

A Literature Review Exploring the use of Programming in Mathematics Education

Sanna Erika Forsström
Østfold University College
Halden, Norway

Odd Tore Kaufmann
Østfold University College
Halden, Norway

Abstract. Programming is now included in mathematics curricula in several countries; thus, the purpose of this literature review is to determine the research-based justifications for these educational decisions. From a selection of relevant articles, 15 articles were identified and analyzed, each of which had varying study types, themes, and designs. Three themes from the studies were identified: the motivation to learn mathematics, student performance in mathematics, and the collaboration between students and the changed role of the teacher. It was found that in certain circumstances, including programming in mathematics education could improve student motivation to learn mathematics and improve student performance in mathematics. To gain a better understanding of the potential of programming in mathematics education, the entire collective learning process should be considered by discussing the roles of the teacher and the collaboration between students as part of these roles.

Keywords: mathematics education; programming; robots.

1. Introduction

We are facing the fourth industrial revolution, which is characterized by a range of new technologies that are fusing the physical, digital and biological worlds, influencing all disciplines, economies and industries (Schwab, 2017). According to Balanskat and Engelhardt (2015), in the future, many of today's students will be involved in developing technology, which is important for the society. Consequently, programming skills have become increasingly important core competencies for 21st-century skills and have become important in education policies seeking to adapt the education sector to meet future societal demands. Many countries have recognized that programming needs to be integrated into school curricula to equip students with skills, such as problem solving and

logical thinking, which are important in today's digital society. The challenge for the education sector, therefore, is to give students the competencies to master and create their own digital technologies and to prepare them for the future; therefore, learning how to code and program in formal and non-formal education settings is vital.

As programming has come to be recognized as a basic skill for effectively participating in the digital world, there has been increasing interest during the past decade in introducing programming as a school subject (Grover & Pea, 2013). Programming is the process related to the development and implementation of instructions for computer programs so the computer can perform specific tasks, solve problems, and support human interactions. Therefore, programming generally requires programmers to have a knowledge of programming languages; expertise in subjects related to the development of specialized algorithms and logic; and the ability to analyze, understand, and solve problems by verifying algorithmic requirements and assessing the correctness and implementation (often referred to as coding) of the algorithm in a particular programming language. Because these processes have been linked to mathematical thinking, several European countries have claimed that since programming is related to the development of algorithmic thinking (Grover & Pea, 2013), it is an important skill for the digital society and the 21st-century skills of problem solving, creativity, and logical thinking. While there have been many different suggestions as to where programming might fit, there has been little consensus on how to include programming in school curricula (Grover & Pea, 2013), with the debate focusing on whether it is part of Information and Communication Technology (ICT) or whether it should be integrated across the curriculum. Increasingly, schools have integrated programming into other subjects, mostly mathematics, using cross-curricular approaches (Balanskat & Engelhardt, 2015). Finland and Sweden, for example, have both integrated programming into mathematics (Bocconi, Chiocciariello, & Earp, 2018; Opetushallitus, 2014; Skolverket, 2018) with the rationale that it fosters problem-solving and logical-thinking skills and motivates students to learn mathematics. Norway is planning to integrate programming in mathematics in the revised version of the curriculum in 2020 (Bocconi et al., 2018). According to Bocconi et al. (2018), further discussion is needed, however, on the ways in which programming can be linked with other subject areas and the degree to which it influences student achievement. Furthermore, a need exists for a discussion of the type of pedagogical solutions that are effective by considering concrete implementations of programming using a variety of tools and assessments. One relevant topic to consider regarding pedagogical solutions is the role of the teacher. When integrating programming with a mathematics curriculum, the role of the teacher may become challenging because the mathematics teacher may not have previous knowledge of programming.

Using programming in mathematics education is not a new concept. As early as 1980, Papert (1980), who associated learning through programming with Piaget's constructivist learning theory, developed a Logo environment that required students to program a computer to steer a turtle on a computer screen, with the

intention of providing a different environment for learning mathematics and motivating students to engage with mathematics.

