CERN IdeaSquare Journal of Experimental Innovation, 2021; 5(2): 56-60
DOI: https://doi.org/10.23726/cij.2021.1077

ORIGINAL ARTICLE

A quantitative experiment setup for testing the effect of desirable difficulties on teaching robotics

Achim Gerstenberg,[1*] Martin Steinert[2]

[1] Center for Interdisciplinary Research in Space, NTNU Samfunnsforskning, 7491 Trondheim, Norway
[2] Department of Mechanical and Industrial Engineering, NTNU, 7491 Trondheim, Norway
*Corresponding author: achim.gerstenberg@samforsk.no

ABSTRACT

Desirable difficulties such as generating one's own solution instead of replicating a provided solution is associated with improved long-term memory. Disseminating misleading information has shown improved learning in science education over consuming concise and clear learning instructions. We describe an experimental setup aimed at quantifying if a tutorial about programming a mobile autonomous robot that requires having to correct misleading instructions leads to better problem-solving capabilities than providing correct and clear tutorial instructions when asked to solve a complicated open-ended robotics task. The presented experimental setup is aimed towards a controlled comparative human-subject study that compares the effect of desirable difficulties on participant's performance in solving a complicated open-ended task after completing an introductory tutorial. We explain the experiment timeline, the tasks of the tutorial, as well as the open-ended task including the robot and how this experiment can be executed under very controlled, repeatable and as unbiased as possible conditions. We share and qualitatively discuss some observed problems in this setup from early trials with 8 participants.

Keywords: Desirable difficulty; human subject experiment; problem solving.

Received: August 2020. Accepted: December 2021.

INTRODUCTION

Desirable difficulties such as generating a solution instead of being told a solution (Bjork 1975; Bjork 1994; Bjork 2011) and using tests instead of presentations (Roediger & Karpicke 2006) have been shown to improve long-term retention of the learning content. In addition, Muller found that noticing and correcting false information in science education videos significantly improved the understanding (Muller 2007).

We present an experimental setup for a comparative study that quantitatively investigates if introducing desirable difficulties improves short-term performance in open-ended problem solving. In the learning phase, we use flaws in an example solution that the participants need to correct as a generative desirable difficulty. We furthermore describe how such quantitative experiments can be conducted under controlled conditions to minimize biases introduced by inconsistent interaction with the participant.

EXPERIMENT AGENDA

During a series of tutorial tasks that familiarize the participants with the programming of a robot, the participants are either presented a working solution to the task or a dysfunctional, flawed solution that they need to correct. In addition, all participants receive an explanation of the library functions needed. The flawed or unflawed solutions are presented on a separate screen so that participants cannot automatically copy and paste the solution into the programming environment. This ensures that they need to either create their own solution or need to fully read and then retype the provided solution. After completing nine tutorial tasks, which introduce the participants to the functionalities of the robot necessary for this experiment, all participants are presented an open-ended task. The time from when the robot is started until it has completed the open-ended task is used as a measure of performance. The evaluation takes place after the programming phase. We suggest a programming phase of eighty minutes.

This experimental setup aims to research if desirable difficulties increase problem-solving performance in an open-ended task. Accordingly, the hypothesis to be tested is: Participants that correct flawed solution suggestions in tutorial tasks afterwards generate solutions that solve the open-ended task faster than participants that received unflawed solutions during the tutorial tasks.

The following paragraphs describe the robot used, the tutorial tasks and the open-ended task.

Additional information about the physical setup, a more precise description of the robot and the library, the

2413-9505 / © The authors, 2021
Published by CERN under the Creative Common Attribution 4.0 Licence (CC BY 4.0)

standardized interaction with the participants and other experiment examples this setup can be used for are described in (Gerstenberg & Steinert 2018).

THE ROBOT

The robot is based on Lego Mindstorms NXT 2.0 and has two motors to drive forwards, backwards and turn, and sensors to detect the reflectivity of the surface underneath the robot, an ultrasonic distance sensors and light sensors that can detect the color of an object and recognizes a blinking light. It is programmed in the NXC (not exactly C) programming language. A library that simplifies the programming of the robot is provided to the participants. This library includes functions for sensors, outputs and movement. For example, a function that separately defines the PID controlled motor speeds on each side in percentage values from negative 100 to positive 100, spares the participant the need to program a closed feedback loop for the voltages to each motor. Other functions simplify turning of the robot, reading out sensor values, generating random numbers, play tones and display characters on the robot's screen. Figure 1 shows a front view of the robot.

