12-2018

Effectiveness of physical robot versus robot simulator in teaching introductory programming

Oka KURNIAWAN
Singapore University of Technology and Design

Norman Tiong Seng LEE
Singapore University of Technology and Design

Subhajit DATTA
Singapore Management University, subhajitd@smu.edu.sg

Nachamma SOCKALINGAM
Singapore University of Technology and Design

Pey Lin LEONG
Singapore University of Technology and Design

Citation

KURNIAWAN, Oka; LEE, Norman Tiong Seng; DATTA, Subhajit; SOCKALINGAM, Nachamma; and LEONG, Pey Lin. Effectiveness of physical robot versus robot simulator in teaching introductory programming. (2018). *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE): Wollongong, Australia, 4-7 December: Proceedings*. 486-493.

Available at: https://ink.library.smu.edu.sg/sis_research/6009

Effectiveness of Physical Robot Versus Robot Simulator in Teaching Introductory Programming

Oka Kurniawan
Information Systems Technology and Design Pillar
Singapore University of Technology and Design
Singapore
kurniawano@ieee.org

Norman Tiong Seng Lee
Information Systems Technology and Design Pillar
Singapore University of Technology and Design
Singapore
norman_lee@sutd.edu.sg

Subhajit Datta
Information Systems Technology and Design Pillar
Singapore University of Technology and Design
Singapore
subhajit_datta@sutd.edu.sg

Nachamma Sockalingam
Learning Sciences Lab
Singapore University of Technology and Design
Singapore
nachamma@sutd.edu.sg

Pey Kin Leong
Engineering Product Development Pillar
Singapore University of Technology and Design
Singapore
peykinleong@sutd.edu.sg

***Abstract*—This study reports the use of a physical robot and robot simulator in an introductory programming course in a university and measures students' programming background conceptual learning gain and learning experience. One group used physical robots in their lessons to complete programming assignments, while the other group used robot simulators. We are interested in finding out if there is any difference in the learning gain and experiences between those that use physical robots as compared to robot simulators. Our results suggest that there is no significant difference in terms of students' learning between the two approaches. However, the control group that uses the physical robot shows a more positive response in their attitudes towards computing. We discuss the implications of our findings in relation to engaging students and challenges in using physical robots from the learner perspectives and ways to alleviate this. Finally, by considering the insights from students' comments, we also suggest an alternative that may give both benefits of using both physical robots and robot simulators.**

***Keywords*—programming, robotics, simulator, computational thinking**

I. Introduction

Computational Thinking (CT) [1] has been identified as one of the key skill sets needed for a 21st century workforce. With this realization comes the demand for teaching introductory programming to a wider group of students who may not necessarily have an engineering or computing background. To teach a course to students with a very wide spectrum of abilities and prior knowledge is challenging. This requires instructors to innovate their pedagogy to ensure that students can keep pace, are engaged, can learn and apply what they have learnt to real-life problems. Using robots in introductory programming course has been recognized as one way to achieve these objectives.

Robotics activities have been recognized by Papert to have potential in education [2]. One of the reasons could be due to the multiple pathways of learning that robotics provides. Resnick noted that different students are attracted to different types of robotics activities and so provide multiple entry points for students with diverse interests and learning styles [3]. Early studies of robotics were limited only to physics and mathematics and conducted primarily in pre-university settings [4]. Recent studies have described how robots were employed in introductory computing courses in university settings [5]-[8].

Unfortunately, research results on the effectiveness of robots as a teaching tool have been mixed [4]. For example, while Barker [9] and Nugent et. al. [10] reported that robots are effective as a teaching tool for programming, Sullivan reported otherwise [11]. While some studies report that robots are effective in increasing thinking or problem-solving skills [10][11], there are others that report to the contrary [12][13].

Fagin and Merkle conducted an extensive study with about 800 university freshmen on the use of robotics to teach programming [14]. Compared to students who did not use robots, they found that students who used robots scored lower on tests. They also observed that the use of robots did not have any measurable effect on students' choice of discipline. They attributed this to a lack of a robot simulator, which is a software tool that enables students to test their code before deploying it on an actual robot. Also, physical robots have several hardware-related challenges that are not present in software. Having to overcome these challenges reduces the time available for students to actually learn programming. At the same time, they acknowledged that using robots had its own advantage in terms of student motivation. The student

Published in 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE): Wollongong, Australia, 4-7 December: Proceedings, pp. 486-493. DOI: 10.1109/TALE.2018.8615190

feedback tended to be positive and found the learning to be interesting, fun, challenging, and relevant.

