World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:15, No:5, 2021

An Empirical Study of the Effect of Robot Programming Education on the Computational Thinking of Young Children: The Role of Flowcharts

Wei Sun, Yan Dong

Open Science Index, Mechanical and Mechatronics Engineering Vol:15, No:5, 2021 publications.waset.org/10012025.pdf

***Abstract*—**There is an increasing interest in introducing computational thinking at an early age. Computational thinking, like mathematical thinking, engineering thinking, and scientific thinking, is a kind of analytical thinking. Learning computational thinking skills is not only to improve technological literacy, but also allows learners to equip with practicable skills such as problem-solving skills. As people realize the importance of computational thinking, the field of educational technology faces a problem: how to choose appropriate tools and activities to help students develop computational thinking skills. Robots are gradually becoming a popular teaching tool, as robots provide a tangible way for young children to access to technology, and controlling a robot through programming offers them opportunities to engage in developing computational thinking. This study explores whether the introduction of flowcharts into the robotics programming courses can help children convert natural language into a programming language more easily, and then to better cultivate their computational thinking skills. An experimental study was adopted with a sample of children ages six to seven (N = 16) participated, and a one-meter-tall humanoid robot was used as the teaching tool. Results show that children can master basic programming concepts through robotic courses. Children's computational thinking has been significantly improved. Besides, results suggest that flowcharts do have an impact on young children's computational thinking skills development, but it only has a significant effect on the "sequencing" and "correspondence" skills. Overall, the study demonstrates that the humanoid robot and flowcharts have qualities that foster young children to learn programming and develop computational thinking skills.

***Keywords*—**Robotics, computational thinking, programming, young children, flowcharts.

I. Introduction

RECENTLY, early childhood (preschool to grade two) education paid attention to integrate digital technologies into young children's curricula [1]. As smart devices, for example computers and AI learning toys, are increasingly becoming popular among young children. Therefore, how to define appropriate learning content and activities for children of different ages to develop their digital literacy competence is a challenge faced by educators and researchers. Programming is considered crucial digital literacy [2], and everyone should learn how to programme [3]. Computers can become useful programming learning tools, as it offers new ways of representing and interacting with information and a new category of "objects to think with" [4]. In the process of programming learning, individual can understand how computers work and improve computational thinking. With the development of information technology, young generation have more access to artificial intelligence, such as virtual reality, intelligent robots, etc. Robotics become powerful learning tools, because it provides a playful and tangible way for young children to be exposed to technology, and allows children controlling a robot through programming to develop computational thinking concept [5]. Computational Thinking (CT) has been described by [6] as a crucial 21st Century skill, and it is intrinsically related to programming [7]. Previous studies have shown that four-six years old children are already capable to program basic robotics projects [8]-[10]. Although great efforts have been devoted to study on robotics and programming education on later schooling, teaching these contents at the foundations stage of early childhood can be a beneficial experience for young learners [5].

Although research shows that engaging in robotic programming is beneficial for children develop fine-motor skills and CT, there are still some challenges in teaching programming to children, especially young children [11]. These challenges are, for example, the effective ways of helping children switch between natural and programming languages [12]; and the best ways of teaching CT [6]. Previous study has shown that flowcharts could help students visualize their program projects and avoid syntactic bugs when they design algorithms [13]. It has been suggested that flowcharts play a major role in the programming teaching [14]. However, so far, less efforts have been devoted to exploring flowcharts in young children programming learning. In this sense, the present study aims to fill in this research gap and explore the effects of flowcharts of programming learning of early childhood. As such, our study addresses the following research questions:

- RQ1: Whether robot programming education can improve the computational thinking of young children?
- RQ2: Whether flowcharts can help to improve the programming learning performance of younger children?

II. Related Work

A. Robot Programming for Young Children

Since 1960s, children programming has been on the spotlight. Based on Papert's constructionism ideas, a large amount of programming languages for children have been

Wei Sun is with the Faculty of Education, Beijing Normal University, Beijing, 100875, P. R. China (e-mail: bnusw0512@mail.bnu.edu.cn).
Yan Dong is with the Faculty of Education, Beijing Normal University, Beijing, 100875, P. R. China (corresponding author, e-mail: yan.dong@bnu.edu.cn).

