Guiding the Development of Undergraduate Educational Robotics

Bryce Ikeda
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
bikeda@cs.unc.edu

Janine Hoelscher
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
jhoelsch@cs.unc.edu

Ron Alterovitz
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
ron@cs.unc.edu

Daniel Szafir
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
dszafir@cs.unc.edu

ABSTRACT

Educational robotics, in which students program a physical robot to interact with the real world, can provide tangible active learning opportunities that are often linked to increases in student computational thinking, creativity, and motivation. To date, ER has focused on the use of robots to augment learning of adjacent fields (e.g., mathematics, programming, digital media) for K–12 students. As a result, we lack ER guidelines for: (1) supporting college robotics students learning the discipline of robotics itself and (2) college robotics students, who may possess distinct abilities and needs compared to K–12 students. To address this gap, we present a semester-long exploration of a college-level *Introduction to Robotics* course. Through student feedback, we identified three themes: (1) Positive learning opportunities, (2) Dealing with uncertainty, and (3) Successful results with simple solutions. We detail these themes and provide guidelines for improving ER in the context of college students learning to program and debug robots for the first time.

CCS CONCEPTS

• **Social and professional topics → Computing education**; • **Computer systems organization → Robotics**.

KEYWORDS

Educational Robotics (ER); undergraduate introduction to robotics

ACM Reference Format:
Bryce Ikeda, Janine Hoelscher, Ron Alterovitz, and Daniel Szafir. 2023. Guiding the Development of Undergraduate Educational Robotics. In *Proceedings of the ACM Conference on Global Computing Education Vol 1 (CompEd 2023), December 5–9, 2023, Hyderabad, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3576882.3617928

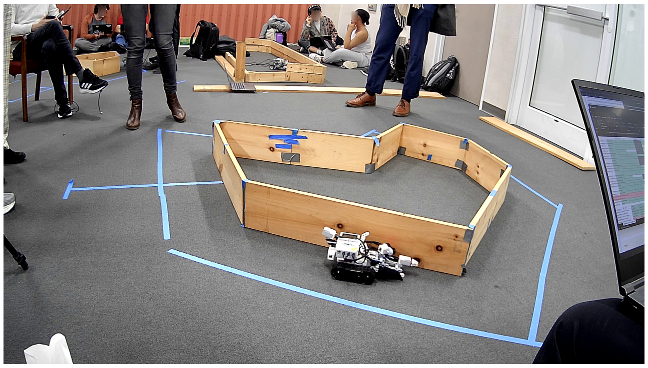


Figure 1: This study examines undergraduate students' experiences with learning and implementing robotics concepts in *Introduction to Robotics*. Here, a student evaluates the performance of their robot on a physical course.

1 INTRODUCTION

In Educational Robotics (ER) students program a robot to interact with the physical world. This hands-on approach motivates students to learn, leading educators to use robots as an educational tool to enhance students' understanding of subjects such as math and physics [2, 6, 28, 36].

To date, ER research has primarily focused on the impact of robotics in K–12 education. At these levels, robots are often utilized as a tool for teaching subjects in science, technology, engineering, and math (STEM). Furthermore, educators in these settings typically employ visual programming interfaces that lower the learning curve compared to traditional programming languages such as Python [7, 8, 27, 29]. Beyond K–12 education, colleges are increasingly offering undergraduate and graduate level programs focused on the discipline of robotics itself in recognition of the growing importance of this field to society. Higher education robotics differs from K–12 in that it supports college robotics students: (1) who are learning the discipline of robotics itself, and (2) who may possess distinct abilities and needs compared to K–12 students, necessitating different learning goals. Given these differences, it is yet to be seen whether guidelines for K–12 ER from prior work also apply to college-level ER.

To address this gap, we investigated the question: **how can ER curriculum and course instructors better support university robotics students during their learning process**? To this end, we explored the experiences of 20 computer science undergraduate students who learned to program a robot in a college-level *Introduction to Robotics* course. We administered surveys and semi-structured

---

CompEd 2023, December 5–9, 2023, Hyderabad, India

ACM ISBN 979-8-4007-0048-4/23/12…$15.00
https://doi.org/10.1145/3576882.3617928

verbal interviews to obtain feedback regarding the students' experiences during their laboratory assignments. To analyze our data, we used reflexive thematic analysis, a method for grouping data into relevant themes that answer our research question [5]. Finally, we present the resulting themes and guidelines to inform future development of undergraduate robotics courses.