Major et. al. surveyed the literature on the use of robots in teaching novice programmers [15]. Most of the papers that they surveyed reported that robots are effective as a teaching tool and can help novice programmers in their studies. Most of those papers used physical robots and only a few used simulators. Out of 23 papers that use physical robots, 16 claimed that robots are effective, four indicated mixed results, and one showed a negative result. On the other hand, out of seven papers that use simulators, six works claim it to be effective and one is categorized as unclassifiable. Since no negative results is reported for those papers using simulators, they proposed an intriguing question of whether simulators can be more effective than physical robots in teaching novice programmers.

We pick up this question and our study aims to compare the effectiveness of using physical robots with using robot simulators in an introductory programming course. Our main question is to investigate whether there is any significant difference between the use of physical robot as compared to simulators on student learning of programming skills and concepts. We are also interested if such differences can also be observed in students' interests, motivation, and confidence in computing.

We first review some related works on this area in section II. Section III then describes the context of our study and our methodology. The results of our study are presented in Section IV. These include the surveys, programming tests, as well as comments from participants. Finally, we discuss our results in Section V and compare it with some other works which may help to give insight to these results.

II. Related Works

There have been plenty of works in studying the use of robots for teaching as shown in the survey in reference [8]. Many of studies these used robotics to teach physics and mathematics. Recently, there are more works on the use of robotics to teach programming as shown in the papers surveyed in [15]. However, most of these works were conducted in pre-university settings. Moreover, they used either physical robots or simulators, but not both. An extensive study in this area using physical robots was done by Fagin and Merkle in 2003 [14]. They reported that the use of robot resulted in a negative effect on students' performance. This was contradicted by a recent study that reported positive learning gains [16].

In university settings, some works have done in studying the effective of robots for teaching introductory programming course. Similar to pre-university, most of these works only study either physical robots as in [6][8] or only robot simulators as in [5]. None of those seem to address which of the two approaches are more effective in teaching introductory programming course.

Two papers that were closely related to our works were done by Brauner, et al. in [17] and Wu, et. al. in [18]. Brauner, et. al. compares the effect of using tangible robots to using visual representations for 12- to 13-year-old students. The impact was measured on learning outcomes, self-efficacy, class feedback, and attitudes towards STEM (science, technology, engineering, and mathematics). They saw a small but no significant difference in the understanding of a computer program. Their work, however, focused on pre-university setting. The programming environment used Scratch, which is more visual programming rather than text-based programming commonly found in university setting.

The study by Wu, et. al. in [18] is similar to our study described here. In that work, two classes of high school students used Lego Minsdstorms Robot and the other two classes used Lego Mindstorms Simulator. Students used the Java language in this programming environment. They found there were no significant difference on students' performance. At the same time, the groups using the physical robot demonstrated a more positive attitude toward the activities and they could better imagine the program behavior. This work also focused on pre-university setting and only on attitude toward the learning activities. We are more interested on how physical robot or robot simulator affects their attitudes towards computing in general in the future.

III. Methods

A. Context of Study

Our study is conducted with the student participants of a five-day workshop titled "Introduction to Computational Thinking". This workshop is aimed at helping students with little or no programming background to cope with a subsequent programming course that is compulsory for all first-year students in our institution.

The participants were recruited from the first-year undergraduate students in our university in 2018. Although 100 students enrolled in the workshop, 48 students completed the surveys and the pre- and post-tests. This was because participation in the study was not compulsory for the workshop participants. About 40% of the participants were female which is about the same proportion of the University's female population. The typical age of the male and female participants are 22 years and 20 years respectively.

Each day of the workshop comprised of a three-hour session involving online learning, in-class problem solving, and an end-of-day challenge. The end-of-day challenge aimed to help students to see immediately how their learning can be applied.

In the first three days of the workshop, participants were taught basic algorithmic structures, i.e. sequential, branch, and iteration. In the remaining two days, they were then introduced to programming challenges that involves a combination of these structures.

Fig. 1. Thymio robot [19] is used together with Raspberry Pi for students to program using Python.

