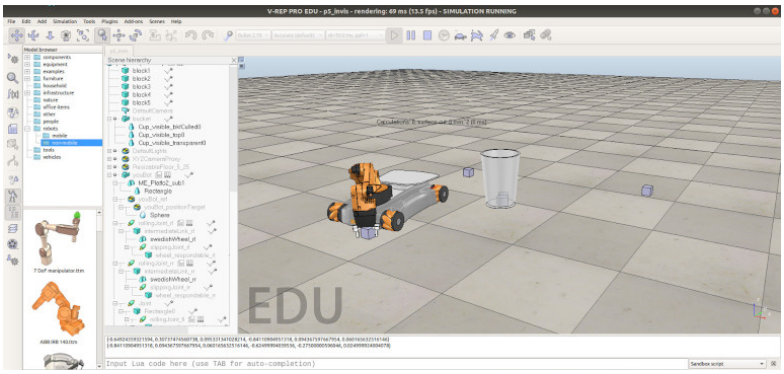

Figure 2: In CS 4750/5750, the students gain a mathematical introduction to robotics. The V-REP simulator gives the students access to sophisticated robots. They learn kinematics, uncertainty, optimization, and control. In the project shown here, the students program the robot to gather blocks and put them in the basket.

The course is designed to function as the first in a series, with advanced electives requiring it as a prerequisite. As such, it is important that the course be accessible to the broadest possible range of students from diverse backgrounds. We require a fairly minimal set of classes (calculus in 3D, linear algebra, and an introductory programming course). The objective is to bring all students from all backgrounds up to a basic level of competence across a suite of topics. The effort to include this diverse audience has profoundly affected the way that I teach the course, as I describe below in Section 1.5.

1.4.3 CS 4754/INFO 4410: Human-Robot Interaction

I co-taught the undergraduate-level Human-Robot Interaction course once with Prof. Malte Jung (Cornell, Information Science) as a voluntary overload teaching assignment. This course examines the technical and design aspects of doing work in HRI. The students started by observing a worker (for example, a cafeteria worker) doing a job, and then they imagined how a robot could help improve the workflow in that job. For one project, the students designed and built a robot to help with a task or used my Baxter robot (which was simultaneously being used for my Robotic Manipulation course). The course includes many active-learning exercises to help stimulate open-mindedness and good design instincts. For example, another project asked students to build a passive "robot" that elicits human interaction without any sensors, computation, or actuators. The robots that the students built traveled all over campus, created dialog among strangers, and facilitated connections within the campus community.

1.4.4 CS 6751: Introduction to Robotic Mobile Manipulation

Mobile manipulation combines the problems of navigation and manipulation. This combination compounds challenges like uncertainty and complex environments. It also enables many exciting applications, such as cooking and furniture assembly in unstructured human environments. The course covers many topics within robotics, including planning, control, perception, learning, and systems. The course is organized as a mock program committee meeting, in which the

readings — classic papers in robotics — are reviewed and discussed using Microsoft CMT and in person. Ultimately, students decide which papers get accepted to our mock conference. In parallel, students complete their own course projects and develop papers suitable for submission to a robotics conference.

1.5 Insights on Teaching a Diverse Cohort

Robotics is an application, not a discipline. As such, it borrows tools and techniques from many disciplines, including math, statistics, operations research, mechanics, engineering, and computer science. This diversity is a strength of the robotics community. However, it creates a significant obstacle to teaching robotics. Students from many different majors enroll in my courses, bringing with them different subsets of the tools they need to know. As an instructor, it would be appealing to add prerequisites to cover all the tools so that I can depend on a high level of common knowledge among students. However, the number and diversity of tools means that the reach of robotics courses would be restricted to the most privileged students. I see my teaching mission as reaching, educating, and inspiring a cohort of students from diverse disciplines, backgrounds, identities, and life experiences.