2 BACKGROUND

Prior research has explored various strategies that support ER in K–12 classrooms. For example, Benitti, 2012 [3] and Xia et al., 2018 [35] both conducted ER literature reviews to understand the benefits of robots as an educational tool and compiled guidelines for future ER curricula. These guidelines emphasize the importance of providing ample space for robot experimentation [17], ensuring accessible resources for students [33], and developing well-structured activities so that the robot may guide students through the learning process [23]. These strategies illustrate how thoughtful development of ER curriculum may enhance learning for K–12 students. However, it remains unexplored whether these guidelines also apply to undergraduate ER curriculum. College students often possess a higher educational background, allowing them to explore complex robotics concepts that may require different types of support.

Recently, studies have begun to examine the effectiveness of robotics courses specifically designed for undergraduate students. These studies detail educators' experiences teaching and developing curricula for humanoid robots [14, 18], autonomous mobile robots [9, 12, 21, 26], and other STEM-related topics [1, 30, 34]. The insights from these experiences have led to suggestions for future ER curricula, such as using laboratory exercises along with in-class lectures [14] and to spend ample time teaching complex robotics concepts [18]. To date, Lalonde et al., 2006 [16] provides one of the only focused inquiries into how to develop future ER curriculum to support undergraduate students studying robotics. This study outlines strategies that include providing readily available debugging tools and giving students challenges that are within their reach. While their insights are valuable, more research is necessary to both confirm and expand upon their findings, motivating our work.

3 APPROACH

In this study, we followed 20 undergraduate students in the *Introduction to Robotics* course offered by our university's computer science department. Our focus was on four programming laboratory assignments, illustrated in Figure 2, with the goal of evaluating students' experiences while programming a robot.

3.1 Course Background

Introduction to Robotics is an undergraduate robotics course that teaches students the algorithms necessary for robots to utilize sensory information from the environment and purposefully act upon it. Students are taught topics in robot kinematics, actuation, sensing, configuration spaces, control, and motion planning. Once completed, students are expected to have an understanding of robotic systems, the ability to analyze and implement robotics algorithms, and an awareness of current challenges in the field of robotics. It is offered to undergraduate students who have taken *Foundations of Programming* and *Data Structures and Analysis*, and therefore already possess a foundation in programming. The course consists of twice-weekly lectures of 75 minutes each, six written homework assignments, four programming labs (Figure 2), a midterm exam, and a final exam. To facilitate additional student engagement, this course used Piazza, an online forum, as well as office hours offered by the instructor. A weekly breakdown can be found here: https://github.com/hri-ironlab/comped-2023-supplemental-materials.

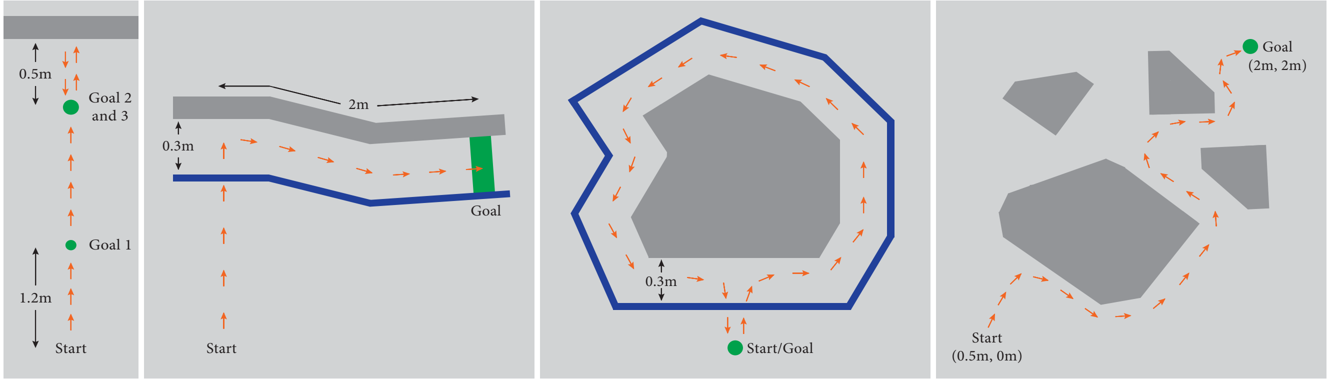
Our analysis focused on the programming labs. To begin, students worked in groups of one to three members, and were tasked with constructing a mobile robot using Lego Mindstorms EV3 robot sets that included three motors with built-in rotation sensors, two touch sensors for collision detection, an ultrasonic sensor for measuring distances, and a gyroscope for estimating orientation. To program the robot, students used MicroPython, a Python module with a Visual Studio Code extension designed for programming Lego robots. The students were also provided with a dedicated lab space for building custom obstacle courses to test their robot programs. Students began each lab with a pre-lab assignment instructing them to design flowcharts and pseudocode. Flowcharts are used to visually represent program processes using shapes and arrows, while pseudocode is used to describe algorithm logic in plain language. Both tools aid with planning approaches before actual implementation. Next, students developed and tested their code on their robot. Labs were graded based on the robot autonomously driving accurate distances, minimizing deviations from goal locations, and completing the task within the allotted time. Students were also allowed lab retakes, with the goal of alleviating stress caused by unpredictable robot hardware. To qualify, students submitted a lab retake questionnaire outlining their hypothesis for their robot's failure and their corresponding fixes.