Based on Papert's Logo environment, Yelland (1995) examined "the potential of Logo to act as a mathematical environment" (p. 853) in a review article examining the relationship between cognitive gains, problem-solving, and social interaction skills in student mathematics achievements in dozens of quantitative and qualitative studies. Yelland found that there had been varying results regarding the cognitive gains in problem-solving and mathematics achievements. Yelland (1995) noted, however, that Logo was a useful learning environment for students from both individual and group perspectives and that it was a helpful way for researchers to understand the thinking and learning processes since Logo gave students the opportunity to explore mathematics in a meaningful way. However, the outcomes were contradictory. Although some studies showed evidence of the positive impact of Logo's inclusion in mathematics, others failed to detect any differences in the students' problem-solving skills and mathematics achievements after completing the Logo programming projects.

Since Yelland's review in 1995, some significant technological developments have resulted in a number of different programming environments for classroom use, such as Scratch, and programmable robots, such as Lego Mindstorms. Benitti's (2012) literature review, "Exploring the educational potential of robotics in schools: a systematic review," examined ten quantitative studies on the educational potential of robotics and concluded that even though some studies had found no differences in the students' learning, robots were useful in understanding science, technology, engineering, and mathematics (STEM) concepts. For example, positive mathematics achievements were found for certain topics, such as circle geometry and degrees, fractions, and proportions, and for certain groups of students, such as those with average scores or in certain grades (Benitti, 2012; Lindh & Holgersson, 2007; Nugent, Barker, & Grandgenett, 2008). However, the studies did not find any improvements in student achievement for certain topics, grades, or groups of students, such those with high and low scores (Benitti, 2012; Lindh & Holgersson, 2007), and some of the results for student problem-solving skill developments were ambivalent. Based on various mathematics and problem-solving tests, such as pre- and post-tests with control groups (Benitti, 2012; Hussain, Lindh, & Shukur, 2006; Lindh & Holgersson, 2007), some studies (Nugent, Barker, Grandgenett & Adamchuk, 2009) found positive results, while others (Benitti, 2012; Hussain et al., 2006) detected no improvements.

In a more recent review, "How have robots supported STEM teaching?," Benitti and Spolaôr (2017, p. 104) analyzed 60 studies from 2013 to 2016 and found that technology and engineering education appeared to benefit most from the inclusion of robotics; however, the potential use of robots in mathematics education was seen as a support tool. In general, robotics education tended to be used as part of extracurricular activities (57% of studies) or out-of-school activities (25% of studies) rather than as part of general curricular activities (18%

of studies). Benitti and Spolaôr argued that one reason for this phenomenon may have been the teacher's poor or inaccurate knowledge of robots; however, this argument requires further evidence. Regardless, the most often-observed skills development associated with robotics was problem-solving and teamwork skills; however, even though teamwork has been commonly connected with robotics education, Benitti and Spolaôr (2017) reported that only three of the 60 studies included collaborative learning theories, all of which had been out-of-school activities.

Overall, in the current climate, Yelland's (1995) review is now out of date, and Benitti (2012) and Benitti and Spolaôr (2017) only examined robots rather than programming and only discussed mathematics as part of their reviews. Because programming has become or is becoming compulsory in the education of students aged 6–16 in many countries and because some countries have already integrated programming into mathematics curricula (Balanskat & Engelhardt, 2015), the contribution of this paper is to provide an updated review of studies on the use of programming and robots in mathematics education for students aged 6–16.

The main aim of this article is to answer the following question: What is the educational potential of programming in mathematics education?

2. Methodology

To answer our research question, we conducted a literature review with a systematic search and selection of articles. The implementation of the search, selections of the articles and design and theme of the selected articles are presented in this section.