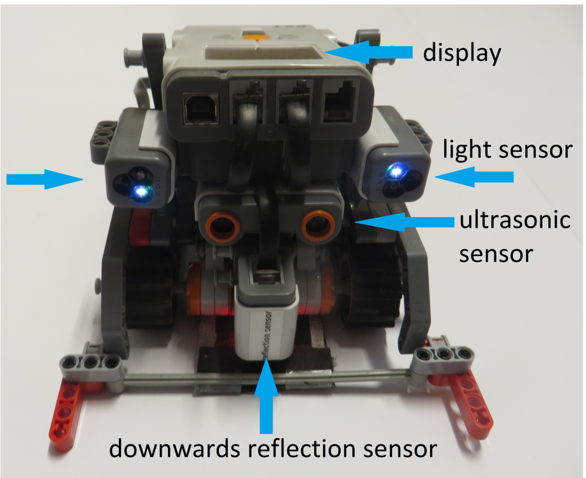

Fig. 1. Detailed front view of the robot with its four sensors. The motors and belts for moving the robot are positioned on the sides and are not visible.

TUTORIAL TASKS

The aim of the tutorials is on the one hand to introduce the participants to the robot and its capabilities and on the other hand to provide the differentiation between the flawed and unflawed condition. The tutorial consists of 9 separate tasks where each tasks introduces one new functionality from the library. While working on the tutorial the participants can execute the codes that they have created. They are presented the next tutorial task when the robot fulfils the current task. In the case that the experimental group, that needs to find the flaw in the solution, does not solve the task within the time limit they are shown the solution for 1 minute. The following table lists the nine tutorial tasks with a short description of the flaw used for the experimental group and the provided time limit.

#	Task description	Flaw description	Time limit
1.	Write a program that each second plays a sound for 100 ms at 440 Hz.	Sound has 4400 Hz and the time between sounds is 1 ms instead of 1 second	3 min
2.	Show the elapsed time in seconds on the display.	Time variable starts at 42 seconds and the display position changes each second instead of the value	3 min
3.	Drive straight forward as slowly as possible (but visibly moving) for 5 seconds then turn 90 degrees counterclockwise with speed 10 using the turn function. When the turn is completed, drive straight backwards as fast as possible for half a second and then stop both belts for two seconds. Spin the robot clockwise for 2 seconds at half speed using both belts and stop.	Robot turns in the opposite direction in both cases and uses a forbidden speed value.	5 min
4.	Place the robot on the aluminum foil and drive forward as long as the robot is on the aluminum foil. Use the downwards light reflection sensor to find out when the robot (or more precisely the sensor) reaches the edge of the aluminum foil and stop the robot.	Too low threshold for detecting the edge of the aluminum foil. The robot keeps driving after reaching the edge.	5 min
5.	Place the robot in front of the red or green cardboard wall. Drive towards the wall. If the wall is green stop in front of the green wall. Keep driving and crash into the wall if it is red.	Robot only compares the green light reflection between the sensors on each side and does not compare to the red.	5 min

6.	Place a blinking light 20 cm behind the robot. Rotate the robot on the spot until the robot detects a blinking light. Stop the robot when it detects the blinking light.	Robot turns so quickly that the light sensors are not facing the blinking lights for the duration of one blinking period and consequently are not properly detecting the lights.	5 min
7.	Place the robot facing towards the 50 cm wide cardboard wall and drive towards it. Stop between 20 and 30 cm in front of the wall using the ultrasonic distance sensor. Turn the robot until the ultrasonic sensor no longer detects it and drive past the cardboard wall.	The threshold is chosen lower than the minimum distance the sensor can detect and therefore the robot keeps driving into the wall.	5 min
8.	Drive around with random speed (between full speed backwards and full speed forwards) on each wheel by using the random function. Change wheel speeds every second	The input order of the parameters to the random() function is inverted resulting in the function returning 0 instead of random values between -100 and 100.	3 min
9.	Equivalent to the previous task but stop the robot after 5 seconds using timers	Timer 1 is started but timer 3 is used for the timing. After this is corrected the time until the program stops is 5 milliseconds instead of 5 seconds.	3 min