3.2 Participants & Procedure

After receiving approval from our university's Institutional Review Board, we recruited a total of 20 (12 male and 8 female) out of the 40 computer science students enrolled in our university's *Introduction to Robotics* course. These participants were part of 13 distinct lab groups, with four of the groups having had a single partner sign up for our study. Of the participants, one (5%) was a sophomore, four (20%) were juniors, fourteen (70%) were seniors, and one (5%) was a fifth year. All of the participants had at least two years of programming experience with eight (40%) participants reporting they were most comfortable with Python and eight (40%) with Java. Two (10%) students stated they participated in a robotics club, four (20%) reported some experience with robotics projects, and fourteen (70%) had no experience with robots. All students that signed up for our study participated via online surveys, with eight (40%) of the participants additionally volunteering for five semi-structured verbal interviews. These eight participants represented five (38.46%) of the thirteen lab groups. We believe our sample of participants provides an accurate representation of the typical cohort one might expect in an introductory robotics course.

3.3 Data Collection

At the beginning of the term, a survey was administered to gather students' demographics and their expectations for the course. For



(a) Lab 1 **(b) Lab 2** **(c) Lab 3** **(d) Lab 4**

Figure 2: We investigate how to support undergraduate students who are learning about and implementing robotics concepts through a series of four labs. Each lab has the following objectives: (a) Lab 1: The robot autonomously drives from the start position to Goal 1, then to Goal 2, and finally reverses after a wall bump to reach Goal 3, emitting a beep at each stop, (b) Lab 2: The robot autonomously moves forward until it is within $0.3m$ of the wall, turns right, and follows the wall for $2m$ while maintaining a maximum distance of $0.3m$ from the wall, (c) Lab 3: The robot autonomously moves forward until it is within $0.3m$ of the wall, then turns right and circumnavigates the obstacle while staying within $0.3m$ of the wall, with the goal of returning back to the starting point, (d) Lab 4: The robot autonomously maneuvers around two to six obstacles to reach the goal. These obstacle arrangements are examples, and students did not know the exact setup until their evaluation.

each lab, we collected pre-lab assignments, students' lab code, recordings of lab evaluations, lab retake reflections, and post-lab feedback. After each lab, we conducted semi-structured interviews to gain detailed insights into their lab experiences. Additionally, we collected discussion information from Piazza and took field notes of students' activities during labs. At the end of the semester, students provided feedback about their overall experience in the course. Finally, we conducted a closing semi-structured interview with the students who volunteered. As a result, the data collection process yielded over 600 excerpts of student survey feedback and discussion posts, 3.5 hours of verbal interview transcriptions, and 3 hours of robot recordings. The questionnaires and interview questions can be found at this link: https://github.com/hri-ironlab/comped-2023-supplemental-materials.

3.4 Data Analysis

The verbal interviews were transcribed using an intelligent verbatim approach, aligning spoken data with written conventions while preserving the intended meaning [22]. Our goal was to prioritize the communicated content over how it was said and to enhance readability. For analysis, the survey responses, online forum comments, and interview transcriptions were compiled into one file for each lab, then further sorted into lab groups. To analyze the participants' experiences, we used reflexive TA, a research method for exploring and interpreting qualitative data [5]. Following this method, we first read and reread the survey and interview responses. Next, two researchers independently coded the data from the first lab in ATLAS.ti, a qualitative analysis software tool. We then compared the coding sets and noted any differences. Then, the first author coded the rest of the data. Following coding, themes were identified from groups of codes that presented evidence regarding our research question [4]. We then reviewed these themes by analyzing the quotes and refined the themes to accurately represent participants' experiences. The final revision of our analysis resulted in three themes that are described in Table 1 and discussed in Section §4. These include (1) Positive learning opportunities, (2) Dealing with uncertainty, and (3) Successful results with simple solutions.