2.1 Planning and conducting the review

To answer the research question, studies that examined programming, coding or robots and mathematics education for students aged 6–16 were searched for using the following search terms: teach*, learn*, education*,robot*, Lego, programming, coding, school, K-12 and mathematics*.

The search in the databases employed the Boolean operators AND, OR, and NOT and used the search terms in the keywords, topics, titles, and abstracts of the articles. For the final search, five databases were used: IEEE XPLORE, ScienceDirect, Education Resource Information Center (ERIC), Wilson Education, and the Web of Science. Articles written in peer-reviewed English language journals published between 1995 and 2018 were searched to identify all articles written since Yelland's 1995 review.

The initial search identified several articles that were outside the scope of interest. We therefore developed five additional exclusion criteria:

1. The article does not deal with programming, coding, or robots.
2. The article does not deal with education.
3. The article does not deal with mathematics.
4. The article does not deal with students aged 6–16.

5. The article does not outline a research design and research questions (these articles are often experience based).

Initially, both authors independently read the titles, article abstracts, and whole articles to determine which to select for further reading. After careful independent second readings of the whole articles and following thorough discussion, the relevant articles for this review were chosen. The articles that were not relevant dealt with the technical details of robots or had no empirical data from the schools. Articles that only focused on ICT education without reference to mathematics education and articles analyzing activities outside the classroom, such as summer camps, were also omitted.

Table 1 shows the articles identified in the search, the number selected for further reading, and those selected for the final analysis.

Table 1: Article selections

Database	Articles	1st selection	2nd selection
IEEE XPLORE	166	11	0
Web of Science	197	11	7(4 duplicates with ERIC)
ERIC	143	28	8
ScienceDirect	261	3	3 (2 duplicates with ERIC)
Wilson Education	150	20	11 (8 duplicates with ERIC)
Total	917	73	15

2.2 Design and theme of the studies

After careful reading, we identified and compared the themes and designs of the studies and selected four dominant themes for further discussions on the educational potential of programming in a mathematics education: students' motivation to learn mathematics, students' performance in mathematics, and collaboration between students and the changed role of the teacher. Table 2 shows the themes and designs for the 15 relevant articles identified for the literature review. However, it was difficult to determine the design for some studies.

Table 2: Articles and the robot type or programming language, topics, student ages, and data analysis methods.

Article	Robot type/ software	Topic	Age	Methods	Duration of the data gathering
Lambic (2011)	C++ Builder	The motivation of students to learn mathematics	13-19	114 participants, pre- and post-questionnaires	9x45 min
Moreno-León, Robles, and Román-González (2016)	Scratch	Impact of introducing programming in several subjects: 1. academic performance, 2. student perception, 3. assessment of projects with Scratch	11-12	129 students, experimental and control groups, pre- and post-tests	8 weeks
Taylor, Harlow, and Forret (2010)	Scratch	Potential of Scratch to enhance mathematical and technological thinking	9-10	60 students, observations, video recordings, teachers blogs, teacher interviews	
Lindh and Holgersson (2007)	Lego Mindstorms	Pupils learning', learning context/classroom environment, the role of the teacher	11-12 15-16	322 students, experimental and control groups, observations, interview, inquiry	12x8h (12 months, 2h/week)
Hussain, Lindh, and Shukur (2006)	Lego Mindstorms	Pupils' learning, learning context/classroom environment, the role of the teacher	11-12 15-16	322 students, experimental and control groups, observations, interview, inquiry	12x8h (12 months, 2h/week)
Khasawneh (2009)	Logo Programming	Student achievement, correlation between achievement in Logo programming and school mathematics achievement. Problem-solving ability	12-13	228 students, post-test	15x45 min