Open Science Index, Mechanical and Mechatronics Engineering Vol:15, No:5, 2021 publications.waset.org/10012025.pdf

developed [15]. However, there is little doubt that programming is a challenging process for all-ages learners. With the development of technology, developers create new programming languages where children can learn programming by interacting with physical objects based on tangible user interfaces [16]. These new languages are believed to reduce learners' cognitive effort during the learning process, and enable them exclusively devoted on the programming itself. Robots provide a playful and tangible way for young children to be exposed to technology, and controlling a robot through programming offers the young generation opportunities to engage in learning CT concepts and developing other skills. For example, robot programming allows children to develop meta-cognitive, collaboration, and reasoning skills [17], [18]. Additionally, previous research shows that human-like robots could enhance young early childhood social interaction, increase their interest and develop children's programming language skills [19]. Even though robots are believed to be more efficient than other ways, there is limited empirical research exploring the possible advantages of robot programming on young children's CT. Thus, this study aims to examine the effect of robot programming and provide ideas to future young children's programming instructional designs.

B. Computational Thinking

CT has been first reported by Wing [6] as CT is taking approaches to solving problems, designing systems, and understanding human behavior that draws on the concepts fundamental to computer science. The essence of CT is thinking like a computer scientist when confronted with a problem, so it has become a necessary 21st Century skill. Thus, including CT in the basic curriculum is considered vital for citizens living in an increasingly programmable world. Moreover, there is an increasing pressure to engage early elementary school children in CT, as they are faced with marketed ever-changing array of digital tools [20]. However, CT cannot be integrated into K-12 curriculum successfully if without attention to assessment. Furthermore, measures could enable instructors to assess what children have learned need to be validated. Despite many studies have explored integrating CT into early elementary school, there is still a lack of evaluation of students' CT after participating in activities. Therefore, this study aims to examine participants' CT after the designed activities.

C. Flowchart in Programming

Previous study has shown that when students are learning coding, they often spend more time dealing with syntax issues rather than solving the problem itself [13]. Especially for young children, in our experience, in the process of learning programming, young children have some difficulty in converting natural languages into programming languages. For example, if the teacher asks the children to execute the instruction of "go straight first and then turn left", the children will be easy to operate successfully. But when the children were asked to write the instructions and help the machine/robot complete the action, the success rate was significantly reduced. Although the development of tangible programming tools provides a powerful environment for children to master programming, in many cases children just complete the task in the process of trial and error, but do not really master. Additionally, despite some studies have explored how to help students better realize the conversion between natural language and programming language, such as using mind maps [21], flowcharts [22], etc. But these studies have focused on students in the upper grades of elementary school and beyond, with few focusing on young children. Thus, this study adds flowcharts drawing into children's programming learning to help children better convert natural language into programming language.

III. Research Design

A. Participants

The study adopted an experimental design, in which 16 first graders (age from 6 to 7) from one class. All students were told that participating in the experiment was voluntary and would not affect their grades, and they could quit from the experiment at any time. Besides, their personal information was hidden to protect them.

B. Procedure

Fig. 1 shows the experimental procedure. This experiment lasted for 2 months, with 6 classes and 1 hour for each class. Before the learning activity, the students spent 40 minutes to complete a set of CT tests as a pre-test. During each learning process, the students were asked to draw flowcharts before coding (see Fig. 2 for more details), and then check the results of programming (see Fig. 3) with a one-meter-tall humanoid robot (see Fig. 4). After these 6 classes, students took the post-test of CT and some were interviewed by the researcher.