4 FINDINGS

Overall, the students were able to successfully complete each lab. Multiple teams required lab retakes, averaging 2.46 (SD = .63) times, significantly influencing their scores. Specifically, six (46.15%) teams used retakes in Lab 1, seven (53.85%) in Lab 2, nine (69.23%) in Lab 3, and ten (76.92%) in Lab 4. The average grades before the retakes were 87.23% (SD = 11.23) for Lab 1, 90% (SD = 8.90) for Lab 2, 74.77% (SD = 28.08) for Lab 3, and 66.85% (SD = 38.94) for Lab 4. After retakes, the averages improved to 95.08% (SD = 4.65) for Lab 1, 94.85% (SD = 8.21) for Lab 2, 92.85% (SD = 7.22) for Lab 3, and 87.31% (SD = 8.95) for Lab 4. It is worth noting that Lab 4 presented challenges for multiple teams due to an edge case caused by a concave wall. As a result, the instructor reoriented the course and gave students another chance, producing better scores for several teams. The instructor's flexibility on this matter was appreciated by the students.

P18[Lab 4]: *"I like that [the instructor] was merciful when moving the obstacles that were really bad. Like a sharp, sharp turn, [the instructor] ended up moving them to be less sharp for the second take on the same day. I felt like that was great."*

Over all four labs, eighteen (90%) students recognized they were able to successfully implement class concepts such as robot motion planning (e.g., the Bug algorithms [20]) and control.

Table 1: Thematic structure

Theme	Sub-Themes	Summary	Guidelines
Positive learning opportunities	Pre-lab assignments Learning from online resources Collaborating with peers	Students benefited from planning their lab solutions before beginning to code. Also critical to their success was informative online resources and a dedicated classroom to work with others.	Provide positive learning opportunities such as pre-lab activities to facilitate early problem solving. Additionally, a dedicated robot testing space can promote learning through collaboration.
Dealing with uncertainty	Hardware versus software bugs Inadequate debugging techniques Difficulties testing the robot	Debugging was challenging for students because they often lacked the knowledge to properly debug hardware and software problems. Most groups employed a trial-and-error approach, while some groups used more strategic techniques.	Incorporate effective robot debugging strategies and tools into ER curriculum, such as visualization libraries for monitoring and debugging robot data streams, or robot simulation tools for testing.
Successful results with simple solutions	Applying simple solutions Lab retake pros and cons Success attributed to luck	Students programmed each lab with the simplest solution, incorporating the course-taught material only when required. Some students hard-coded parts of their solutions and some attributed their success to luck.	Guide students towards implementing class concepts earlier and encourage them to reflect on the benefits of these approaches. Instructors may also provide correct lab solutions to prevent errors from propagating, or grade based on a student's ability to implement class concepts.

P7[Final Feedback]:*"I would say bug two that we used for the M line. I think that that's the one that we implemented the best."*

Still, the course was not without its challenges, and the students' responses to these challenges were influenced by various factors such as the course structure, learning materials, the classroom environment, and coding challenges. In the subsequent sections, we discuss these experiences while highlighting examples of effective classroom support as well as areas that can be improved.

4.1 Positive learning opportunities

Overall, the feedback suggests that students found the robotics labs to be challenging but rewarding, and their experiences were positively influenced by careful planning, online resources, and collaboration with their peers. Nine (45%) students reported their flowcharts and pseudocode prepared them to tackle the lab problems. One group even devised a unique hardware design during their pre-lab, that helped them succeed in the labs. This solution entailed rotating the ultrasonic sensor by attaching it to a motor, to use it as both a bumper sensor and a sensor for wall following.

P10[Lab 2]:*"We came up with the idea of like rotating the sensor in our pseudocode .... I think it was a eureka moment."*

However, one group found the first pre-lab unhelpful because the task was considered to be too straightforward to require a diagram. Instead, this group emphasized online resources, considering them crucial for their success. Likewise, fifteen (71.42%) students stated that online resources such as application protocol interface (API) documentation was essential for them to program the robot.

P18[Final Feedback]:*"Looking at the py, I mean EV3 functions, and understanding what each thing was doing or how you're supposed to code the motors, like the gyroscope, the various run functions that the motors have. I thought that was kind of the thing that ended up helping us a lot."*

This feedback supports prior work by Williams et al., 2007 [33] that highlights the need for accessible and high quality robot documentation to facilitate student learning. When online resources did not meet all of the students needs, fourteen (70%) students mentioned seeking assistance from their peers For example, one group had trouble getting back to the starting point in Lab 3 and used part of a solution they observed from other students.

P10[Lab 3]:*"We spent a decent bit of time in that room next to the robotics lab where we were doing all of our testing in our labs. And we saw some of those teams that went back until they saw the start of the final little stretch. And that was something that we encoded in as well. We thought that was a great idea .... So looking at other people definitely helped. And there's like a collaborative effort, of course."*

The dedicated classroom facilitated a collaborative atmosphere that enhanced the students' learning experience. This feedback supports the guidelines outlined by Lindh and Holgersson, 2007 [17], which emphasizes the importance of providing a dedicated space for robotics projects. We observed that this arrangement promoted positive discussions among students. In summary, students benefited from planning their lab solutions beforehand, informative online resources, and a dedicated classroom for collaboration.