Bartolini Bussi and Baccaglioni-Frank (2015)	Bee-bot	Semiotic potential of bee-bot with learning of rectangles	6-7	18 students, observations, photos, graphical productions, video recordings	4 months (15 sessions)
Falloon (2016)	Scratch Jnr. on the iPad	General thinking skills	5-6	32 students, audio capture on iPads	5x(25-40)min
Ardito, Mosley, and Scollins (2014)	Lego Mindstorms, Turtle Art	Student mathematical understanding, student experiences and practice in problemsolving and collaboration	11-12	Teacher interviews, classroom observations, State exam	14 weeks
Ke (2014)	Scratch	Student participant attitudes toward mathematics before and after game-making activities, mathematical thinking	13-16	64 students, pre- and post-inventory	6x1h
Leonard et al. (2016)	Lego Mindstorms	STEM attitudes, computational thinking, self-efficacy in technology	13-16	124 students, pre- and post-survey	60 h
Barak and Assal (2018)	Robots	Students' working patterns, achievements and difficulties in learning a STEM-oriented robotics course, impact on student motivation to learn STEM subjects	13-14	32 students, pre- and post-questionnaires	15x90 min
Sinclair and Patterson (2018)	Dynamic geometry environments	How computational thinking and mathematical thinking relate?	14-16	Student sketches	2 years

Husain, Kamal, Ibrahim, Huddin, and Alim (2017)	Scratch	Mathematical thinking skills, problem solving	10-12	95 students, pre- and post-tests	1,5 day
La Paglia, La Cascia, Francomano, and La Barbera (2017)	Lego Mindstorms	Mathematical and metacognitive skills, reasoning and problem-solving capabilities, attitudes toward mathematics	10-12	60 students, experimental- and control groups, questionnaires	10x3h

3. Results

We will discuss four different dominant themes from studies on the educational potential of programming in mathematics education: the students' motivation to learn mathematics, the students' performance in mathematics, and the collaboration between students and the changed role of the teacher. In this section, these four themes are separately discussed. Articles that mentioned student interest, attitudes, mind-set, contribution, engagement, joy or happiness in learning mathematics are discussed under the motivation category. Student performance refers to students' academic achievements as measured quantitatively in test results as well as improvements in students' mathematical thinking and problem-solving skills based on similar quantitative and qualitative data, such as classroom observations, teacher and student interviews, and teacher and researcher blogs. Increased collaboration between students and the changed role of the teacher are discussed as potential in students' learning processes in mathematics.

3.1 Student motivation to learn mathematics

In this section, we review how programming and robots affected the student motivation to learn mathematics. Five articles discussed students' motivation or attitudes to learning mathematics or their interest in STEM topics.

Regarding our analysis, programming provides an opportunity for students to connect mathematics to real life in a new way; thus, programming has the potential to influence their attitudes toward mathematics (Ke, 2014; Lambic, 2011). La Paglia et al. (2017) discovered that using Lego Mindstorm robots improved their attitudes towards mathematics. Barak and Assal (2018) found no significant change in students' attitudes toward STEM topics in tests before and after their activities with Lego Mindstorm robots because students' motivation was already quite high before the programming activities. In Leonard et al. (2016), no significant changes were observed in either students' STEM attitudes or interest in STEM careers during the intervention period.

Regardless, although some studies found that programming and the use of robots motivated students or improved their attitude toward mathematics, generalizing these findings was not possible. First, each study was conducted outside the mathematics classroom as an extracurricular activity or as part of a science or technology education. Leonard et al. (2016) focused only on underserved and underrepresented students. In Ke (2014), only 20% of the participants were native, and the student age range in Lambic (2011) was wider than other studies. In all studies, the student arrangements were different from regular classroom activities since they were extracurricular activities or there were additional people in the classroom, which could affect student motivation. Second, no evidence was provided on what happened over time, especially when the programming activities were integrated into normal classroom routines. Therefore, it was not possible to assess whether programming enhanced motivation. Furthermore, a comparison of the study designs showed that the motivation to learn mathematics was examined as part of a broader study that also examined other STEM subjects, computational thinking, and metacognitive skills.