OPEN-ENDED TASK

The aim of this task is to attain a performance measure that allows for a quantitative comparison of the two conditions. The task is open-ended, meaning that there is not one single clear solution and many different ways of finding them. While the tutorial tasks each contain a single new code component, the open-ended task requires a creative combination of several components to be solved. The participants need to transfer knowledge from the tutorials and adapt it to the new context of the open-ended task.

The task is to remove a green and blue cube-like object from a white area of 1.8 square meters in the shortest time possible after starting the robot. The red object shall not be removed. The starting position of the robot is unknown to the participant and the solution is supposed to work from any possible starting position. Up to three blinking lights are provided and can be placed anywhere including inside the cube-like objects. The robot cannot be manually influenced after it is started.

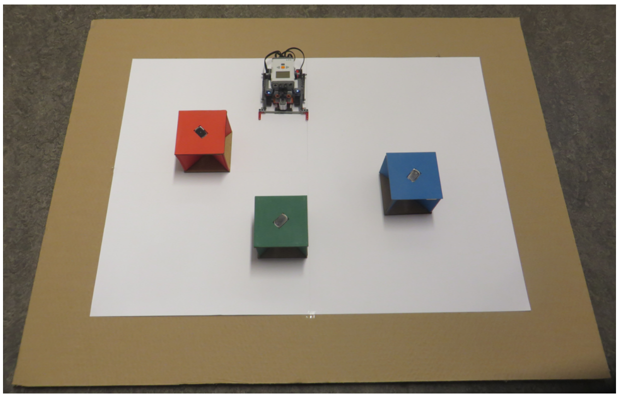

Fig. 2. Setup of the three coloured cubes on the 1.8 square meter large white area that the autonomous robot needs to push onto the surrounding.

INTERACTION CONTROL

Qualitative research offers insights into why differences occur while quantitative research setups, like the one presented here, allow researchers to quantify effect size with usually fewer insights about why a difference between conditions exists. This means that it is important to have one exclusive difference between conditions such that differences in the results can be linked to this single stimulus. Therefore, everything else that may have an influence on the results needs to be kept equal for every participant.

Common biases in human subject experiments are dependent on the behaviour of the experimenter and the perceptions of the participant. Apart from carefully designing and testing instructions, we cannot control for how a participant perceives and interprets them. However, we can control how the instructions are presented and can reduce biases introduced by direct human-human interaction with the experimenter. This experiment setup is designed without any direct oral and visual experimenter to participant interaction by providing instructions through pre-recorded voice and video instructions that use computer generated voices to avoid emotional inflictions through voice tonality. Other instructions are given as text on a screen or paper, and the timing when information is presented follows a predefined script. This process ensures that each participant receives neutral and similar instructions.

PRELIMINARY RESULTS

The experiment was conducted with 7 male participants and 1 female participant. 5 participants were in the experimental condition with the flawed tutorial tasks and 3 participants were in the control condition. The participants were between 23 and 24 years of age and recruited on a voluntary basis from a mechatronics course of a mechanical engineering study program.

All participants successfully completed the tutorial tasks in the provided time without additional help. However, only 1 participant from the flawed condition and 1 participant from the control condition successfully programmed a solution that removed the blue and green objects and thus solved the open-ended task. Therefore, only in two cases can the performance measure, namely the time the robot takes to remove the objects, be measured. From these two data points no statistically significant result can be concluded.

The participants showed qualitative differences in programming skill and behaviour. A major qualitative difference was how often the participants tested their designs and how quickly they learned from testing.

LIMITATIONS AND STRENGTHS

Given that only two out of eight participants found a solution to the open-ended task one can say that the difficulty of the task was not matching the skills of the recruited participants. Although all participants were recruited from a mechatronics programming course their programming skills were insufficient for the experiment and differed greatly between participants. The tutorials and the simplifying programming library did not provide sufficient support to overcome the lack of programming skill or equalize skill differences. We believe that this limitation can be overcome by recruiting more skilled participants from a very homogeneous group. This would restrict the validity of the study to this narrowly selected type of participant but it also allows statistically significant findings with fewer participants.