4.2 Dealing with uncertainty

One of the learning objectives was to educate students about the uncertainty that occurs when robots interact with the physical world. Roboticists often encounter uncertainty in sensing (e.g., noisy measurements) and in the robot's motion (e.g., unpredictable slip between wheels and the floor). This makes robot programs challenging to debug because running a robot in the same environment multiple times can produce different real-world motions [13]. Understandably, students encountered challenges while debugging their robot's behavior, with six (30%) students finding it particularly difficult to distinguish between hardware limitations and software bugs.

P4[Lab 2]: *"How the code interfaces with the hardware is very difficult, the sensor can be unreliable for a variety of reasons so it's often difficult to tell if the problem is in our code or the hardware."*

Since this was a new experience for students, eleven (84.62%) groups applied guess and check debugging methods to examine how parameters in their code affected the robot's performance.

P9[Lab 1]: *"A lot of it was guess and check . . . trial and error. We just kept tweaking our numbers until we got distance perfectly right."*

This trial-and-error debugging approach is often observed in introductory courses, but can hinder students when compared to more strategic methods [10]. Five (25%) students recognized they needed a more focused approach to identify the source of their bugs. This was typically described as a bug isolation method.

P4[Final Feedback]: *"Isolating a line or chunk of code and running it alone to understand it, was most effective."*

By Lab 3, one group devised a visual debugging technique, similar to what is commonly utilized in the robotics community with tools like ROS Visualization [15]. They used a Python library called Matplotlib to code a graphical debugging tool that helped them visualize their robot data. As a result, this group was able to successfully pinpoint errors by visually debugging the relationship between the ultrasonic sensor values and the calculated robot position.

P7[Lab 3]: *"We implemented a relatively accurate x y coordinate position where it's, as the robot goes to 70 in the x and 10 in the y, it actually knows that it was 70 in the x and 10 in the y because of the graph that we had for lab three. It was like the shape of the object in the scans was pretty messed up . . . it was fairly off."*

Later, when students wanted to test and debug their code on the physical robot, they had difficulties constructing test courses that would provoke edge cases. Even for seasoned roboticists, this is a challenging task that requires experience and critical thinking to develop scenarios that may elicit unwanted robot behavior. One group described their dissatisfaction with the results of this process.

P18[Lab 4]: *"The day before the lab we built this obstacle to test it, then we kind of knew that we can test it off obstacles that other people built. So we kind of knew that our robot would work almost perfectly on like, 95% of the tracks . . . so there's only a very small subset of tracks where, you know, it would have the wheels get caught. But unfortunately, that happened on the actual lab."*

Although this group struggled to develop enough representative test cases, it is a positive sign that the dedicated classroom fostered positive idea sharing among students. Another difficulty students faced when testing their physical robot was coordinating with their lab partner. Three (15%) students mentioned difficulties finding time to test their code when their lab partner possessed the robot. This was a bi-product of the limited number of robot kits and the team-based projects. Therefore, it may be relevant to find ways to help remedy this issue in future courses.

P5[Final Feedback]: *"The thing about having partners with a robot is that even if one person writes up some code, there's only one person who can actually debug it. . . . This means that most of the work hinges on the person who has the robot, unless you manage to find time to meet up (which was like pulling teeth)"*

In summary, the debugging process was challenging for students because of their limited knowledge of robot debugging techniques. Further difficulties arose when students could not access the shared robot for testing. Therefore, future work should explore how to integrate robot debugging into ER curricula and devise strategies to mitigate issues caused by limited physical resources.

4.3 Successful results with simple solutions

To program an autonomous robot, the course covered concepts for localization, control, and motion planning. While implementing these concepts, students frequently opted for the simplest solutions. For instance, students were introduced to the concept of wheel odometry for Lab 1, which involves using wheel velocities and positions to calculate the robot's position and orientation. However, none of the teams implemented wheel odometry for Lab 1. Instead, students opted for a simpler approach, which was to manually adjust the time it would take for the robot to cover a particular distance. For Lab 2, the course taught two methods for controlling the robot to follow a wall: a simpler "bang-bang" controller and a more challenging but reliable proportional controller. Unsurprisingly, eleven (84.62%) groups opted for the "bang-bang" approach, with one group citing ease of implementation as their rationale.