3.2 Student performance in mathematics

Five of the articles quantitatively examined student learning by measuring changes in the students' grades or test results. All of these studies had something positive to say about students' learning mathematics after the test periods with programming. Even if the test results did not show any improvements in some of the cases, each study reported positive improvements for some groups. Moreno-León et al. (2016) found the use of Scratch to have accelerated the mathematics learning of the experimental group; however, the effect was larger for social studies. Lindh and Holgersson (2007) and Hussain et al. (2006) were based on the same study and data. It was found that the Lego Mindstorms robot activities were possibly useful for some groups; however, there was no overall effect. While the fifth-grade students' mathematics results improved after the Lego training, there were no changes for the ninth-grade students and no noticeable improvements in problem-solving skills in either the fifth- or ninth-grade students. The teacher in Ardito et al. (2014) found that the students showed improvements in some mathematical topics, such as area and circumference, the quantitative data did not support these findings when comparisons were made across the whole state. However, other data indicated that the students had better results in problem solving and logical thinking. Khasawneh (2009) compared student mathematics achievements with student Logo programming achievements. A positive but low correlation was found in seventh-grade students.

Even if the studies brought out some positive effects on student performance, the results are not generalizable. First, the improvement was only visible in certain groups. Second, the comparison concentrated on different components in different studies. For instance, the comparison in Khasawneh (2009) focused on programming achievements. Ardito et al. (2014), Lindh and Holgersson (2007), and Husain et al. (2017) used tests that corresponded to the national tests in the

countries in which the studies were conducted. Third, improvement was shown only in certain mathematical topics because some of the topics were better suited to programming activities. According to this review, programming tasks are often connected to geometry. The connection between circle geometry and robotics activities is natural since in some cases, the students needed to determine the circumference of the robot wheels when programming the robot to move a certain distance (e.g. Leonard et al., 2016). Additionally, the activities with Scratch often are connected with geometry. To use plane geometry, such as squares, triangles, circles and angles, is common in the Scratch activities (e.g. Moreno-León et al., 2016).

3.3. Collaboration between students and the role of the teacher

Different pedagogical practices, such as collaboration between students and the role of the teacher in the classroom, were part of the discussion in the articles. Student collaboration was widely used in programming and robot-based activities. Based on the articles, the collaboration between students depends on several different ways on the role of the teacher in the classroom. First, the role of the teacher as a support and guide instead of as a lecturer enables students to solve problems in groups (Taylor et al., 2010). Second, the teacher acts as a conflict solver in the classroom. The teacher needs to be present with arguments when students face challenges and their collaboration breaks down. The teacher is able to make the collaboration work again by discussing the problems with the students (Hussain et al., 2006; Lindh & Holgersson, 2007). Third, the classroom climate created by the teacher is important for students' collaboration. According to Taylor et al. (2010), the classroom culture, in which students respect others' views and listen to each other, increases the collaboration among students and changes in their ordinary roles. The free environment in the classroom provides opportunities for students to adopt group roles that are different from that of their mathematics group's ordinary lessons. Students who are normally categorized with low ability can gain the opportunity to lead the group and come out with sophisticated mathematical ideas.

Because the role of the teacher in the classroom affects student collaboration, the collaboration between students in programming activities affects their learning. Through collaboration and knowledge sharing, students gain an opportunity to learn from each other (Falloon, 2016; Taylor et al., 2010). Students share their knowledge in and among the groups (Barak & Assal, 2018; Hussain et al., 2006). Knowledge sharing even affects students' choices in their problem-solving strategies. As Hussain et al. (2006, p. 188) stated:

"One way to learn by children is by a "trial-and-error method." Another way is more "cooperative": by asking their fellow workers. Alternatively, they ask another pupil in the class that is considered to know the material much better than oneself."