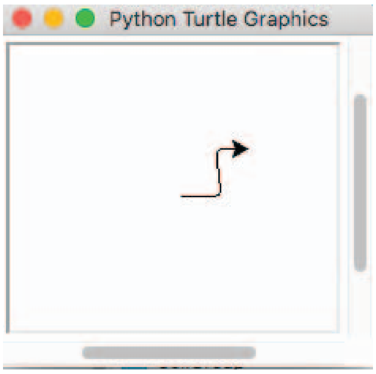

Fig. 2. Simulation view. The API makes use of Python's built-in turtle module.

Participants were randomly assigned to two groups. One group used a physical robot during the end-of-day challenges (the "physical robot group"), while the other used a robot simulator ("simulator group"). For the robot, we used the Thymio as the physical robot controlled from a Raspberry Pi (Fig.1) [19]. We wrote a Python library that allowed students to have access to the wheels and its sensors [20]. For the robot simulator (Fig. 2), the same library makes use of Python's turtle module instead. The robot is represented as a triangle and a trail can be set to see the motion of the robot. Both the physical robot and the simulator uses the same programming interface. The only difference is the Python class name that is used to initialize the object (Fig. 3). Both groups were given identical end-of-day challenges. The challenges comprise of tasks related to sequential, branching, and iterative structure of computer codes. The final challenge incorporates all the structures taught in the previous days.

`from pythymiodw import ThymioReal`
 	
`robot = ThymioReal()`
`robot.wheels(100,100)`
`robot.sleep(2)`
`robot.wheels(100,-100)`
`robot.sleep(2)`
`robot.quit()`
	`from pythymiodw import ThymioSim`
	
	`robot = ThymioSim()`
	`robot.wheels(100,100)`
	`robot.sleep(2)`
	`robot.wheels(100,-100)`
	`robot.sleep(2)`
	`robot.quit()`

Figure 3. Python code comparison between students using the physical robot and the robot simulator. The only difference is the third line, which is the class name to instantiate the object.

B. Measurements

The workshop participants were invited to complete a pre-test and a post-test to measure their conceptual learning. Two other survey questionnaires were administered to measure their programming background and learning experience.

1) Conceptual Learning

The pre-test and post-test used a programming quiz that we wrote. The test aimed to assess students' programming knowledge and skills, hence the questions focused on students' ability to read and write programs using both pseudocode and Python code. The same programming quiz was used for both the pre-test and the post-test. Before deploying the quiz on the workshop participants, we tested the questions with two student helpers to ensure that the questions were clear and without errors. The pre-test was conducted at the start of the first lesson and the post-test at the end of the last lesson after the final challenge.

2) Students' programming background

Together with the pre-test, a short questionnaire on students' background and their familiarity with programming was administered at the start of the first lesson.

3) Students' learning experience

At the end of the workshop, which is after the completion of the final challenge, the participants completed a questionnaire containing two parts. The first part of the questionnaire surveyed students on their perceptions of the workshop and the use of robots or simulation in learning computing.

The second part contained questions selected from an attitude survey intended to measure student perceptions on computer science developed and validated by Hoegh et. al. [21]. This survey was chosen as the authors provided evidence to show that it is valid and reliable. We used the questions intended to measure the constructs of students' confidence and interest in computing.

Our perception questionnaire consists of 29 questions containing a mixture of Likert scale, multiple choice, and open-ended questions. The questions are divided into several categories that probe students' interests, engagement, and confidence in computing. To check for consistency, the questions are divided into positive statements and negative statements.

C. Analysis

Survey responses were collected using an online platform and downloaded as an Excel sheet. The analysis was done using Tableau to calculate and visualize the average Likert score of the survey responses.

The pre-test and post-test were done on paper and students did not have access to the computer or any other resources during the tests. Average scores were computed from the pre-test and post-test answers. T-test comparisons were conducted to determine significant differences.

Responses on open-ended questions were text-analyzed by word-frequency counting and categorized by theme. Categories that had the highest frequency of keywords were deemed to be of importance and ranked by frequency.

III. Results

Participants' programming background before the workshop are depicted in Fig. 4. The majority of the students reported that they have not written any programming code before (46%) or small programs consisting of between 1 to 10 lines of code (34%). Hence, this shows that participants joining the workshop had little programming skills.

Out of the 100 students who registered for the workshop, only 48 students completed both the pre-test and the post-test. Of these, 25 students were from the physical robot group and 23 were from the simulator group. The average of pre- and post-test results for both classes are shown in Table 1.