My Foundations of Robotics course (Section 1.4.2) embodies this approach. Since the beginning, it has enrolled students from many majors, including computer science, information science, mechanical and aerospace engineering (MAE), and electrical and computer engineering (ECE). The course teaches much of the math and programming required to do the robotics work. However, the quest for broader reach also comes with challenges, as I discovered when I cross-listed the course.

The third offering of Foundations of Robotics, taught during the fall 2018 semester, was cross-listed in two new departments (MAE and ECE) for the first time. Each major has many students interested in robotics. Even though the course previously had seen many students sign up from those majors, those students knew they were signing up for a CS course. The choice to cross-list had a profound effect on the expectations and preparation of the students who enrolled, since they were now signing up for a course in their own department rather than a CS course. This group was the most diverse I had taught. However, these students struggled in new ways, which led me to major new insights about how academically-diverse students learn. I discuss several of these insights below.

Struggling students. An early warning sign that something had changed came in the form of an increase in students struggling with parts of the course. I monitor students who are not submitting assignments. Some students are brave enough to approach me either before or after I reach out to them. Sometimes, they are hung up on conceptual issues like mechanics; other times it is programming. When meeting with students, it is important to triage why they are struggling. Common causes include mental illness, low self-confidence, and lack of preparation. Each of these causes requires different forms of assistance. I meet regularly for one-on-one meetings with any student who needs extra help. At times, this can be a significant time commitment, but it yields advantages: it helps me stay grounded with where the students are in their learning progression, and often a little extra help is enough to catch up a student by getting past a few conceptual stumbling blocks.

Collaboration styles. Another new phenomenon that emerged after cross-listing involved group work. The course policy encourages students to work in groups, since research shows that students learn material more effectively when they are placed in the position of teaching concepts to one another [3]. With cross-listing, there was also an increase in academic integrity violations,

particularly on the course projects. Some students coming from the engineering majors were struggling with the coding assignments, but this fact alone did not fully explain why they would cheat rather than seek out help. As a computer scientist, I am trained to communicate abstract algorithms. In contrast, these students had no language to describe an algorithm other than the code itself. Consequently, in sharing high-level ideas, students ended up effectively sharing code as well, leading some of them to submit nearly identical code.

To bolster students' ability to complete the programming projects successfully, I made significant structural changes to the course. These changes include more lectures on the practical aspects of programming robots, giving more examples in lecture of how to use the math taught in the course, a gentler learning curve as the programming projects ramp up, and assigned groups for projects. I now carefully craft the assigned groups to meet several important criteria. There is significant evidence that randomizing the groups in such a way leads to a fairer experience for several categories of students who face additional obstacles beyond the learning of the materials. I randomize the groups subject to a ranked list of criteria:

1. assign three students to a group
2. ensure women and underrepresented minorities (URMs) are not isolated within a group
3. diversify majors within a group
4. balance within each group to have a range of GPAs

Clustering the women and URMs ensures that although they are in a minority of the total groups, they form a majority with each group they are in. These identities are often associated with stereotypes of being worse at math, computer science, and engineering than white men. When faced with adversity on an exercise that stereotypes predict an individual would perform worse at, such students are more likely to disengage [6]. Research shows that when an individual in one of these categories is alone on a team, he or she is more likely to succumb to the stereotype and drop out, thus reducing learning outcomes [5]. By forming these majority-minority groups, every student plays a more active role in the projects and all are empowered to achieve better learning outcomes.

The remedies I describe above have achieved several ancillary benefits. They have decreased the incidence at which students report boredom with the course. Furthermore, they have reduced course attrition overall as well as the tendency for women and underrepresented minorities to drop the course at higher rates. Finally, the number of academic integrity violations has dropped to near zero.