Probably the main limitation of the experiment setup is that different participants have varied methodological approaches to open-ended problem solving and this difference has a notable influence on the programming performance. For example, some participants read the provided information about the robot more carefully and approach the problem in a very structured way, while others rely on frequently testing prototypes. Furthermore, the ability to gain insights from testing varies between participants. Although this cannot be statistically concluded from the quantitative results, we see this as a major concern that may be a decisive confounding variable that overshadows the possible effect of introducing desirable difficulties during the tutorial. It led us to eventually adapt the experiment design towards investigating the influence of early prototype testing onto robot development (Gerstenberg 2018; Gerstenberg 2019).

Having an experimental setup that is precisely predefined and largely automated limits the flexibility of quickly adjusting the study setup. The desirable difficulties presented in the tutorial need to be difficult, yet solvable for the participants. Finding this challenge point requires already extensive testing prior to the actual experiment with a group of participants that have a similar skill level as the participants used later in the actual study. Since participants cannot partake in the study twice, those used in the preliminary study will not be able to participate again and finding enough suitable participants to gain sufficient statistical power can become a challenge and a compromise between efficiently designing an experiment setup versus improving experimental control.

An additional downside to a rigid setup is that it makes qualitative research more difficult. The setup allows for standardized qualitative data gathering such as analysing the codes the participants write, their keystrokes, video and audio recordings and questionnaires. However, such a setup does not allow for situational inquisitions like interviewing the participant during the experiment or asking for feedback on specific design decisions or on potential problems. Direct human interaction between the experimenter and the participants is possible after gathering the data and retrospectively asking question may still lead to insightful observations.

Another limitation and simultaneously a strength is that each participant works alone. This makes the experiment less meaningful as programming nowadays is often done in groups, but it also eliminates the influence of group interactions on the experiment which is difficult to control for.

The aforementioned experimental setup is optimized towards repeatability with a special focus on how to provide equal experiences to every participant by standardizing instructions and presenting them in an as neutral as possible form. This reduces the possibilities for induced biases and thus for confounding variables and we see this as an important step towards more controllable experiment setups. However, it is only meaningful when more influential confounding variables can be avoided. The current experiment setup needs improvements to minimize inter-participant behavioural differences to allow for meaningful conclusions.

Albeit the experimental setup described in this paper shows challenges concerning confounding variables, we aim to contribute with an experiment idea and setup for testing the hypothesis if desirable difficulties can improve open-ended problem solving. We hope to encourage the further development of quantitative studies in the field of computer science education with highly controlled interactions between the participants and the experimenter.

ACKNOLEDGEMENTS

This research is supported by the Research Council of Norway (RCN) through its user-driven research (BIA) funding scheme, project number 236739/O30

REFERENCES

Bjork, R.A., 1975, Retrieval as a memory modifier: An interpretation of negative recency and related phenomena. Information processing and cognition, Loyola Symposium, 123 – 144.

Bjork, R.A., 1994, Memory and metamemory considerations in the training of human beings, Metacognition: Knowing about knowing, 185 – 205.

Bjork, R.A., 2011, Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning, Psychology and the real world: Essays illustrating fundamental contributions to society, Worth Publishers, New York, NY, US.

Gerstenberg, A. & Steinert, M., 2018, Open ended problems – a robot programming experiment design to compare and test different development and design approaches. Proceedings of NordDesign 2018.

Gerstenberg, A. & Steinert, M., 2019, The Relevance of Testing in Engineering Product Design – Investigations on a Robot Programming Task. Procedia CIRP 84, 586-592.

Muller, D.A., 2007, Saying the wrong thing: improving learning with multimedia by including misconceptions, Journal of Computer Assisted Learning, 24(2), 144 – 155.

Roediger, H.L. & Karpicke, J.D., 2006, Test-enhanced learning: Taking memory tests improves long-term retention, Psychological Science, 17, 249–255.