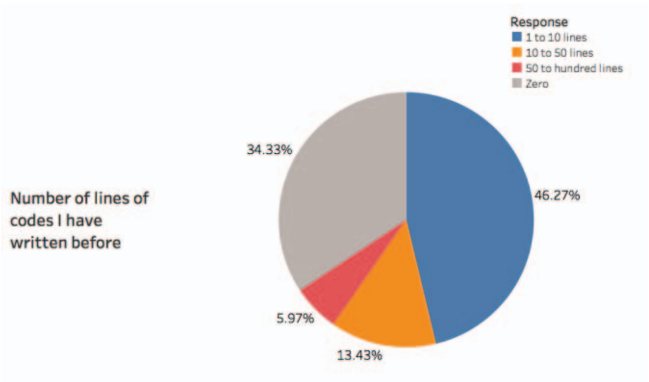

Fig. 4. Participants' programming background. About 34% of participants have not written any lines of code, and 46% of participants have only written 1 to 10 lines of code.

TABLE I. Pre-Test And Post-Test Result For Programming Knowledge and Skills

		Robot Simulator	***Physical Robot***
	Number of students	23	25
Mean	**Pre-Test**	67.9	74.5
	Post-Test	85.5	88.3
Median	**Pre-Test**	68.3	73.3
	Post-Test	83.3	91.67
Standard Deviation	**Pre-Test**	11.2	12.5
	Post-Test	7.9	8.1

To further confirm if the two groups, physical robot and robot simulator, were comparable in their programming prior knowledge, we conducted a t-test on the mean values of the pre-test scores. We did not find a statistically significant difference in the mean scores, at the p-value of 0.05. This confirmed that the distribution of the two groups were random, and both the groups were comparable.

Comparison of the pre- and post-test scores for both the groups show an increase in scores. We did a t-test to see if there is significant increase in scores. Both the groups showed a significant increase at the p-value of 0.05. This shows that the workshop did help students to gain knowledge in programming concepts and skills regardless of whether a physical robot or robot simulator was used. This can also be seen from the decrease in the standard deviation of both groups on their post-test results.

To examine if one method of teaching, that is, using the physical robot versus the robot simulator, was better than the other, we compared the mean post-test scores between the two groups by a t-test. We found no significant difference between the physical robot group and the simulator group. This is an interesting result as it suggests that both simulation and

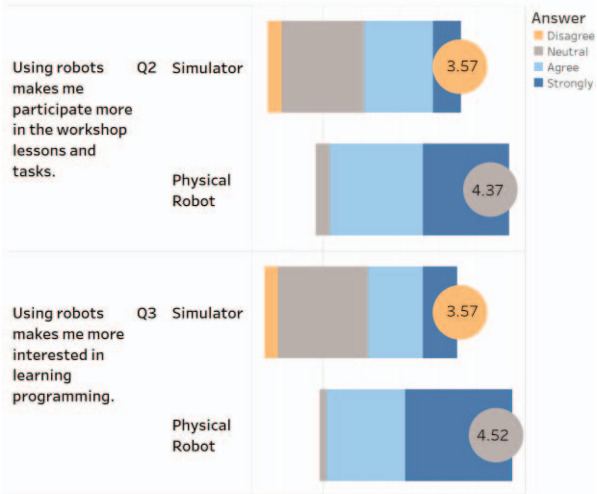

Fig. 5. Students' interest and engagement on the workshop's activity at the end of workshop. Both groups have positive scores though the physical robot group seems to have more positive scores.

physical robot are equally effective for students' learning.

Besides data from the pre and post-tests, we also analyzed student responses from the end of the workshop survey on their learning experience. Selected extracted results are shown in Figs. 5 to 8.

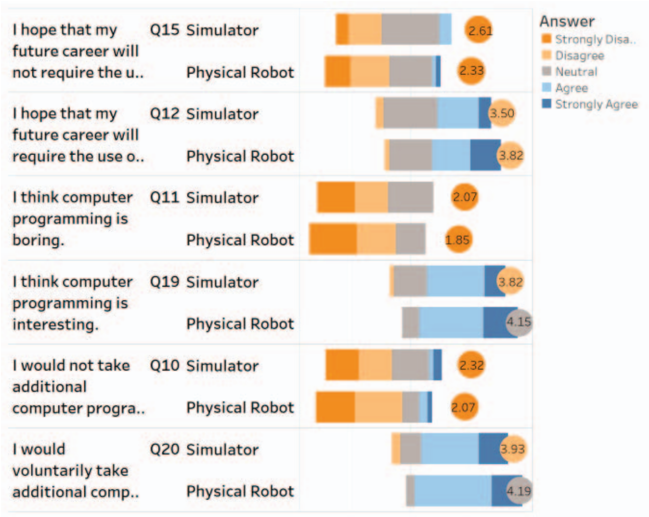

Fig. 6. Students' general interest in computing and programming at the end of workshop. Consistent results show that the physical robot group has a slightly more positive score than the simulator group.