The articles reported a strong cohesion in the student groups; even so, students saw themselves as a group instead of as individuals who are conducting tasks by viewing their achievements as the group's achievements (Ardito et al., 2014). Barak and Assal (2018) reported that students' success and achievements as a

group provided more valuable feedback than the teacher's feedback for the students. Regardless, learning processes with programming activities depend on the choices that students make as a group during their problem-solving activities (Taylor et al., 2010); thus, learning using programming activities cannot be predicted beforehand.

No deeper discussions occurred on the effect of collaboration, especially on students' mathematics learning. Furthermore, even if studies discuss collaboration as an important factor in students' learning, the learning is viewed in most studies as a change in individual knowledge instead of something that the group achieves as a group through collaboration. As in Khasawneh (2009, p. 623), students work individually only because of the assessment: "Often students work in groups in order to cover the turtle activities in the textbook. In the meanwhile they work individually for the purpose of assessment."

4. Conclusion

The fourth industrial revolution will create both opportunities and challenges. The digital technologies merge with physical, biological and economic systems. In the long term, this will create upheavals for all industries and technologies. We have to adapt these changes, and we must understand new technology and acquire skills such as critical thinking, computational thinking and interdisciplinary to handle these changes. The schools have a social mandate. This mission starts with the individual student, who is to acquire knowledge, skills and competencies, and educate and mould students to become citizens who will support and continue the society. Therefore, we need to adapt new digital technologies in schools. In addition, we need to know more about how we can integrate such technology in a school environment, as for instance, the use of robots in teaching mathematics programming. Education is a key arena for using and understanding digital information and programming in society. Education offers an extraordinary opportunity for developing programming skills (Balanskat & Engelhardt, 2015). Simultaneously, programming is transforming education. Programming is not only an educational tool but also creates new ways of learning and understanding knowledge. Therefore, policies for programming in education are crucial, as how policies are developed in educational practices. In general, there is agreement in policy that programming is important. As this literature review showed, sparse research exists on the educational potential of programming in a mathematics education. On the one hand, most of these articles drew out results showing better performance in mathematics and higher motivation to learn mathematics. On the other hand, the generalizability of these result is less clear. Most European countries face the situation, in which programming is included in a mathematics education (Balanskat & Engelhardt, 2015). However, we call for more research and research-based arguments in the policy for including programming in a mathematics education. A need exists for a better understanding of how programming is politically conceptualized and how these conceptualizations constitute educational practice.

There appear to be two ways in which countries could introduce programming into the curriculum: as a separate subject (such as technology or computing) or via integration into existing subjects. With regard to integration, since programming has most often been linked to mathematical thinking, there has been a tendency in several European countries to include programming to develop algorithmic thinking (Grover & Pea, 2013). While there is little doubt that programming skills are important and that they will become more important in the future, firmer guidelines are required as to the role programming has in school and the role it could play in mathematics education. The idea that programming could be helpful in mathematics education was first raised by Papert in the 1980s using the LOGO programming language. During the 1980s, there was great enthusiasm and confidence that LOGO and similar programming languages would radically reform mathematics teaching in primary schools; however, the results from mainstream implementations did not entirely live up to expectations (Misfeldt & Ejsing-Duun, 2015).

Because there have been strong moves to associate programming and mathematics, there is a need for these associations to be better reflected in the research literature. This literature review focused on programming and robots in mathematics education. The aim was to map existing research examining the use of programming in mathematics education to determine whether there was sufficient evidence to justify the integration of programming into mathematics curriculum and to identify areas for further research. In all, 15 selected articles were analyzed to determine the educational potential of programming in a mathematics curriculum. The characteristic themes discovered were increasing student motivation to learn mathematics, improving mathematics performance, and increasing collaborations with different types of teacher roles.

This study concentrates only on studies discussing programming and mathematics education. The limitation of this study is that it does not consider programming in a broader educational perspective, for instance, in other STEM subjects. We are also aware of several studies conducted that have been connected to after school programs and summer camps (especially in USA). Even though they provide interesting information about programming education, we did not consider them as a part of this review about mathematics education. Furthermore, we concentrated on compulsory school education and did not include studies discussing upper secondary education or pre-school education.