1.6 Courses I Would Be Excited to Teach

I have extensive experience teaching **robotics** at all levels. I design and cross-list my courses with the intent of reaching students of computer science, electrical engineering, mechanical engineering, operations research, and others. I am qualified to teach many other courses at the undergraduate level along this spectrum of majors, including **machine learning**, **artificial intelligence**, **operating systems**, **computer networks**, **computer architecture**, **embedded systems**, **optimization**, **simulation and numerical methods**, **mechatronics**, **statics**, **dynamics**, and **control**. My breadth in systems and architecture stems from my experience as a developer of the Tru64 Unix operating system at Compaq and Hewlett-Packard.

Within robotics, I would be excited to teach many courses at the undergraduate or graduate level, including: **human-robot interaction**, **planning**, **manipulation**, **robot learning**, **foundations of robotics**, **robot systems**, and **kinematics, dynamics, and control**.

2 Advising

I have advised ten Ph.D. students, of whom two have graduated and gone on to serve as postdocs (at the University of Texas at Austin and the University of Washington). I have also graduated two M.S. students. I have served on 24 Ph.D. thesis committees, including two external to Cornell. Ten of these students have graduated.

My advising style for postdocs and graduate students is rooted in the importance of students learning from their own experiences by making their own choices. I work closely with them, but I rarely tell students what to do. I share both my wisdom and my joy for research and teaching by treating students as equals and leading by example. We discuss directions to take with the work, and we freely share ideas with each other. Ultimately, the student decides how to proceed with their own work. The sense of investment that my students feel in their work makes both successes and failures feel like learning experiences. I counsel my students that graduate school is designed to be a safe place to fail and that if they never fail at anything, they should be taking bigger risks. It is too soon to assess the long-term effectiveness of my advising style, but both students that I graduated followed an academic track by taking postdoctoral positions in prestigious labs.

I love every aspect of my job, and I make all aspects transparent to my Ph.D. students because many of them are considering pursuing faculty jobs themselves. For example, although my students do not need to worry about where their funding is coming from, I allow students to opt in to assisting in writing grant proposals and preparing deliverables for grants. Allowing students to provide intellectual and writing input encourages a sense of buy-in with the work; three of my awards, totaling over $950k, stem from particularly strong student involvement.

Lab culture is very important to me. I have found a synergistic sweet spot around six to eight Ph.D. students, plus around ten undergraduate RAs. I hold a weekly catered lab lunch for both grads and undergrads to give students a chance to connect. The lunches feature technical presentations, lightning research updates, and culture days. In the latter, we talk about non-technical issues of importance to society. I started a weekly paper club for my group to help bridge research topics. I choose not to attend these meetings because it gives the students a greater sense of ownership of the institution and the choice of topics. The students sometimes invite me to sit in after an initial discussion to offer extra context.

3 Mentoring

I serve as a mentor for many projects at the undergraduate and graduate levels. I contribute expertise in robot algorithms to these projects. Four undergraduate researchers in my lab have been awarded grants from the Engineering Learning Initiatives program at Cornell. I have served as a mentor to six students in the Hunter R. Rawlings III Cornell Presidential Research Scholars (RCPRS) program.

Figure 3: The undergraduate Autonomous Solar Airship project team.

One major, multi-year project I organized and led at Cornell is the Autonomous Solar Airship project (Figure 3). A team of more than a dozen undergraduates built and automated a five-meter helium blimp for use in applications such as biology and conservation.

4 Program Development

I have co-organized two new degree programs while at Cornell. In collaboration with faculty colleagues in the departments of Mechanical and Aerospace Engineering (MAE), Electrical and Computer Engineering (ECE), and Information Science (IS), I established an undergraduate minor in robotics that started in January 2019. The minor has depth and breadth course requirements that require a robotics student to take courses across multiple departments, stressing the interdisciplinary nature of robotics. The minor has already generated great interest from students across these departments.