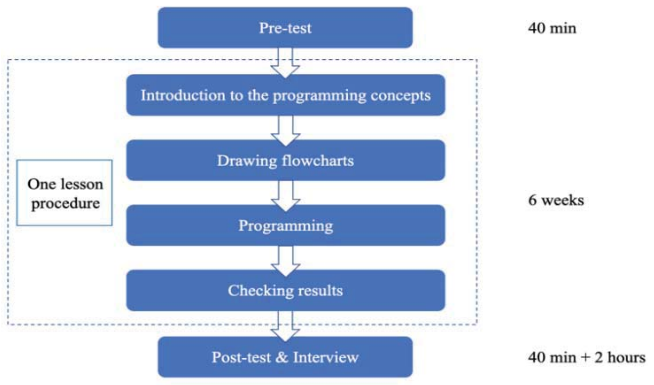


Fig. 1 Diagram of the experimental procedure

C. Instruments

The pre- and post-CT tests were adopted Bebras International CT Test, which is aiming to promote CT among school students at all ages. The present study used tests for children aged 5 to 8 years, and each test includes 9 items. The researchers translated items into Chinese and invited two experts to review them in order to make sure the translations were accurate. Fig. 5 shows the sample item.

Open Science Index, Mechanical and Mechatronics Engineering Vol:15, No:5, 2021 publications.waset.org/10012025.pdf

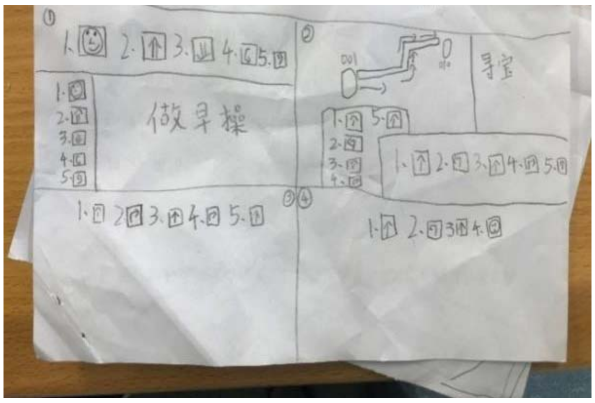


Fig. 2 A flowchart drawn by a student

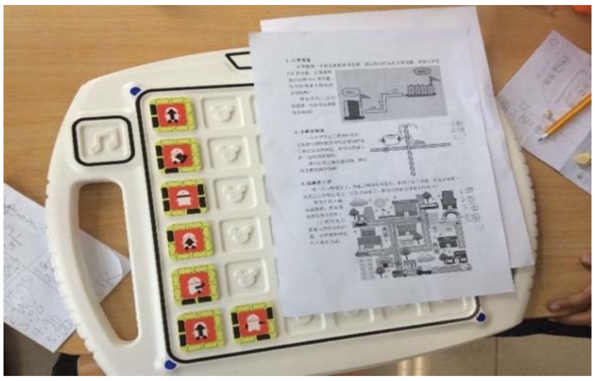
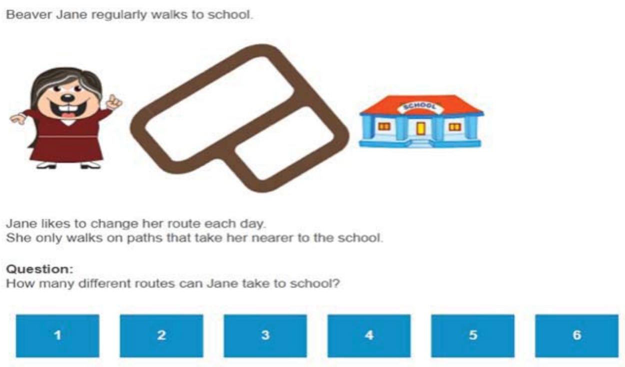


Fig. 3 The instruction blocks placed by the students

Fig. 4 The humanoid robot for the experiment

For the evaluation of CT, this study refers to the evaluation dimension of Bers et al. [17] mainly focusing on debugging, correspondence, sequencing, and control flow. Debugging is a form of problem-solving used in computer science. It includes four steps, and the first step is to recognize that something is not working. Then children need to decide to keep the original goal or change to alternative approaches. The third step is to generate a hypothesis which may be answering the question. The last step is attempting to solve the problem. Each programming instruction is a symbol for the action the machine/robot could act. In order to program correctly, students need to understand the meaning of each instruction and select specific one to represent their intended outcome for the robot's behavior. Correspondences is the process of matching exactly what the robot wants to act with the block of instructions. Sequencing refers to put actions in the correct order. Control flow means that programmers can control the order, and robots can carry out instructions non-sequentially. After each activity, researchers evaluated the program made by children to assess child's level of understanding of programming knowledge. Researchers scored students' achievement following 6-point Likert scale, from 0 did not attempt to 5 complete achievement of the goal, task, or understanding. Moreover, after all learning activities, researchers evaluated students' programming concepts through a comprehensive project. The final work's evaluated scale was developed by the researchers according to the knowledge involved in the activity.