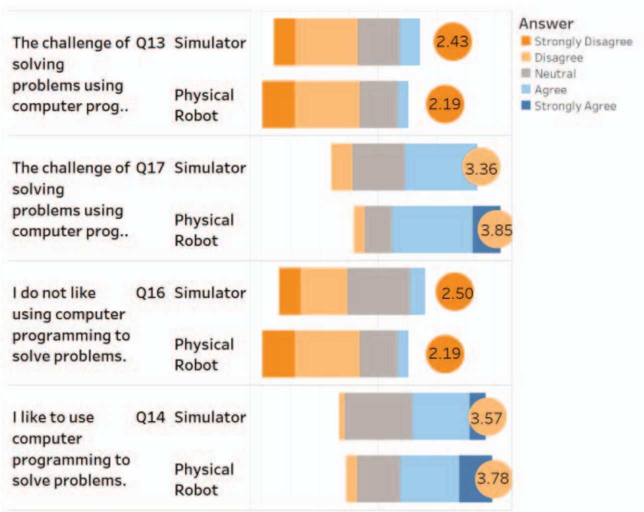

Fig. 7. Post workshop survey on students' aptitude towards solving problems. The physical robot group is consistently more positive than the simulator group.

We asked whether using the physical robot or the robot simulator made students more interested in learning programming and whether it helps them to participate more during the lessons. Fig. 5 shows the response of the participants in a Likert scale where a score of 1 indicates "Strongly Disagree" and 5 indicates "Strongly Agree". Both groups have a positive response (Likert average score > 3). This shows that both the physical robots and the robot simulator contributed to their interest and participation in the lessons. We also see that the physical robot group has a higher score for both questions. This suggests that physical robots are more effective in creating interest as well as in increasing participation.

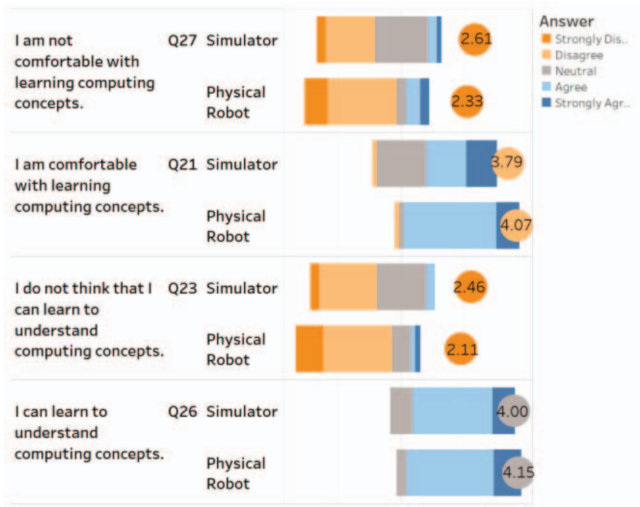

Fig. 8. Post workshop survey students' confidence. The physical robot group is consistently more confident than the simulator group.

In fact, we found that such a trend can also be observed in other survey questions. Fig. 6 shows the results of questions asking students about their general interest in computing. The questions were asked in both positive and negative terms to check for consistency. For all positive statements, both groups have scores greater than three which suggests a positive response. On the other hand, for all negative statements, both groups have scores less than three, suggesting that student responses to both the positive statements and negative statements are consistent. The physical robot group has a more positive score as compared to the simulator group for all the positive statements. Regarding the negative statements, the physical robot group has a more negative score. This result supports the possibility of physical robot to be more effective in increasing interests and motivations.

It seems that the trend in the responses between the physical robot group and the simulator group is also seen on students' problem-solving aptitude (Fig. 7). It can be observed that the physical robot group tends to have more positive scores as compared to the simulator group for the positive statements. The reverse is true for the negative statements. Similarly, Fig. 5 shows participants' response in terms of their confidence. The same trend can be observed. Overall, students from the physical robot group seem to perceive that they are more interested, capable and confident in programming than those in the robot simulator group.