As Papert (1980) suggested in 1980, programming has potential in a mathematics education. Programming and robots provide a real-life connection in a mathematics education, which is an important factor in motivating students (e.g. Ke, 2014; Leonard et al., 2016). This potential is important in a mathematics education, which otherwise is experienced as quite an isolated school subject. A typical issue in a mathematics education is that students do not understand the purpose for their learning (e.g. Lambic, 2011).

According to our analysis, at least the geometry part of the curriculum has a natural connection with programming activities. Much of the research on connecting programming to mathematics focused on geometry. We call for more research that connects programming with other fields in mathematics. Programming is related to the development of algorithmic thinking (Grover & Pea, 2013), with the rationale that it fosters problem-solving and logical-thinking skills and motivates students to learn mathematics. If students are given the opportunity to develop such abilities, they must use programming in subjects other than geometry. Regardless, programming activities provide opportunities to make connections with the mathematics curriculum. Programming activities can be connected to curriculum mathematics as least in geometry, but not necessarily in a traditionally prescriptive manner. The curriculum connection depends on the collective choices that students make during their problem-solving activities in programming. Thus, the connection with a mathematics curriculum cannot be predicted.

The potential for programming in a mathematics education make it well suited for mathematics, and the political decision to integrate programming in a mathematics curriculum can be justified. The concrete benefits of programming in a mathematics education depend on many factors that should be considered along with the integration. Programming activities and mathematics learning through these activities do not correspond to traditional learning situations in the classroom. Students' learning processes with programming are often collaborative, and the teacher plays a different role than normal. The most commonly used learning theories in the studies were constructivist or social constructivist learning theories (e.g. Benitti & Spolaôr, 2017). To gain a greater understanding of the potential for programming in a mathematics education, the entire learning process should be considered by viewing learning as a collaborative process of the entire group instead of viewing learning only as an individual cognitive process or a socio-cognitive process. To consider the collective learning of the entire group or class instead of individual learning by analyzing interactions among students, the teacher and the programming tools can provide valuable information in addition to the current knowledge of the usefulness of programming in a mathematics education.

Furthermore, regarding pedagogical practices in the classroom, the role of the teacher is worth consideration because ordinary mathematics teachers will be required to teach programming. Although integrating programming in a mathematics education is a political decision, a comprehensive discussion on required competencies for mathematics teachers is needed. While programming is integrated in the mathematics curriculum, the highly considerable discussion is to also integrate programming to the pre service and in service teacher education curriculum.

Based on our conclusions, our suggestion for future studies is to consider students collective learning processes in mathematics through programming activities, by also discussing the influence of the role of the teacher in students learning processes.