Beaver Jane regularly walks to school.

Jane likes to change her route each day.
She only walks on paths that take her nearer to the school.

Question:
How many different routes can Jane take to school?

Fig. 5 Bebras International CT Test

IV. Results

Wilcoxon analyses were conducted to compare scores on each concept from one lesson to the next. Wilcoxon is a non-parametric test. Due to the small number of samples in this study, it may not conform to the normal distribution, so Wilcoxon is selected. SPSS 20.0 which is a widely used statistical analysis software was adopted to do data analysis. Table I shows the Wilcoxon result of the students' pre- and post-test of CT. It was found that students showed significantly higher CT after the learning activity. This finding conforms to previous studies' reports that young children could develop CT by programming activities [8].

TABLE I
THE PRE- AND POST-CT TEST WILCOXON RESULT

	N	M	SD	*p*
pre-test	15	3.376	0.96	0.002^{**}
post-test	15	5.133	0.99	

$^{**}p < .01$

A. Debugging

Average scores (see Table II) of each step of debugging fell in the range of partial to mostly complete achievement of the goal, task, or understanding. From average scores, we could find that the scores of children in the aspect of debugging are gradually improved. Wilcoxon analyses were run for all the debugging skill variables. As shown in Table III, average debugging score did not change significantly.

Open Science Index, Mechanical and Mechatronics Engineering Vol:15, No:5, 2021 publications.waset.org/10012025.pdf

TABLE II
THE PRE- AND POST-CT TEST WILCOXON RESULT

Lesson	Step 1			Step 2		
	N	Mean	SD	N	Mean	SD
1	6	3.67	1.03	6	2.82	1.72
2	6	3.83	1.17	6	3.17	0.75
3	6	4.33	0.52	6	3.83	0.75
4	6	4.00	0.63	6	3.67	1.86
5	6	4.33	0.52	6	4.17	0.75
6	6	4.50	0.55	6	4.33	0.82
Lesson	Step 3			Step 4		
	N	Mean	SD	N	Mean	SD
1	6	2.83	0.75	6	3.16	0.98
2	6	3.17	0.75	6	3.50	1.05
3	6	3.33	0.52	6	3.83	0.75
4	6	3.50	0.52	6	3.83	1.17
5	6	4.00	0.89	6	4.33	0.82
6	6	4.50	0.55	6	4.50	0.84

TABLE III
WILCOXON ANALYSES OF FOUR DEBUGGING STEPS

	Step 1 (*p*)	Step 2 (*p*)	Step 3 (*p*)	Step 4 (*p*)
1	0.785	0.564	0.317	0.655
2	0.276	0.157	0.564	0.157
3	0.157	0.655	0.705	1.000
4	0.157	0.317	0.180	0.317
5	0.317	0.317	0.180	0.564

B. Correspondence

The mean score on students' abilities to choose correct instructions was rising steadily in all activities (see Table IV for detailed means). In the first two classes, students' correspondence skills were lower, however, in Activity 5 and 6, students could quickly and correctly choose the instructions to complete projects. From Table V, it can be found that there are significantly changes of average correspondence score.

TABLE IV
CORRESPONDENCE AVERAGE SCORES

Lesson	N	Mean	SD
1	6	1.501	0.548
2	6	1.670	0.516
3	6	2.672	0.516
4	6	3.330	0.516
5	6	4.502	0.548
6	6	4.834	0.408

TABLE V
WILCOXON ANALYSES OF CORRESPONDENCE

	p
1	0.655
2	0.034*
3	0.046*
4	0.038*
5	0.157

*$p < .05$

C. Sequencing

Sequencing ability was introduced in all activities, in which students need to arrange instructions in the correct order. From Table VI, it can be found that average scores of sequencing skills increased across the six activities, and between lesson 2 and lesson 3 it was significantly changed (see Table VII).