P7[Lab 2]: *"That's [the proportional controller] the last effort that we're gonna take in the beginning. But then, the code that we did followed the wall almost perfectly. So it's like, yeah, I don't see a reason to make it proportional."*

For Lab 3, seven teams (53.85%) implemented both wheel odometry and a "bang-bang" controller, and by Lab 4, eleven teams (84.62%) also successfully coded the taught motion planning algorithm. From these observations, it is evident that students tackled problems with the simplest approach in mind, a common strategy deployed in robotics. As the complexity of the lab increased, students eventually realized that more complex solutions were necessary.

It is worth noting that the option of retaking labs allowed students to experiment and adjust parameters to improve the robot's performance. Upon reviewing video recordings and submitted code, fourteen (43.75%) out of thirty-two retakes used some form of hardcoding for part of their code to succeed in the lab. Eighteen (56.25%) retakes were utilized to overcome seemingly uncontrollable robot problems such as motor or gyroscope malfunctions. Thus the students used the same code for the rerun, ultimately achieving a better score. Thirteen (65%) students expressed appreciation for the ability to retake the lab, citing the setbacks caused by robot malfunctions.

P5[Final Feedback]: *"The sensors themselves were finicky at best. They worked when we needed them to, but that wasn't the case for everyone. This is a major reason why the retakes are so helpful: sometimes a robot just needs to be turned off and back on again."*

While students were able to achieve satisfactory lab grades, due to the troubles caused by the robot sensors and hardware, students still attributed their success to luck. For example, one group obtained a successful lab evaluation by restarting their malfunctioning robot. But, when development began for the subsequent lab, the errors persisted thus frustrating the students.

P3[Lab 2]: *"We just rebooted [the robot] and then it started working again . . . . It didn't feel like our hard work paid off. Because it was if it works, it works . . . and now we're kind of suffering from that because now our robot wont work so we still don't know why."*

According to Weiner, 2014 [32], when students attribute their academic success to luck, this triggers uncertainty and apprehension about future success. This highlights the need for standardized and reliable hardware platforms for undergraduate robotics courses. While numerous platforms exist [24, 25], none have achieved widespread adoption. Still, students could have improved their understanding of robotics and the robustness of their robot behavior to uncertainty by implementing class concepts earlier. As mentioned previously, only two (15.38%) groups opted to implement a proportional controller for more reliable robot control. Consequently, it may be beneficial to incentivize students to implement more complex but robust class concepts earlier, thus setting them up for success in future labs. In summary, students succeeded in completing the labs, often utilizing the simplest solutions.

5 DISCUSSION

In this section, we address each theme with guidelines for improving future ER undergraduate courses. These include promoting positive learning opportunities, integrating common robot debugging tools to address student challenges, and guiding students to utilize class concepts to produce more positive learning outcomes.

Promote positive learning opportunities: During the course, we observed positive learning opportunities that should be provided in future ER curricula. For instance, assigning flowchart and pseudocode activities as pre-lab assignments helped facilitate students' planning of solutions and fostered early problem-solving, and thus should be adopted as a common practice. Second, as recommended by prior work in K–12 ER, we suggest providing dedicated robot testing spaces to students [3, 17]. This setup naturally facilitated idea-sharing and group problem-solving, which multiple students appreciated. Third, as outlined by Williams et al., 2007 [33], accurate robot documentation should be readily available, as this was critical for students' success.

Incorporate robotics debugging tools: Future ER curriculum should also incorporate robotics debugging tools as advised by prior work [16]. This is because students employed a tedious and often ineffective trial-and-error approach, with only one group developing a graphical tool to monitor their sensor and positional data. Robot data visualization tools are commonly utilized by roboticists to pinpoint errors in robot code, such as inaccuracies in a computed robot position [19]. Therefore, future ER curriculum should either provide robot data visualization tools or teach students how to develop their own using familiar graphical libraries to better understand and debug their robot's behavior. Another common debugging tool in robotics is simulators. These offer programmers a fast and efficient way to test and validate their algorithms on a variety of simulated test cases before being deployed in the real world, and could be taught along robotics coursework [11, 31]. Simulators enable students without access to a physical robot to test their code. However, the absence of a standardized visualization and simulation toolkit for introductory robotics courses emphasizes the need for the ER community to develop debugging tools along with relevant curricula for popular ER robots.