The open-ended questions in the survey provided further insights. Students were asked on their best and worst experiences in using either physical robot or simulator robots. These qualitative responses were text-analysed and frequency counted. Most of the responses on the best experiences cited *visualizing*. Words related to "seeing" and "visualize" are dominant. This means that using the physical robot or the robot simulator helps them to visualize their programs.

The next frequent categories after visualizing are *fun* and *practical*. Many participants commented on having fun playing with the robots. Other participants remarked how coding knowledge can be applied to something real and related to real-life applications. The next category of responses falls under *sense of achievement*, *interactivity* and *facilitation*. Under the *sense of achievement* category, students reported the satisfaction in solving the robot challenges. On the other hand, the responses from *interactivity* category speaks on how the learning using robots are more interactive and encourages participation. Similarly, the responses under *facilitation* speaks on how using robots helps them to learn.

TABLE II. COMMENTS CATEGORY OF PARTICIPANTS COMPLAINTS

	Nil	*Hard-ware*	*Debug*	*Time*	*Not using physical, engagement*	*Soft-ware*	*Other*
Robot Simulator Group	6	3	5	1	7	3	3
Physical Robot Group	7	9	4	3	0	0	4

In the final open-ended question, we asked what the worst part was in using physical robot or the robot simulator. The category of the different complaints is presented in Table 2. The majority of responses reported *nil* or *nothing*. Both groups have similar number of such responses, suggesting that most participants do not find any difficulties and found the use of either physical robots or robot simulator to be fine. After this, comments on hardware issues had the second-highest number, coming mainly from the physical robot group. Having to tackle problems with hardware is a known issue when physical robots are used in teaching. Such problems cause students to spend more time in debugging and solving hardware issues, on top of completing the programming task. Hence, it is no surprise that the next highest reported issues were related to *debugging* and *time*. Most of the comments indicated that these come from the physical robot group where they require more time to setup, harder to debug, and time consuming. There were also comments when they indicated that they were not sure it is a hardware problem or software problem. Similar issues were also reported in [14]. Although *debugging* problems are also faced by the simulation class, we found that this class reported another kind of issue. A number of them complained that they did not use physical robots but only the robot simulator or the lesson were not engaging enough. As expected, most complaints on software related issues come from the robot simulator group.

IV. DISCUSSION

The key research question asked in this study is whether there was any difference in using physical robots and robot simulator in teaching introductory programming, in terms of conceptual learning. The results from the pre-test and post-test show that there is no significant difference in the learning gained between the physical robot group and the simulator group. This suggests that using either a physical robot or robot simulator can be equally effective in helping students to learn introductory programming concepts. These results seem to agree with most of the previous studies in surveyed in [15].

Though both approaches seem to be similar in effectiveness, our surveys seem to indicate that robots can spark a greater interest in learning computing as compared to using a simulation. From the attitude survey, both groups reported a positive interest in computing, but students using physical robots showed a larger interest compared to the simulation group.

The results of the open-ended questions also suggest that students using the physical robots are more engaged during lessons. Brauner et al. reported a similar conclusion in a study comparing the use of tangible artifacts with using only visuals for STEM subjects [17]. Their study showed that using tangible robots encouraged more positive attitudes towards STEM topics, although no considerable effects were found in terms of the understanding or learning. Wu et al. also reported that physical robot group demonstrated more positive attitudes toward the learning activities as compared to the simulation group even though there were no significant difference on students' performance between the two groups [18]. In terms of confidence, Ortiz, et. al. also showed that using physical robot showed an increase of confidence in the learning process [22]. This agrees with the result shown in Fig. 8.

The results from students' comments also show that using physical robots have the usual recognized disadvantages. Students have to tackle hardware issues and debugging can consume more time compared to a simulator, hence more hands-on time is needed compared to using a simulator. Such issues were also reported by Fagin and Merkle in [14].

What can we learn from this study? Our study suggests that, compared to a robot simulator, using physical robots would be advantageous in teaching introductory computing as it sparks students' interest in computing and makes them better engaged during lessons. While one argument could be that the conceptual learning gain between the robot simulator and physical robot is not significant, the physical robot has the additional value of arousing curiosity and stimulating motivation and interest in students.