I also helped lead an effort to create a new Robotics Ph.D. program at Cornell. This will be an innovative program that students may apply to through any of the traditional departments: CS, IS, MAE, and ECE. After it is approved by the New York legislature, the program is expected to matriculate its first class by 2022. These students will major in robotics and minor in the field of the department through which they applied. A unique feature of the Robotics Ph.D. is its inclusion of community engagement as an explicit curricular component. Community engagement is broadly construed to include any activity that brings a benefit to society, such as K-12 education, outreach, technology transfer, and broader impacts of research.

5 Outreach

The many benefits of technology are spread very inequitably. Part of my role as a technologist is to bring those benefits to everyone. The major challenges that society faces include lack of access to technology due to poverty or circumstance, as well as societal biases that cause disengagement. Many societal factors, such as scarce opportunities and peer pressure, steer a child's path of development. I am committed to helping children see a broader scope of possible future careers, and I demonstrate this commitment by running regular outreach events targeting middle- and high-school students.

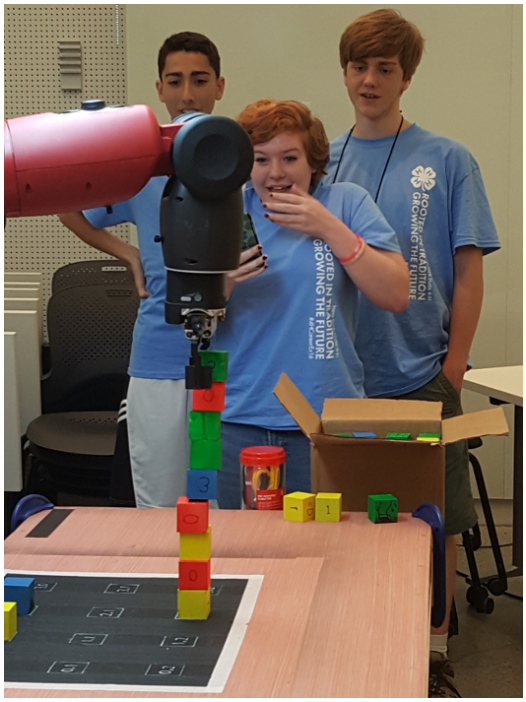

Figure 4: In partnership with 4-H of New York State, I have run an annual Career Explorations event to teach computational thinking using robotics. The program, which encourages creative exploration, targets students from underserved school districts throughout New York State.

To provide an example, since summer 2016, I have overseen an annual three-day event for high school students from underserved communities; this event is in partnership with 4-H of New York State as part of their Career Explorations program. My event is designed to teach students about careers in computing and robotics as well as give them hands-on experience using computational thinking skills. The students program a robot to do tasks such as stacking blocks or sorting them by color (Figure 4). Many students may be unaware that a career path in computer science or engineering is an option for them. I recruited a small staff of talented and devoted CS degree holders to run the event. Each member of the team started programming at a different age and under different circumstances. We illustrate through our personal stories that it is never too late to begin programming.

References

[1] Marcia B. Baxter Magolda. *Knowing and Reasoning in College: Gender Related Patterns in Students' Intellectual Development*. Jossey-Bass, San Francisco, 1992.

[2] Mary Field Belenky, Blythe McVicker Clinchy, Nancy Rule Goldberger, and Jill Mattuck Tarule. *Women's Ways of Knowing: The Development of Self, Voice and Mind*. Basic Books, New York, 1986.

[3] Aloysius Wei Lun Koh, Sze Chi Lee, and Stephen Wee Hun Lim. The learning benefits of teaching: A retrieval practice hypothesis. *Applied Cognitive Psychology*, 32(3):401–410, 2018.

[4] William G. Perry. *Forms of ethical and intellectual development in the college years: a scheme*. Holt, Rinehart & Winston, New York, 1970.

[5] Sue V Rosser. Group work in science, engineering, and mathematics: Consequences of ignoring gender and race. *College Teaching*, 46(3):82–88, 1998.

[6] Claude M Steele. *Whistling Vivaldi: How stereotypes affect us and what we can do*. WW Norton & Company, 2011.