Guide students toward using class concepts: When writing lab code, students opted for the simplest approach and only implemented more advanced solutions when absolutely necessary. While this approach is valid, instructors could guide students toward stronger solutions earlier, setting them up for success in future labs. For example, pre-lab assignments could instruct students to outline their solutions using class concepts and reflect on the robot uncertainties these approaches might help mitigate. Another potential strategy is to provide students with the correct solution of the lab at the end of each evaluation, in order to prevent misunderstandings and errors from persisting throughout the semester. However, this would require the student's robot provides similar functionality to the robot built for the solution code. A third approach could involve grading student labs based on both performance metrics and their ability to apply learned class concepts to their programs. This approach could both motivate students to develop robust solutions and encourage them to engage with the course material earlier. By offering clear guidance towards solutions that promote resilient robot behavior, students stand to gain a deeper understanding of robotics concepts that address uncertainty in robots and alleviate associated stress.

Limitations and Future Work: In addition to these guidelines, future work should address the limitations of our study to further understand how robotics curricula may be improved. For example, although we believe the collected student experiences were representative of an introductory robotics course, only half of the class volunteered for our study. Therefore, future work should strive to include the entire class so that all opinions are taken into account. Additionally, the richest feedback came from students who participated in the verbal interviews since they gave more detailed responses compared to the online surveys. Future work should incorporate more students into the verbal interview process. Lastly, future research might explore our guidelines with linked assessments to explore the efficacy of such course adjustments.

6 CONCLUSION

In our study of an undergraduate introductory robotics course, students successfully learned about and implemented robotics concepts onto an autonomous mobile robot. This achievement is noteworthy given the complexity of robotics, highlighting the success of the current curriculum. Additionally, we identified guidelines for designing future robotics courses informed by our reflexive thematic analysis of student feedback. These guidelines include: promote positive learning opportunities, incorporate robotics debugging tools, and guide students toward using class concepts in programming labs. We hope this work serves as a guide for the development of future undergraduate introductory robotics courses and inspires further research into robotics curricula.

REFERENCES

[1] Feri Ardiana Aristawati, Cucuk Budiyanto, and Rosihan Ari Yuana. 2018. Adopting educational robotics to enhance undergraduate students' self-efficacy levels of computational thinking. *Journal of Turkish Science Education* 15, Special (2018), 42–50.

[2] Alireza Badeleh. 2021. The effects of robotics training on students' creativity and learning in physics. *Education and Information Technologies* 26, 2 (2021), 1353–1365.

[3] Fabiane Barreto Vavassori Benitti. 2012. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education* 58, 3 (2012), 978–988.

[4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[5] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[6] Erin Cejka, Chris Rogers, and Merredith Portsmore. 2006. Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education* 22, 4 (2006), 711.

[7] Vidushi Chaudhary, Vishnu Agrawal, Pragya Sureka, and Ashish Sureka. 2016. An experience report on teaching programming and computational thinking to elementary level children using lego robotics education kit. In *2016 IEEE Eighth International Conference on Technology for Education (T4E)*. IEEE, 38–41.

[8] Jennifer Cross, Christopher Bartley, Emily Hamner, and Illah Nourbakhsh. 2013. A visual robot-programming environment for multidisciplinary education. In *2013 IEEE International Conference on Robotics and Automation*. IEEE, 445–452.

[9] Ana Cruz-Martín, Juan-Antonio Fernández-Madrigal, Cipriano Galindo, Javier González-Jiménez, Corin Stockmans-Daou, and José-Luis Blanco-Claraco. 2012. A LEGO Mindstorms NXT approach for teaching at data acquisition, control systems engineering and real-time systems undergraduate courses. *Computers & Education* 59, 3 (2012), 974–988.

[10] Stephen H Edwards. 2004. Using software testing to move students from trial-and-error to reflection-in-action. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education*. 26–30.

[11] NM Fonseca Ferreira, André Araujo, MS Couceiro, and David Portugal. 2018. Intensive summer course in robotics–Robotcraft. *Applied Computing and Informatics* 16, 1/2 (2018), 155–179.

[12] Mohd Faisal Ibrahim, Aqilah Baseri Huddin, Fazida Hanim Hashim, Mardina Abdullah, Ashrani Aizzuddin Abd Rahni, Seri Mastura Mustaza, Aini Hussain, and Mohd Hairi Mohd Zaman. 2020. Strengthening Programming Skills among Engineering Students through Experiential Learning Based Robotics Project. *International Journal of Evaluation and Research in Education* 9, 4 (2020), 939–946.

[13] Bryce Ikeda and Daniel Szafir. 2022. Advancing the Design of Visual Debugging Tools for Roboticists. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 195–204. https://doi.org/10.1109/HRI53351.2022.9889392

[14] Seul Jung. 2012. Experiences in developing an experimental robotics course program for undergraduate education. *IEEE Transactions on Education* 56, 1 (2012), 129–136.

[15] Hyeong Ryeol Kam, Sung-Ho Lee, Taejung Park, and Chang-Hun Kim. 2015. Rviz: a toolkit for real domain data visualization. *Telecommunication Systems* 60 (2015), 337–345.