Using physical robots, however, introduces its own set of challenges. We may overcome the disadvantages of using physical robots by taking the following steps. Firstly, students can be given the chance to run their program on a simulation to debug their program before it is deployed on the robot. Secondly, explicit instruction should be given to the class that uses physical robots to help them be familiar with the platform. This can be in the form of video tutorials which are more apt for today's generation than text instructions. Thirdly, sufficient time should be given for students to work with the physical robots to gain familiarity and confidence. This can be done by assigning some pre-class activities that students need to complete before coming to class. Lastly, students should be given the option whether to submit their work using the robot simulator or the physical robot. In this way, student work on the physical robots are not linked to any assessments which must be done within a specific deadline. This removes the anxiety of having limited time. At the same time, students who have completed their program on the robot simulator will be more willing to venture using physical robots out of interest. In this way, we take advantage of both the robot simulator and physical robots to address the different goals instructors may have. A robot simulator is sufficient to increase their learning gain, and the physical robots can be used to increase their motivation and interest in the subjects. Teaching is structured such that it breaks down the activities into meaningful chunks to scaffold learning.

It is proper for us to state the limitations of the current study. First, as the workshop is aimed at novices in programming, the content of the workshop is limited to discussing essential Python syntax and basic programming structures. We are not able to say if the results of this study will remain the same if additional content is included in the workshop e.g. functions, more complicated data structures, object-oriented programming, and so on.

Second, the instrument during the pre-test/post-test programming test is not a validated concept inventory, hence we are not able to say if the learning gains achieved in this study are significant. From a survey of the research, it seems that a concept inventory for introductory computing is still under development, and that a version for Python is not yet available [23].

Third, our study is based on a short course conducted over five consecutive days. Each day, the students spend only approximately three hours to carry out the tasks, including the robot challenge. Moreover, the number of participants in this workshop may not be large enough to make a strong conclusion. At the same time, we see that our results agrees with some other related works as shown previously. Therefore, we have to be careful in applying the results and suggestion found in this study to a full course in a university setting.

Lastly, the robot used in this study and the visuals displayed by the simulator could also be a factor. We find that the Thymio robot is relatively easier to use than other educational robots that we have experience with. This could be the reason that we have a result that is opposite to the study by Fagin and Merkle [14]. At the same time, the visuals in the simulation using Python's built-in module are extremely simple. The robot is represented only as a triangle and the path that it travels is shown by a line. We do not know yet if better visuals in the simulator will result in an increase of interest and motivation in a similar manner as seen with the physical robot.

In addition to the earlier suggested method of combining both simulator and physical robot, another possible future study is to overcome the robot simulator's lack of interactivity while maintaining its advantages. We know that physical robots have issues with hardware and it requires more time from students to debug. At the same time, students like physical robots because it is more *fun, practical,* and *interactive*. One solution that involves only software and provides a fun experience is augmented reality or mixed reality technology. With such technology, students can interact with a virtual robot as if it is a physical robot and not encounter the hardware issues associated with a physical robot.

We can also ask if the impact of the positive outcome of students' experience in using physical robots leads to deep learning and lifelong learning. Since our study was limited to five days, it was not possible to investigate this. Further study on students' learning orientation and process skills in addition to learning outcome would share more light.

V. Conclusion

In this study, we have compared the effect of using physical robots versus using robot simulator in an introductory programming workshop. The results show that the learning gains from using either tool are similar. However, students who used physical robots reported a higher interest in computing than those who used a simulator. Our results suggest that using a physical robot in teaching could result in students having more interest in learning computing. Future studies on how to incorporate such benefits while eliminating the disadvantages should be explored. One could study the impact of such positive interest on their lifelong learning orientation and skills.

Acknowledgment

The project is funded under the Pedagogy Innovation Grant 2017-6037 by Singapore University of Technology and Design.

References

[1] J. M. Wing, "Computational Thinking," Commun. ACM. vol. 49, no. 3, pp. 33–35, 2006.

[2] S. Papert, Mindstorms: Children, Computers, and Powerful Ideas, New York, NY:Basic Books, 1980.

[3] M. Resnick, "Xylophones, Hamsters, and Fireworks: The Role of Diversity in Constructionist Activities," in Constructionism, I. Harel, S. Papert, Eds. Ablex: Norwood, NJ, pp. 151–158, 1991.

[4] F. B. V. Benitti, "Exploring the Educational Potential of Robotics in Schools: A Systematic Review," Comput. Educ., vol. 58, no. 3, pp. 978-988, 2012.