TABLE VI
SEQUENCING AVERAGE SCORES

Lesson	N	Mean	SD
1	6	2.83	0.753
2	6	3.00	0.632
3	6	3.67	0.816
4	6	4.17	0.753
5	6	4.50	0.578
6	6	4.67	0.516

TABLE VII
WILCOXON ANALYSES OF SEQUENCING

	p
1	0.564
2	0.046*
3	0.180
4	0.414
5	0.564

*$p < .05$

D. Control Flow

The concept of "control flow" was introduced for creating looping or branching programs in the Lesson 4. Young children achieved a "partially complete understanding of the goal or task", comparing to other concepts, the master level of control flow needs to be improved. There was no significant difference found during the last three lessons. The results were listed in Tables VIII and IX.

TABLE VIII
CONTROL FLOW AVERAGE SCORES

Lesson	N	Mean	SD
4	6	3.50	1.048
5	6	3.67	0.816
6	6	3.83	0.753

TABLE IX
WILCOXON ANALYSES OF CONTROL FLOW

	p
1	0.783
2	0.564

After all lessons, the final programming project was used to test students' programming knowledge. Sequence, parallel and determine statement were easy to master, compared to determine statement and function concept (see Table X for detailed scores).

TABLE X
THE FINAL PROJECT SCORES

	Sequence (20)	Parallel (20)	Looping (20)	Determine statement (20)	Function (20)	Total
Mean	20	18.5	18.5	16.4	16.5	89.9

Open Science Index, Mechanical and Mechatronics Engineering Vol:15, No:5, 2021 publications.waset.org/10012025.pdf

V. Discussion and Conclusions

This study shows that there is significant difference between pre- and post-CT test. Young children's CT could be improved through robot programming learning, which confirms previous studies [8]. Through our observation, young children are willing to interact with the robot because of the playful and tangible learning environment. On one hand, humanoid robots can stimulate students' interest in programming learning, especially young children. “The robot looks like our friend, so we can talk and play with it” (Student 3). “It is so cute; I like it very much” (Students 4 and 5). On the other hand, robots provide a more tangible programming learning environment compared to text programming or other means. Robots can carry out the actions following the instructions, which will also enhance students' interest in programming learning. “We could program a robot, it was amazing, and we want to learn more (Student 1)”. “Before I thought programming is difficult and boring, because I saw my father just programming in front of computer. But now I learn programming with the robot, I find it is more interesting and I enjoy it” (Student 2). Robots enhance the interests of learning programming [23], so young children enjoy learning programming. In the process of learning, they master basic programming concepts and develop their CT.

In the present study, we add drawing flowcharts activities in each lesson. Through our observation, during the process of drawing flowcharts, the students discussed more and asked more coding-related questions, which shows that the students were thinking while drawing the flowchart. By comparing the programming learning performance of the students in this term with that of the students in the previous term, the teachers found that the accuracy rate of this term was higher, particularly in the process of converting natural language into programming language. The process of drawing flowcharts can help students carry out logical analysis [22]. This process can help students to have a "programming exercise" in advance so that they can perform better in later programming activities. “Flowchart is a useful tool that I find easier to do programming tasks with (Student 6).” “Without the flowcharts, I just kept trial-and-error, because after placing the instruction blocks, I could just let the robot operate according to the programming to check right or wrong. I usually didn't think about it in detail, because robots could give programmers feedback quickly. But when it comes to flowcharts, I need to think more logically and in detail (Students 2 and 4).” The teacher also mentioned that flowcharts did help. It can not only promote students' thinking, but also improve the accuracy of programming compared to the students' performance in the last term. Additionally, the statistical results show that flowcharts have a significant effect on the "sequencing" and "correspondence" skills.

Robotics programming education provides an environment for young children to learn programming. Its characteristics of playful and tangible can enhance the learning interest of young children and promote their programming learning. At the same time, with the learning of programming, their CT has also been developed. Flowchart is an effective tool for young children to learn programming, which can help children to better think logically and improve their programming learning performance.