[16] Jean-franois Lalonde, Christopher P Bartley, and Illah Nourbakhsh. 2006. Mobile robot programming in education. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 345–350.

[17] Jörgen Lindh and Thomas Holgersson. 2007. Does lego training stimulate pupils' ability to solve logical problems? *Computers & education* 49, 4 (2007), 1097–1111.

[18] Hsin-Yu Liu, Wen-June Wang, and Rong-Jyue Wang. 2010. A course in simulation and demonstration of humanoid robot motion. *IEEE Transactions on Education* 54, 2 (2010), 255–262.

[19] Yanan Liu, Rui Fan, Bin Yu, M. Junaid Bocus, Ming Liu, Hepeng Ni, Jiahe Fan, and Shixin Mao. 2018. Mobile Robot Localisation and Navigation Using LEGO NXT and Ultrasonic Sensor. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 1088–1093. https://doi.org/10.1109/ROBIO.2018.8665350

[20] Vladimir J Lumelsky and Alexander A Stepanov. 1987. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica* 2, 1-4 (1987), 403–430.

[21] Bruce A Maxwell and Lisa A Meeden. 2000. Integrating robotics research with undergraduate education. *IEEE Intelligent systems and their applications* 15, 6 (2000), 22–27.

[22] Caitlin McMullin. 2021. Transcription and qualitative methods: Implications for third sector research. *VOLUNTAS: International journal of voluntary and nonprofit organizations* (2021), 1–14.

[23] Ruben Mitnik, Miguel Nussbaum, and Alvaro Soto. 2008. An autonomous educational mobile robot mediator. *Autonomous Robots* 25 (2008), 367–382.

[24] Liam Paull, Jacopo Tani, Heejin Ahn, Javier Alonso-Mora, Luca Carlone, Michal Cap, Yu Fan Chen, Changhyun Choi, Jeff Dusek, Yajun Fang, et al. 2017. Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1497–1504.

[25] Fanny Riedo, Morgane Chevalier, Stéphane Magnenat, and Francesco Mondada. 2013. Thymio II, a robot that grows wiser with children. In *2013 IEEE Workshop on Advanced Robotics and its Social Impacts*. IEEE, 187–193.

[26] Laurel D Riek. 2012. Embodied computation: An active-learning approach to mobile robotics education. *IEEE Transactions on Education* 56, 1 (2012), 67–72.

[27] Michael Rubenstein, Bo Cimino, Radhika Nagpal, and Justin Werfel. 2015. AERobot: An affordable one-robot-per-student system for early robotics education. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6107–6113.

[28] Javier Ruiz-del Solar and Roberto Avilés. 2004. Robotics courses for children as a motivation tool: the Chilean experience. *IEEE Transactions on Education* 47, 4 (2004), 474–480.

[29] David Scaradozzi, Laura Sorbi, Anna Pedale, Mariantonietta Valzano, and Cinzia Vergine. 2015. Teaching robotics at the primary school: an innovative approach. *Procedia-Social and Behavioral Sciences* 174 (2015), 3838–3846.

[30] Andrés-David Suárez-Gómez and Wilson Javier Pérez-Holguín. 2020. Physical visualization of math concepts using LEGO Mindstorms. *JOTSE: Journal of Technology and Science Education* 10, 1 (2020), 72–86.

[31] Christopher Steven Timperley, Afsoon Afzal, Deborah S Katz, Jam Marcos Hernandez, and Claire Le Goues. 2018. Crashing simulated planes is cheap: Can simulation detect robotics bugs early?. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 331–342.

[32] Bernard Weiner. 2014. The attribution approach to emotion and motivation: History, hypotheses, home runs, headaches/heartaches. *Emotion Review* 6, 4 (2014), 353–361.

[33] Douglas C Williams, Yuxin Ma, Louise Prejean, Mary Jane Ford, and Guolin Lai. 2007. Acquisition of physics content knowledge and scientific inquiry skills in a robotics summer camp. *Journal of research on Technology in Education* 40, 2 (2007), 201–216.

[34] Yi Wu, Charlotte de Vries, and Qi Dunsworth. 2018. Using LEGO Kits to Teach Higher Level Problem Solving Skills in System Dynamics: A Case Study. *Advances in Engineering Education* 6, 3 (2018), n3.

[35] Liying Xia and Baichang Zhong. 2018. A systematic review on teaching and learning robotics content knowledge in K-12. *Computers & Education* 127 (2018), 267–282.

[36] Baichang Zhong and Liying Xia. 2020. A systematic review on exploring the potential of educational robotics in mathematics education. *International Journal of Science and Mathematics Education* 18 (2020), 79–101.