References

- Ardito, G., Mosley, P., & Scollins, L. (2014). We, robot: Using robotics to promote collaborative and mathematics learning in a middle school classroom. (Report). *Middle Grades Research Journal*, 9(3), 73.
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Brussel: European Schoolnet
- Barak, M., & Assal, M. (2018). Robotics and STEM learning: Students' achievements in assignments according to the P3 Task Taxonomy – practice, problem solving, and projects. *International Journal of Technology and Design Education*, 28(1), 121–144. doi: 10.1007/s10798-016-9385-9.
- Bartolini Bussi, M., & Baccaglioni-Frank, A. (2015). Geometry in early years: Sowing seeds for a mathematical definition of squares and rectangles. *Mathematics Education*, 47(3), 391–405. doi: 10.1007/s11858-014-0636-5
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978–988. doi: 10.1016/j.compedu.2011.10.006
- Benitti, F. B. V., & Spolaôr, N. (2017). How have robots supported STEM teaching? In M. S. Khine (Ed.), *Robotics in STEM Education: Redesigning the Learning Experience* (pp. 103–129). Springer, Cham. doi: 10.1007/978-3-319-57786-9_5
- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). *The Nordic approach to introducing Computational Thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group*. doi:https://doi.org/10.17471/54007
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. on the iPad. *Journal of Computer Assisted Learning*, 32(6), 576–593. doi: 10.1111/jcal.12155
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. doi: 10.3102/0013189X12463051
- Husain, H., Kamal, N., Ibrahim, M. F., Huddin, A. B., & Alim, A. A. (2017). Engendering problem solving skills and mathematical knowledge via programming. *Journal of Engineering Science and Technology*, 12(12), 1–11.
- Hussain, S., Lindh, J., & Shukur, G. (2006). The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data. *Educational Technology & Society*, 9(3), 182–194.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26. doi: 10.1016/j.compedu.2013.12.010
- Khasawneh, A. A. (2009). Assessing Logo programming among Jordanian seventh grade students through Turtle geometry. *International Journal of Mathematical Education in Science and Technology*, 40(5), 619–639. doi: 10.1080/00207390902912845
- La Paglia, F., La Cascia, C., Francomano, M., & La Barbera, D. (2017). Educational robotics to improve mathematical and metacognitive skills. *Annual review of Cybertherapy and telemedicine*, 15, 70–75.
- Lambic, D. (2011). Presenting practical application of mathematics by the use of programming software with easily available visual components. *Teaching Mathematics and Its Applications: An International Journal of the IMA*, 30(1), 10–18. doi: 10.1093/teamat/hrq014
- Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 25(6), 860–876. doi: 10.1007/s10956-016-9628-2

- Lindh, J., & Holgersson, T. (2007). Does lego training stimulate pupils' ability to solve logical problems? *Computers & Education*, 49(4), 1097–1111. doi: 10.1016/j.compedu.2005.12.008
- Misfeldt, M., & Ejsing-Duun, S. (2015). Learning mathematics through programming: An Instrumental Approach to Potentials and Pitfalls. In K. Krainer & N. Vondrová (Eds.), *CERME9: Proceedings of the Ninth Congress of the European Society for Research in Mathematics Education* (pp. 2524–2530). Prague, Czech Republic: Charles University in Prague, Faculty of Education and ERME.
- Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education: Research*, 15, 283–303. doi: 10.28945/3521
- Nugent, G., Barker, B., & Grandgenett, N. (2008). *The Effect of 4-H robotics and geospatial technologies on science, technology, engineering, and mathematics learning and attitudes*. Paper presented at the EdMedia: World Conference on Educational Media and Technology 2008, Vienna, Austria.
- Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. (2009). *The use of digital manipulatives in K-12: Robotics, GPS/GIS and programming*. Paper presented at the Frontiers in Education Conference. doi: 10.1109/FIE.2009.5350828
- Opetushallitus. (2014). Perusopetuksen opetussuunnitelman perusteet [Core curriculum for basic education]. Helsinki. Retrieved 12.07, 2018, from https://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.
- Schwab, K. (2017). *The fourth industrial revolution*. Crown Business.
- Sinclair, N., & Patterson, M. (2018). The Dynamic Geometrisation of Computer Programming. *Mathematical Thinking and Learning*, 20(1), 54–74. doi: 10.1080/10986065.2018.1403541
- Skolverket. (2018). Curriculum for the compulsory school, preschool class and school-age educare 2011, revised 2018. Retrieved 12.10, 2018, from https://www.skolverket.se/sitevision/proxy/publikationer/svid12_5dfce44715d35a5cdfa2899/55935574/wtpub/ws/skolbok/wpubext/trycksak/Blob/pdf3984.pdf?k=3984
- Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia - Social and Behavioral Sciences*, 8, 561–570. doi: 10.1016/j.sbspro.2010.12.078
- Yelland, N. (1995). Mindstorms or a storm in a teacup? A review of research with Logo. *International Journal of Mathematical Education in Science and Technology*, 26(6), 853–869. doi: 10.1080/0020739950260607