VI. Limitations and Future Study

There are some limitations to this study that should be noted. First, due to the lack of use of humanoid robots in first-grade formal classrooms in China, the number of participants in this study is not large enough. Future research can invite more participants to make the research results more representative and universal. Second, the data of this experiment were taken by two researchers through the camera to obtain the process data of children's programming operation. This kind of way of collecting distracted the attention of the students; they will try to communicate with the researchers or try to fiddle with data collection devices while recording. Such way of data collection was not able to completely recorder their natural learning. Future research could use more intelligent methods for data collection. Third, this study is an analysis of only one group of students, if there is a controlled group for comparison, more rigorous experimental results will be obtained. Future study may adopt a quasi-experimental method to obtain more detailed experimental conclusions.

References

[1] Barron, B., Cayton-Hodges, G., Bofferding, L., Copple, C., Darling-Hammond, L., & Levine, M. (2011). Take a giant step: A blueprint for teaching children in a digital age. New York: The Joan Ganz Cooney Center at Sesame Workshop.

[2] Wong, G. K. W., & Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. Interactive Learning Environments, 28(4), 438-450.

[3] Guzdial, M. (2008). Education: Paving the way for computational thinking. Communications of the ACM, 51(8), 25-27.

[4] Papert, S. (1980). Mindstorms. Children, computers and powerful ideas. Basic Books.

[5] Bers, M. U. (2008). Blocks, robots and computers: Learning about technology in early childhood. Teacher's College Press.

[6] Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

[7] Wing, J. M. (2011). Computational Thinking: What and Why. http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

[8] Bers, M. U., Ponte, I., Juelich, K., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics into early childhood education. Information Technology in Childhood Education, 123-145.

[9] Kazakoff, E., Sullivan, A., & Bers, M. (2012). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Childhood Education Journal, 41(4), 245–255.

[10] Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. International Journal of the Learning Sciences, 17(4), 517–550.

[11] Clements, D. H., & Sarama, J. (2010). Teaching with computers in early childhood education: strategies and professional development. Journal of Early Childhood Teacher Education, 23(3), 215-226.

[12] Xu, Y. (2011). Break through the difficulty of programming teaching by using the role experience of little turtles. Journal of Fujian Education Institute, 9, 125-126. (In Chinese)

[13] Carlisle, M. C., Wilson, T. A., Humphries, J. W., & Hadfield, S. M. (2004). Raptor: introducing programming to non-majors with flowcharts. Journal of Computing Sciences in Colleges, 19(4), 52-60.

[14] Liang, Y., Zou, H., & Dai, J. (2018). Application of flow chart in programming language teaching, Education Teaching Forum, 386(44), 186-187. (In Chinese)

[15] Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, 37(2), 83-137.

[16] Shaer, O., & Hornecker, E. (2009). Tangible user interfaces: past, present,

and future directions. Human-Computer Interaction, 3(1-2), 1-137.

[17] Bers, M. U., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. Journal of Technology and Teacher Education, 21(3), 355–377.

[18] Clements, D. H., & Meredith, J. S. (1992). Research on logo: Effects and efficacy. http://el.media.mit.edu/logo-foundation/pubs/papers/research_logo.html.

[19] Ioannou, A., Andreou, E., & Christofi, M. (2015). Pre-schoolers' interest and caring behaviour around a humanoid robot. Techtrends: Linking Research and Practice to Improve Learning, 59(2), 23–26.

[20] Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The impact of user interface on young children's computational thinking. Journal of Information Technology Education: Innovations in Practice, 16, 171-193.

[21] Zhang, J. (2018). Mind mapping: Explore new ways for children to learn Scratch programming. Survey of Education, 7(16), 36-38. (In Chinese)

[22] Wei, T., & Qian, Y. (2019). Flowcharts help elementary school students develop their problem-solving skills in Scratch programming learning. China Information Technology Education, 9, 54-56. (In Chinese)

[23] Sapounidis, T., Demetriadis, S., Papadopoulos, P. M., & Stamovlasis, D. (2018). Tangible and graphical programming with experienced children: a mixed methods analysis. International Journal of Child-Computer Interaction, 19, 67-78.

Open Science Index, Mechanical and Mechatronics Engineering Vol:15, No:5, 2021 publications.waset.org/10012025.pdf