[5] B. W. Becker, "Pedagogies for Teaching CS1 with Java," ACM SIGCSE Bull., vol. 33, pp. 50-54, 2001.

[6] D. Barnes, “Teaching Introductory Java through LEGO MINDSTORMS Models,” ACM SIGCSE Bull, vol. 34, no. 1, pp. 147, 2002.

[7] T. Lauwers, I. R. Nourbakhsh, E. Hammer, “CSbots: Design and Deployment of a Robot Designed for the CS1 Classroom,” in Proc. 40th ACM Tech. Symp. on Computer Science Education, vol. 41, pp. 428-432, 2009.

[8] A. Pásztor, R. Pap-szigeti, E. Török, Mobile Robots in Teaching Programming for IT Engineers and Its Effects, ”International Journal of Advanced Computer Science and Applications, vol. 4, no. 11, pp. 162–168, 2013.

[9] B. S. Barker and J. Ansorge, “Robotics as Means to Increase Achievement Scores in an Informal Learning Environment,” J. Research of Technology in Education, vol. 39, no. 3, pp. 229-243, 2007.

[10] G. Nugent, B. Barker, N. Grandgenett, V. Adamchuk, “The Use of Digital Manipulatives in K-12: Robotics, GPS/GIS and Programming,” in Frontiers in Education Conference, pp. 1-6, 2009.

[11] F. R. Sullivan, “Robotics and Science Literacy: Thinking Skills, Science Process Skills and Systems Understanding,” J. Research in Science Teaching, vol. 45, no. 3, pp. 373–394, 2008.

[12] S. Hussain, J. Lindh, G. Shukur, “The Effect of LEGO Training on Pupils’ School Performance in Mathematics, Problem Solving Ability and Attitude: Swedish Data,” Educational Technology and Society, vol. 9, pp. 182-194, 2006.

[13] D. C. Williams, Y. Ma, L. Prejean, M. J. Ford, G. Lai, “Acquisition of Physics Content Knowledge and Scientific Inquiry Skills in a Robotics Summer Camp,” vol. 40, no. 2, pp. 201-216, 2007.

[14] B. Fagin, L, Merkle, “Measuring the Effectiveness of Robots in Teaching Computer Science,” ACM SIGCSE Bull., vol. 35, no. 1, pp 307, 2003.

[15] L. Major, T. Kyriacou, O. P. Brereton, “Systematic literature review: teaching novices programming using robots,” IET Software, vol. 6, no. 6, pp. 502, 2011.

[16] N. Ç. Özüorçun and H. Bicen, “Does the inclusion of robots affect engineering students’ achievement in computer programming courses?” Eurasia Journal of Mathematics, Science and Technology Education, vol. 13, no. 8, pp. 4779–4787, 2017.

[17] P. Brauner, T, Leonhardt, M. Ziefle, U. Schroeder, “The Effect of Tangible Artifacts, Gender and Subjective Technical Competence on Teaching Programming to Seventh Graders,” Lecture Notes in Computer Science, vol. 5941, pp. 61-71, 2010.

[18] C. Wu, I. Tseng, S. Huang, “Visualization of Program Behaviors: Physical Robots Versus Robot Simulators,” Lecture Notes in Computer Science, Mittermeir, R. T., Sysło, M. M., Eds.; Springer: Berlin, Heidelberg, vol. 5090, pp 53–62, 2008.

[19] F. Riedo, M. Chevalier, S. Magnenat, F. Mondada, “Thymio II, a Robot That Grows Wiser with Children,” in Proc. IEEE Work. Adv. Robot. its Soc. Impacts, ARSO, pp. 187-193, 2013.

[20] O. Kurniawan, PyThymioDW Library. Available: https://github.com/kurniawano/pythymiodw, 2018.

[21] A. Hoegh and B. M. Moskal. "Examining science and engineering students' attitudes toward computer science." in Frontiers in Education Conference, pp. 1-6, 2009.

[22] O. O. Ortiz, J. A. P. Franco, P. M. A. Garau, R. H. Martin, “Innovative mobile robot method: Improving the learning of programming languages in engineering degrees,”. IEEE Transactions on Education, vol. 60, no. 2, pp. 143–148, 2017.

[23] R. Caceffo, S. Wolfman, K. Booth, “Developing a Computer Science Concept Inventory for Introductory Programming,” in Proc. 47th ACM Tech. Symp. on Computing Science Education, pp. 364-369, 2016.